Node-based Quality of Service-aware security framework for sinkhole attacks in mobile ad-hoc networks

TL Phakathi

orcid. org/ 0000-0002-9426-7966

Thesis accepted in fulfilment of the requirements for the degree Doctor of Philosophy

at the North-West University

Promoter: Prof B.M Esiefarienrhe

Co-promoter: Dr F.L Lugayizi

Graduation ceremony: November 2021

DEDICATION

I dedicate this research project to the following people:

- *Mrs. S.M Phakathi (mother)*

- *Mr. M.K Phakathi (father)*

Thank you for your support, love, and encouragement you have shown me.

ACKNOWLEDGEMENTS

Firstly I would like to extend my gratitude to all those who contributed to the completion of this project. I give thanks to God for the wisdom and protection that He has given me during this project and throughout my life. I give deep thanks to the Destiny Group, NWU postgraduate bursary and the faculty of Agriculture, Science and Technology at the North-West University for the opportunity given to me to study here. The experience has been an interesting and rewarding one.

I would like to also extend my sincere gratitude to my supervisor Prof B.M Esiefarienrhe for his guidance, support, advice, for proofreading the drafts, and for making this research possible. I would not have put the topic together if it were not for his direction and guidance.

This research would not be possible without my co-supervisor Dr. F.L Lugayizi whom I started this project with from my BSc Honours. He has been a constant guide and his contribution to this work is immeasurable.

I would also like to thank my family for their continuous prayers and unconditional support. I am also thankful to my brother Sydwell Sihlangu with whom I began this journey. He has been a constant source of inspiration and a pillar of strength during my PhD journey. I would also like to extend my special thanks to my friend Tebogo for always being a friend and an advisor both in life and in my academics. I have learnt a lot from you. It would be a big mistake if I did not thank a close friend of mine, Kitsiso Seete, who contributed immensely towards my entire PhD journey. Her support has carried me through and may God bless her journey.

Lastly, I am deeply thankful to my guardians and mentors Mr. Tom Lin and his lovely wife Gugulethu for their financial support and guidance from the time I started working at Destiny Group to date. I pray that God bless them abundantly.

DECLARATION

I, THULANI LINDELWA PHAKATHI, hereby declare that this project report titled *"Node-based Quality of Service-aware security framework for sinkhole attacks in mobile Ad-hoc networks"* is my work carried out at North-West University, Mafikeng Campus and has not been submitted in any form for the award of a degree to any other university or institution of tertiary education or published earlier. All the material used as a source of information has been duly acknowledged in the text.

Signature: _____ _____ Date: _____02/06/2021_____

Thulani Lindelwa Phakathi

ACKNOWLEDGED

Signature: _____ Date: ___30/06/2021_____

Supervisor: **Prof. B.M Esiefarienrhe**

Department of Computer Science
North-West University
Mafikeng Campus
South Africa

Signature: _____ Date: _____01/06/2021_____

Co-supervisor: **Dr F.L Lugayizi**

Department of Computer Science
North-West University
Mafikeng Campus
South Africa.

ABSTRACT

Most networks strive to provide good security and an acceptable level of performance. Quality of service (QoS) plays an important role in the performance of a network. It is a technology-based metric that is employed to measure how a network performs at any point in time. This area has attracted interest from researchers, developers, and designers. This is mainly because organizations are constantly trying to improve the quality of their products and enhancing the Quality of Experience (QoE) of those products. Degradation leads to a complete network shutdown in most cases and this is where QoS comes in. Network security is initialized to preserve information, provide proof of identity, and also ensure that the information has not been intercepted by unauthorized parties. QoS and security may seem like two independent services but when inter-networked they must work together. For example, the choice of security mechanisms has a direct impact on the effectiveness of the QoS, and the same goes for the choice of QoS having an impact on security. Poor security mechanisms can mean a degraded performance of a carefully queued network. Inappropriate QoS class selection can leak information of packets in the traffic stream but a careful choice in QoS parameters might assist in choosing alternate channels. The design of solutions must always be mindful of these two services. Mobile Ad-hoc networks (MANETs) are generally challenging and the provision of security and QoS becomes a huge challenge. Many researchers in the literature have proposed parallel mechanisms that look into either security or QoS. This research work presents a node-based security framework that is QoS-aware in MANETs using a network protocol called Optimized Link State Routing Protocol (OLSR). Security and QoS targets may not necessarily be similar but this framework seeks to bridge the gap for the provision of an optimally functioning MANET. The framework is evaluated for throughput, jitter, and delay against a sinkhole attack present in the network. The contributions of this research are *a)* implementation of a sinkhole attack using OLSR, *b)* the design and implementation of a lightweight intrusion detection system using OLSR, and *c)* a framework that removes fake routes and bandwidth optimization. The simulation results revealed that the QoS-aware framework increased the performance of the network by more than 70 % efficiency in terms of network throughput and this is major shift compared to other related studies. Delay and jitter levels were reduced by close to 85% as compared to when the network was under attack.

LIST OF ACRONYMS AND ABBREVIATIONS

DSR: Distance Source Routing

IEEE: Institute of Electrical and Electronics Engineers

IoT: Internet of Things

IP: Internet Protocol

MANETs: Mobile Ad-hoc Networks

Mbps: Megabits per second

Ms: Milli second

NetSim: Network Simulator

OLSR: Optimized Link State Routing

OPNET: Optimized Network Engineering Tools

OPPNets: Opportunist Networks

P2P: Peer to Peer

QoE: Quality of Experience

QoS: Quality of Service

RO: Research Objective

RQ: Research Question

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

DECLARATION iii

ABSTRACT iv

LIST OF ACRONYMS AND ABBREVIATIONS v

TABLE OF CONTENTS vi

CHAPTER 1 1

INTRODUCTION 1

1.1 Introduction 1

1.2 Problem Statement 3

1.3 Research Questions 5

1.4 Research Aim 5

1.5 Research Objectives 5

1.6 Research methods of investigation 6

1.6.1 Literature Survey 6

1.6.2 Design a Node-based security Framework 6

1.6.3 Simulate to implement the proposed security framework 6

1.7 Research Workflow/roadmap 6

1.8 Research Limitations 7

1.9 Research Contributions 8

1.10 Thesis Outline 9

CHAPTER 2 11

LITERATURE REVIEW 11

2.1 Chapter Overview 11

2.2 MANETs 11

2.2.1 Topology-constraints in MANETs 11
2.2.2 MANET Protocols 12
2.3 Reactive Protocols (On-demand) 13
2.3.1 Dynamic Source Routing (DSR) Protocol 13
2.3.2 Ad-hoc On-Demand Distance Vector (AODV) Routing Protocol 13
2.4 Proactive Protocols (Table-driven) 14
2.5 Optimized Link State Routing (OLSR) Protocol overview 15
2.5.1 OLSR architectural design 15
2.5.2 OLSR attributes 17
2.5.3 OLSR Routing performance 19
2.5.4 Multi-Point Relays (MPRs) 20
2.6 OLSR core functions 21
2.6.1 Link sensing 21
2.6.2 Packet format and forwarding 21
2.6.3 Topology Control Message 21
2.6.4 Neighbour detection 22
2.6.5 Route calculation 22
2.7 Zone Routing Protocol 22
2.7.1 Intra-zone Routing Protocol IARP 23
2.7.2 Inter-Zone Routing Protocol 23
2.8 Quality of Service 23
2.8.1 QoS support for MANETS 24
2.8.2 QoS metrics 24
2.8.3 QoS and Trust 25
2. 8.4 QoS and video streaming applications 26

2.8.5 QoS and security 26
2.9 Security goals in MANETs 27
2.9.1 Availability 27
2.9.2 Confidentiality 27
2.9.3 Integrity 27
2.9.4 Authentication 27
2.9.5 Anonymity 28
2.10 QoS-aware Routing Protocols 28
2.11 Secure routing in IoT and MANETs 28
2.12 MANET security challenges/vulnerabilities 28
2.12.1 Availability 29
2.12.2 Central Controller 29
2.12.2 Dynamic Topology 29
2.12.3 Power limits 29
2.12.4 Resource availability 29
2.13 Security attacks in MANETs 30
2.13.1 Active Attacks 30
2.13.2 Passive Attacks 30
2.14 MANET-specific threats 31
2.14.1 Black Hole Attack 31
2.14.2 Worm Hole Attack 32
2.14.3 Selfish Behavior 32
2.14.4 Greyhole attack 33
2.14.5 Route Error generation 33
2.14.6 Eavesdropping 33

2.14.7 Blackmail attack 33
2.14.8 Sinkhole attack 34
2.14.9 Traffic analysis 35
2.15 Security Mechanisms 35
2.15.1 Preventive Mechanisms 35
2.15.2 Reactive Mechanisms 36
2.16 Intrusion Detection Systems (IDS) 36
2.16.1 Anomaly-based Intrusion Detection System (AIDS) 36
2.16.2 Signature-based intrusion detection system (SIDS) 37
2.16 Security Frameworks related to the research 37
2.17 Cross-Layer Architecture 51
2.17.1 Security in Cross-layer design 52
2.17.1 Energy reservation in Cross-layer design 52
2.18 Chapter Summary 53
CHAPTER 3 55
RESEARCH METHODOLOGY AND EXPERIMENTAL SETUP 55
3.1 Chapter Overview 55
3.2 Introduction 55
3.3 Research Methodology and Research methods 56
3.3.1 Quantitative Research Methods 56
3.4 Methodology used in this thesis 57
3.4.1 State-of-the-Art Literature Review 57
3.4.2 Simulation 57
3.4.3 Experimental Workflow of the simulation in NetSim 64
3.4.4 Simulation parameters 67

3.5 QoS measures 68
3.6 QoS-aware security Architecture 69
3.6.1 Application Layer 70
3.6.2 Transport layer 71
3.6.3 Network Layer 72
3.6.4 Data Link layer 73
3.7 Chapter summary 74
CHAPTER 4 75
FRAMEWORK FORMULATION, DESIGN AND IMPLEMENTATION 75
4.1 Chapter Overview 75
4.3 Framework overview 75
4.4 Framework Design and Implementation 76
4.4.1 Motivation for Framework design 77
4.4.2 Security keying mechanism 78
4.4.3 Admission Control 85
4.4.4 Resource Reservation Scheme 86
4.4.5 Transport Protocol 86
4.4.6 QoS-aware adaptive routing 87
4.4.7 Packet Queue Management 88
4.4.8 Sinkhole Attack 89
4.4.9 Intrusion Detection System 92
4.4.10 Pathrater 96
4.5 Chapter Summary 100
CHAPTER 5 102
RESULTS AND DISCUSSION 102

5.1 Chapter Overview 102
5.2 Comparison of Results 102
5.2.1 Simulation 1: Performance comparison with 16 nodes 102
5.2.2 Simulation 2: Performance comparison 49 Nodes 106
5.2.3 Simulation 3: Performance comparison of 100 Nodes 111
5.7 Chapter Summary 115
CHAPTER 6 117
SUMMARY,CONCLUSION,RECOMMENDATIONS AND FUTURE WORK 117
6.1 Chapter Overview 117
6.2 Research Summary 117
6.3 Conclusion 118
6.4 Recommendations and Future work 121
6.5 Closing Remarks 122
REFERENCES 123

TABLE OF FIGURES

Figure 1.1 Research workflow 7

Figure 2.1 Classification of typical topology-based protocols [14] 12

Figure 2.2 OLSR's Architectural design [25] 16

Figure 2.3 OLSR Flow Chart[26] 17

Figure 2.4 Neighbour sensing in OLSR protocol[28] 19

Figure 2.5 Multi-Point Relays[28] 20

Figure 2.6 Blackhole attack in a MANET [49] 32

Figure 2.7 Eavesdropping[38] 33

Figure 2.8 Cross-layer Model[87] 53

Figure 3.1 Elements of discrete-event simulation 60

Figure 3.2 Implementation of a simulation model 61

Figure 3.3 Experimental workflow 65

Fig 3.4 Conceptual QoS-aware security architecture [100] 70

Figure 4.1 QoS-aware security framework 78

Figure 4.2 AES encryption[109] 83

Figure 4.3 Step-wise AES encryption [110] 84

Figure 4.4 Decryption system [103] 85

Figure 4.5 Modified OLSR flow chart [114] 88

Figure 4.6 IDS Workflow [117] 93

Figure 4.7 Modified Watchdog declaration code [115] 95

Figure 4.8 Modified Pathrater file declaration [115] 97

Figure 4.9 Modified Pathrater file functions for OLSR [115] 98

Figure 4.10 Modified route reply [115] 99

Figure 4.10 Modified deletion of the route from cache [115] 100

Figure 5.1 Network model with sinkhole nodes 103

Figure 5.2 Throughput for sinkhole and framework scenarios 104

Figure 5.3 Delay for 100 nodes 105

Figure 5.4 Jitter for 16 nodes 106

Figure 5.5 Network model with sinkhole nodes 107

Figure 5.6 Throughput for sinkhole and framework scenarios 108

Figure 5.7 Delay for 49 nodes 109

Figure 5.8 Jitter for 49 nodes 111

Figure 5. 9 Network model with sinkhole nodes 112

Figure 5.11 Throughput for sinkhole and framework scenarios 113

Figure 5.43 Delay for 100 nodes 114

Figure 5.44 Jitter for 100 nodes 115

LIST OF TABLES

Table 3.1 Simulation parameters 67
Table 4.1 AES algorithm processing properties[109]..... 79

CHAPTER 1

INTRODUCTION

1.1 Introduction

A mobile Ad-hoc network (MANETs) is considered an autonomous system that consists of a loosely connected set of self-configuring nodes and can be activated by using various techniques e.g. Bluetooth or WLAN. MANET is autonomous in behaviour because each node is regarded as its host and router at the same time [1]. The hosts and routers rely on direct communication and multi-hops for communication between distant nodes within the network making them scalable and robust. These advantages make them more flexible to accommodate many nodes, decentralize administration and their setup can be placed anywhere and at any time [2]. Contrary to other wireless systems, MANETs do not have a central authority that monitors the forwarding of traffic. In this instance, each node acts as a router to forward traffic to other nodes within the network. MANET systems consist of an infrastructure-less system of associated nodes connected through wireless links [1]. Nodes move arbitrarily or rather in a random fashion [3]. All these characteristics render MANETs vulnerable to security threats like wormhole attack, information disclosure attack, Rushing attack, etc. Communication and packet forwarding is fully dependent on the trust of a neighbour node participating in the network.

MANETs are self-organizing and self-healing. The network must be less prone to security attacks to achieve an acceptable QoS. MANETs have become some of the most prevalent areas of research in recent years because of the challenges they pose to related protocols and they are still regarded as emerging technologies that enable network users to interact without any central infrastructure regardless of their geographical location. That is why they are sometimes referred to as infrastructure-less networks [4]. MANETs have been an active area of researches for the last few years and their growth is promoted by the urgent need to provide the users with network support conveniently [5]. MANETs are primarily useful in military and other tactical applications such as emergency rescue or exploration missions [6]. Their existence stretches back to the 1970s where they were used in the military, meaning that they are not a new concept and their commercial success has been due to advances in wireless technology. Standards have been developed for

routing in MANETs. The Internet Engineering Task Force (IETF) regulates the new group for MANETs. The IEEE standard 802.11 has contributed to research interest done with MANETs [4]. What contributes to their continued growth is the making of smaller and faster devices and this makes MANETs the fastest growing networking area. The mobility of nodes, however, is rapid and unpredictable over time. MANETs, like all wireless networks, are more vulnerable to attacks and weaknesses as compared to wired networks. The limitations of the wireless medium become especially severe in multi-hop networks, where multimedia streams are subject to the aggregate effect (e.g., packet drop and propagation delay) of each link along with a node to node path. The mobility of a node can cause frequent route breaks in MANETs. Updating routes can be time-consuming, thus packets might for shorter periods be lost in bursts since they are sent on non-working routes [7]. Routing is done through hops achieved by using routing protocols. These routing protocols are in each device. Routing relies on packet forwarding. This can be done using the longest distance to the destination or short distance. The most important aspect is getting all/most of the packets to the destination.

Quality of Service (QoS) is a growing technological concept. This growth is promoted by the emergence of real-time applications like video conferencing applications, online gaming, chatting, and instant messaging (IMs). In MANETs, trust is the belief of a node that other nodes within the network will cooperate in a particular way without disturbance and compromise in the network's activity and security. QoS, according to [8], is a set of service requirements that need to be met by a network as packet streams move from source to destination. With the increasing needs of QoS provisioning for evolving applications e.g video and audio, it is most appropriate to support these services in *Ad-hoc* networking environments. In the field of telecommunications, QoS was defined by the ITU in 1994 as comprises requirements on all the aspects of a connection, such as response time during service times, loss, signal-to-noise ratio, crosstalk, echo, interrupts, frequency response, loudness levels, and so on. In the field of computer networking, QoS is the ability to provide a different level of priority to different applications, users, or data flows, or to ascertain a precise level of performance to a data flow [8].To achieve QoS, the concept of routing is important. Routing is the act of moving information from a source to a destination in a network [8]. At least one intermediate node within the network is encountered during the transfer of information. In essence, two activities are implemented in this concept: determining optimal routing paths and transferring the packets through an internetwork [9]. Routing may be either dynamic or static.

Static routing refers to routing done manually or statically in the router while dynamic routing refers to routing that is being learned by an inner or outer routing protocol. QoS is measured using specific metrics within the network. Some of the evaluated metrics include throughput, delay, jitter, bandwidth and packet loss.

Therefore this work seeks to bridge the gap between security and QoS by addressing the main challenges in MANETs that directly impact the performance of the network concerning video streaming applications. One of those challenges is sinkhole attacks which affect not only the security of the network but also its performance. In this research, a QoS aware framework was designed and evaluated to directly deal with these challenges. The work ultimately intends to contribute to QoS improvement of video streaming, security, and network resource optimization in MANETs. To achieve this, the framework was presented to monitor, control, manage, and evaluate QoS without affecting network security.

1.2 Problem Statement

MANETs have become a rapidly growing technology mainly because of their features. The biggest challenge in MANETs is the lack of a robust security solution that can protect them from routing attacks. The design solutions needed are not supposed to cause resource constraints like bandwidth and battery power. QoS and security have conflicting interests because security tends to require many network resources and such is not ideal for real-time streaming applications. The “self-organizing” functionality of MANETs can be its greatest advantage and security weakness as it leaves room open for both passive and active attacks. There are many vulnerabilities poised by MANETs. Some of the vulnerabilities are caused by their dynamic topology, limited power supply, limited bandwidth and scalability. Routing is the key component for MANET technology but secure routing is much more preferred. Existing architectures adopted from wired networks cannot be applied directly to MANETs hence the need to design a robust security framework that will be solely implemented on MANETs [25,31]. The QoS-aware security framework must seek to achieve security goals i.e availability, integrity, anonymity, authentication and confidentiality. To achieve a secure and QoS-aware network, a custom-designed security protocol must be implemented. The problem with MANETs is that they are not well known to have combined solutions to security and QoS-based features because of their dynamic nature . This work focuses

on sinkhole attacks through protocol modification to achieve good security detail without compromising the QoS in real-time [7].

Kaur *et al*. [10] analyzed security threats that MANETs face and presented the security objective that needs to be achieved. Their security objective called Cooperative Bait Detection Scheme (CBDS) was successfully implemented on the black hole and grey hole attacks and it proved to be equally effective in the case of denial of service (DoS) attacks and sleep deprivation attacks. Their simulation results have showed increased detection for CBDS and an enhanced response [10]. This framework provided a good background in terms of the detection of attacks. It did not, however, address issues of QoS in terms of security.

Khan *et al*. [11] proposed a multi-attribute trust framework for MANETs (MATF). The framework was designed to minimize bootstrapping time and tackle selective behaviour. It was designed to enhance the security of MANETs by enabling a node to identify and then quarantine malicious nodes from the routing paths. The proposed security framework was not only designed to detect malicious nodes quickly but also to decrease the number of false positives. They used multi-watchdogs as a measure to achieve this [11]. This framework was effective in design but time-consuming in terms of network resources. Multiple watchdogs ultimately meant that the QoS was not a factor of concern in the design process hence our framework seeks to provide not only security solutions but also QoS-aware solutions.

Literature has proved a great deal of background on how much work has been done in terms of security but most works focus on specific attacks and protocols within MANETs. The area is huge and there is still more room to improve the security of *Ad-hoc* networks despite the dynamic topology they operate in. For example, there is little knowledge provided on sinkhole attacks in OLSR but most articles point towards solutions for routing protocols like DSR, AODV and ZRP.

In this study, the researchers mainly concentrated on bridging the gap between QoS and security by implementing a cross-layer QoS-aware framework compatible enough with both technologies and also having to evaluate the impact of a sinkhole attack in a MANET using OLSR. The network QoS was monitored to see the impact of the proposed framework against the sinkhole attack. While designing the framework, the following areas were also put into consideration:

a) Cross-layer optimization and interactions
b) QoS scheduling and queuing based on application types, QoS classes;

c) QoS expectations
d) Security implementation in layers

Thus, the framework presented in this work has a high degree of QoS consideration by putting the performance of the network at the core of the design process.

1.3 Research Questions

To effectively address the problem this research seeks to solve, some questions were asked and answered as a guide to achieving the objectives of this research. Research questions according to [12] are *"like vehicles that you will rely upon to move from your broad research interest to your specific research focus and project, and therefore their importance cannot be overstated".* Therefore, this work posed the following research questions:

RQ1: To what extent have network developers and researchers addressed security issues in MANETs with consideration for QoS in the network?

RQ2: To what extent do malicious nodes disturb the stability and performance of a network?

RQ3: How can we formulate measures of preventing security breaches in MANETs that are QoS-aware?

RQ4: How can a node-based security framework for sinkhole attacks be formulated, simulated, and evaluated such that the security and QoS of MANETs can be enhanced?

1.4 Research Aim

This research aims to formulate, design, and implement an efficient node-based QoS-aware security framework for sinkhole attacks that addresses security and QoS in MANETs.

1.5 Research Objectives

To achieve the research aim and provide solutions to the various research questions, the following objectives are formulated to:

RO1 Analyze and understand the extent to which network developers and researchers have addressed security issues in MANETs with consideration for QoS in the network.

RO2 Investigate the extent to which malicious nodes interfere with the stability and performance of a network.

RO3 Formulate measures of preventing security breaches in MANETs that are QoS aware.

RO4 Formulate, simulate, and evaluate a node-based security framework for sinkhole attacks such that the security and QoS of MANET can be enhanced.

1.6 Research methods of investigation

To achieve the main goal of this research through the objectives specified, we followed the research methodologies stated below to answer and achieve RQ1 to RQ4 and RO1 to RO4.

1.6.1 Literature Survey

An in-depth survey on the current literature was done to find answers to RQ1, RQ2, and also achieve RO1 and RO2 in section 4.

1.6.2 Design a Node-based security Framework

To fulfil research RO3 and RO4 in section 4, a framework was designed and formulated.

1.6.3 Simulate to implement the proposed security framework

The study was be run using scenarios with sinkhole attacks and others with a counterattack mechanism as applied by the proposed framework. The simulations were implemented in different node densities for different performance metrics.

1.7 Research Workflow/roadmap

Ince *et al*. [13] argue that the term "research workflow" is not unequivocally defined because literature brings about terms such as scholarly workflow, academic workflow, and digital research workflow. For example, scientific laboratory research workflow is mapped out as a cascade of *"current awareness, research collaboration, research design, data collection, lab and safety, experiment design, writing, analysis, sharing, submission, publication, review, showcasing* and *assessment"*. A research workflow can generally be defined as a progression of steps involving

more than one event, consisting of an orchestrated and repeatable pattern of activities toward a planned output. The main goal of this research work was achieved through the workflow shown in Figure 1.1.

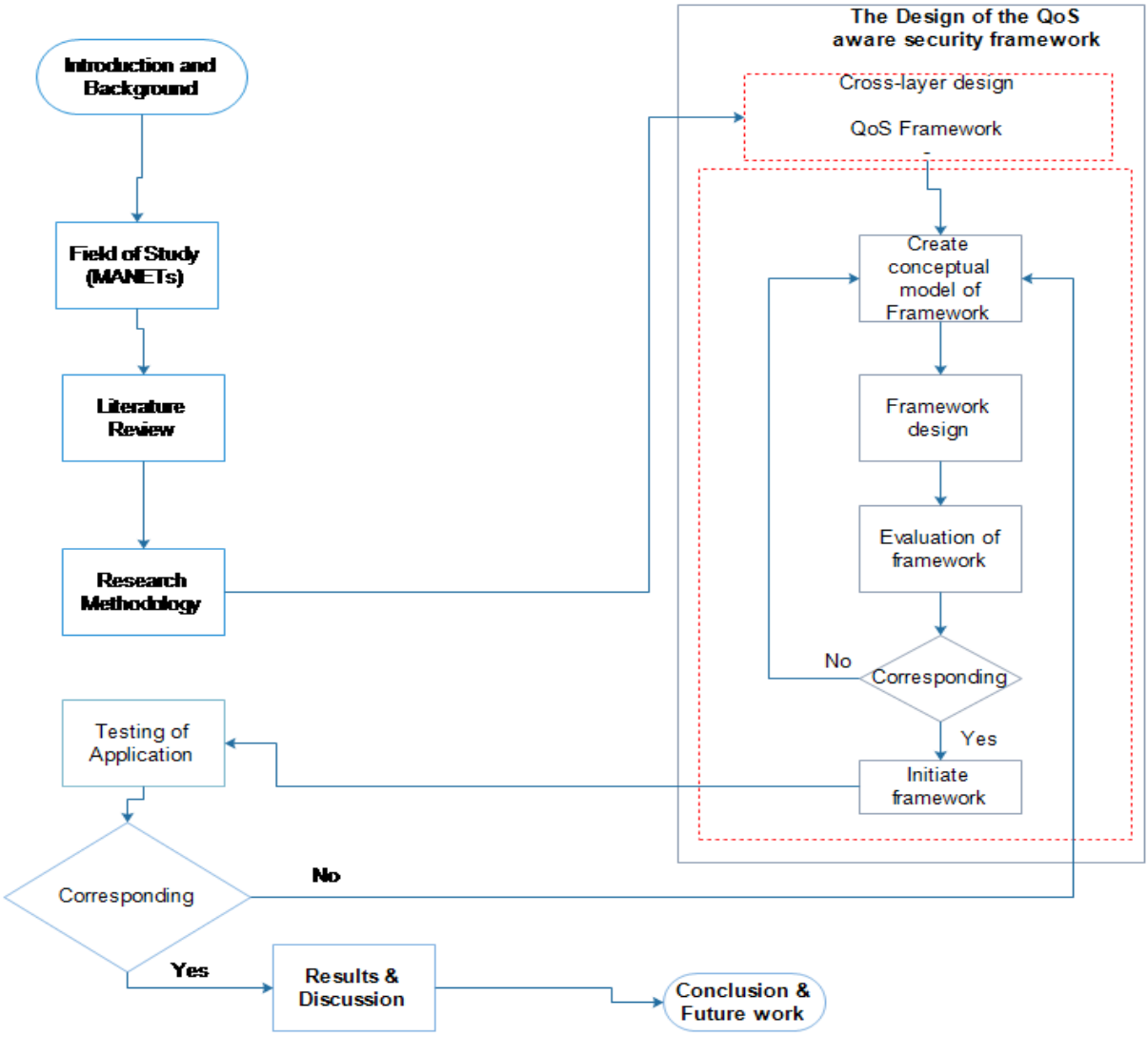

Figure 1.1: Research workflow

1.8 Research Limitations

The scope and quality limitations of this work that may have an impact on the overall performance of the proposed framework include:

a) The evaluation of this work has been on video streaming using OLSR routing in both low, mid, and high-density node environments. Other routing protocols like *Ad-hoc* On-demand Distance Vector (AODV) and Dynamic Source Routing (DSR) were not implemented.

b) The performance parameters this work focused on were throughput, delay and jitter. Other metrics like packet loss and residual error rate, Quality of Experience (QoE), or user perceptions were not included in the overall evaluation process. Network security metrics were not considered as the QoS-aware framework encompasses an intruder detection system that removes blacklisted malicious activity.

c) The open-source code provided by the simulator allows for one attack to be simulated at a time hence the development towards a sinkhole attack using provided code extensions.

1.9 Research Contributions

Security and QoS are two network drivers often considered separately as each follows separate methodologies in terms of evaluation. As much as the performance evaluation of nodes is important, it is equally important to consider the security detail of the spaces in which the network operates. The cross-layered approach of the presented framework is an approach that many engineers can further develop in the quest of making *Ad-hoc* networks more secure and performance-friendly. The development and improvement of the traditional OLSR protocol through the framework is another notable contribution to the area of MANETs and QoS attainment.

a) T. Phakathi, F. Lugayizi, and M. Esiefarienrhe, “Quality of Service-aware security framework for Mobile Ad-hoc networks using Optimized Link State Routing Protocol,” arXiv [cs.NI], 2020.

b) T. Phakathi, F. Lugayizi, B. Esiefarhenrhe, and B. Isong, "On Trust Management Framework in Video Streaming Applications Over Mobile Ad-hoc Networks," in The Third International Conference on Electronics and Software Science (ICESS2017), 2017, p. 97.

c) T. Phakathi, B. M. Esiefarienrhe, and F. Lugayizi, “Comparative analysis of quality of service scheduling classes in mobile ad-hoc networks,” arXiv [cs.NI], 2021.

d) A full paper detailing this work was submitted and still under review (revision) with an accredited international journal called Ad-hoc and Sensor Wireless Networks, ISSN 1551-9899 (print) 1552-0633

1.10 Thesis Outline

In this chapter, we presented the introduction and motivation of this research work and thus provide a brief overview of the current challenges that exist. The work also presents the main goal of the study which is supported by objectives and key research questions that guide the progress of the work. The chapter also outlines the methodology for realizing the main aim of the study and also the study's limitations.

Chapter 2 presents a brief overview of security and QoS. The chapter also brings out the extent to which many researchers have gone to address secure routing models and challenges. An in-depth analysis of the routing protocol, OLSR, is presented from a composition and characteristics perspective to better understand how it could be modified for enhanced performance. It is in this chapter that answers to RQ1 and RO1 were provided. This chapter also presented other works in literature that were related to our framework and discusses both their similarities and differences highlighting the various research gaps which provided the basis for this work.

Chapter 3 presents the research methodology and a detailed experimental setup employed to achieve the goal and objectives of this research. This chapter begins by giving a brief background of the different methodologies that exist to execute scientific research and the different approaches or methods employed to implement them. The chapter also presents the experimental environment where the simulation was conducted including the simulation parameters.

Chapter 4 presents the QoS-aware security framework from a design and implementation perspective together with its operational architecture and components. The chapter also introduces the evaluation arrangement that was used to access the framework architecture implemented. It clearly explains how the sinkhole attack was orchestrated into the network and also how the Intrusion Detection System (IDS) was implemented for the OLSR protocol. The ultimate contribution to this work is presented in this chapter and also provided answers to RQ3, RO3, RQ4, and RO4.

Chapter 5 presents a detailed analysis and discussion of the results obtained in the simulations. It also presented a comparison within network scenarios between a sinkhole attack and the effectiveness of the counterattack framework and thus provided solutions to RQ2 and RO4.

Chapter 6 is the concluding chapter of the work. It summarizes the entire research work and re-echoes the research contributions and future work. What follows this chapter is the Bibliography section containing references and appendixes

CHAPTER 2

LITERATURE REVIEW

2.1 Chapter Overview

In this study, literature remains central and it revolves around mobile *Ad-hoc* network security frameworks that have been implemented by researchers and QoS. This chapter also give answers to RQ1, RQ2, RO1, and RO2.

MANETs have been extensively studied in terms of their operations and intrinsic behaviour. The reviewed literature provided the platform to prove that they are dynamic and that the nodes are independently charged and routing is dynamic as the nodes are mobile. Literature also provides an in-depth outlook of attacks that mostly affect MANETs and driven solutions that aim to prevent such attacks from affecting all possible routing and data integrity. This work looks deeper into the OLSR protocol from architectural design, characteristics, and possible modification of the protocol for better security and QoS routing.

Another aspect of the literature reviewed was the security of mobile *Ad-hoc* networks. Security is important in any network setup as personal or organizational data is involved. Data integrity is key to the day-to-day operations of many organizations.

2.2 MANETs

Traditional networks such as LAN, PAN, and MAN can easily be manipulated as most of them have a centralized controller or server. Topology constraints contribute to the various challenges encountered in MANETs and their routing protocols.

2.2.1 Topology-constraints in MANETs

A secure network topology [1] must be guaranteed through the use of robust routing protocols in the *ad-hoc* networking stream. The protocols are required because of the frequent routing updates caused by the dynamic nature and characteristics existing in MANETs. Providing QoS at a better scale in such environments is a huge challenge [4]. The existing stochastic nature of communications quality in a MANET exhibits challenges in offering concrete guarantees on the applications computed in mobile devices. An adaptive QoS must be implemented coupled with a

strong trust relationship framework over the traditional resource reservation to support multimedia streaming services. Since the topology of the network is constantly changing, the issue of routing packets between pairs of nodes poses a great problem. One may argue that most protocols should at least be based on reactivity instead of proactivity because of this problem. Multicast routing poses a challenge too because the multicast tree ceases to be static since nodes in the network move randomly. The routes between nodes often have multiple hops and that is too complex than a single-hop communication.

In MANETs [4], the nodes are mobile and this feature could result in nodes getting out of range within the network. This causes frequent loss of links between nodes and because of this, it is a general norm that mobility contributes to the additional dynamic nature of the network infrastructure. When nodes are in motion, the state of existing node links are most susceptible to change, or possibly breaks. Re-routing is an alternative to routes that are broken [7].

2.2.2 MANET Protocols

MANET usage depends on the use of routing protocols. These routing protocols are designed to meet the needs of the end-user and to fulfil their primary objective of achieving an excellent QoS in the network. Figure 2.1 shows the protocols in their respective categories.

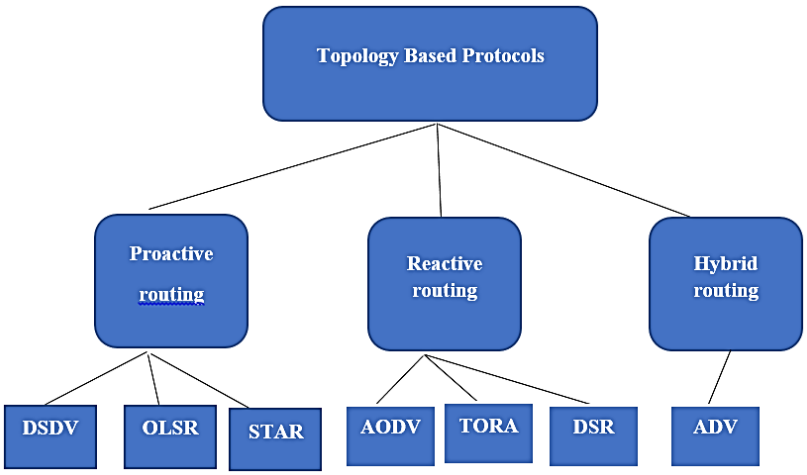

Figure 2.1: Classification of typical topology-based protocols [14]

2.3 Reactive Protocols (On-demand)

According to Hamma *et al*. [15] reactive protocols routing build and maintain routes rather than depending on discovering routes as needed. They do not update their route tables constantly with the latest route topology. In this case, a route to every node is not known at any given time [15]. In essence, reactive routing opens the route only when it necessitates a node to communicate. Additionally, reactive routing consists of a route discovery phase in which the query packets are forced into the network for the path search and this phase finishes when the route is found [16].

2.3.1 Dynamic Source Routing (DSR) Protocol

Dynamic Source Routing (DSR) as a reactive routing protocol starts route discovery only on-demand like AODV but unlike AODV it keeps the entire path to the destination in its routing table instead of the next-hop node. The packet header has the address of all the transitional nodes to which the packet needs to move toward the destination node [17]. The Dynamic Source Routing (DSR) [14] allows nodes to dynamically uncover a source route across complex networks and hop to any destination node. Usop *et al*. [18] wrote that DSR does not have mechanisms in identifying which route in the cache is stable hence the data packet is forwarded to a broken link. According to Singh *et al*. [17], DSR implements three unique methods of controlling packets for path discovery and maintenance namely route reply, route error, and route request.

2.3.2 *Ad-hoc* On-Demand Distance Vector (AODV) Routing Protocol

AODV protocol [19] is an on-demand and table-driven protocol. It embraces flat routing tables with one entry per destination. It is in contrast to DSR, which can sustain numerous route cache entries for every one terminus [20]. AODV [18] surpasses DSR in high mobility environments, in areas where the topology changes swiftly and can adapt to those changes rapidly since it only maintains one route that is actively used. Authors in [18], compared AODV, DSDV and DSR protocols and they found that AODV performance is the best considering its ability to sustain connection by the recurrent trade of information, which is required for TCP based traffic. It virtually dispatched all packets at low node mobility, while failing to converge as the mobility of nodes rises. On the other hand, DSR performed well in terms of mobility rates and movement speeds. The DSDV protocol was found to perform almost as well as DSR, but it still required the

transmission of many routing overhead packets. Furthermore, in higher rates of node mobility, it was found to be more costly compared to DSR [18].

According to Shivahare *et al*. [20], for real-time traffic, AODV is most preferred over the DSR protocol because of its potential to sustain connections by recurrent trade of information, which is strong compliance (requirement) for TCP based traffic [20]. Saravanan *et al*. [21], mentioned that when compared to the other protocols the AODV connection setup delay is very low but also added that sometimes it will squander the bandwidth because of recurrent beaconing leads [21].

2.3.2.1 Advantages of AODV

a) Link breakages within dynamic routes can be handled.

b) Uses destination arrangement numbers to find the latest route to the destination.

c) Route on-demand foundation with a minimal postponement.

2.3.2.2 Disadvantages of AODV

a) Old and un-updated sequence numbers can lead to inconsistent routes.

b) Multiple RREP packets in response to a single RREQ packet may lead to a large overhead.

c) Bandwidth consumption happens with periodic beaconing.

2.4 Proactive Protocols (Table-driven)

Proactive routing protocols are part of topology-based protocols. Proactive protocols utilize the recurrent trade of control messages among routers corroborating that the route to every host is always notable. This, according to Hamma *et al*. [15] table-driven protocols need a high bandwidth overhead. *Ad-hoc* link-state routing algorithm pursues to sustain bandwidth by decreasing the size and the number of control messages [15]. Shivahare in 2012 [20] wrote that each station in the network has a routing table for the broadcast of the data packets that the node wants to establish a connection with on the network. According to the author, these nodes record for all the presented arrival points, the number of hops required to arrive at each destination in the routing table. The routing entry is then compartmentalized with a sequence number that is established by the

destination node. To retain stability, each node broadcasts and alters its routing table timelessly [20].

Xu *et al* [16] wrote that proactive routing protocols tend to update routing information periodically, which performs comparatively in urban environments (with low mobility) but exhibits defective performance in highly mobile and dense scenarios when matched with reactive routing protocols [16]. Furthermore, these findings were further confirmed by Phakathi *et.al*. [22].

2.5 Optimized Link State Routing (OLSR) Protocol overview

OLSR is an optimization of a pure link-state protocol for mobile *Ad-hoc* networks. It falls under the category of proactive routing protocols. In the OLSR protocol, every station found in the network chooses a neighbouring node-set known as the Multi-Point Relays (MPR), which rebroadcasts the packets in turn. To this end, neighbouring nodes identified to be outside the MPR set can only read and process the packets [16]. According to Saravanan and Vijayakumar [21], OLSR retains tracks in the pathfinding table to proffer a route if necessitated. MPRs are primarily responsible for forwarding, declaring link state information, forwarding and controlling traffic, thereby providing effective mechanisms for flooding control traffic by minimizing the frequency of required transmissions [23].

OLSR utilizes two types of control messages: HELLO Messages and Topology Control (TC). HELLO messages are implemented for getting the information concerning the link status and the host's neighbours [24]. OLSR also uses Multi-Interface Declaration (MID) messages. MID messages are transmitted by nodes that are running OLSR on two or more interfaces and they all have listed their IP address as used by that single node.

2.5.1 OLSR architectural design

OLSR's architectural design is that of a protocol best suited for large and dense networks [22]. The protocol is an optimization of the pure link state algorithm. OLSR has a cross-layered design just like that of the OSI (Open Systems Interconnection) model in networks.

Figure 2.2 and 2.3 depict examples of a cross-layer design with three different layers found in one protocol (OLSR) . From the designer's perspective, there are two relative choices in the design

process of the protocol. The first option is to design the protocol following the rules of the reference architecture and that is the higher layer being able to access services provided by the lower layer with no consideration of how such service is made available. Secondly, the routing protocols can be designed in violation of the reference architecture and that is by allowing direct communication between layers that are not adjacent to each other.

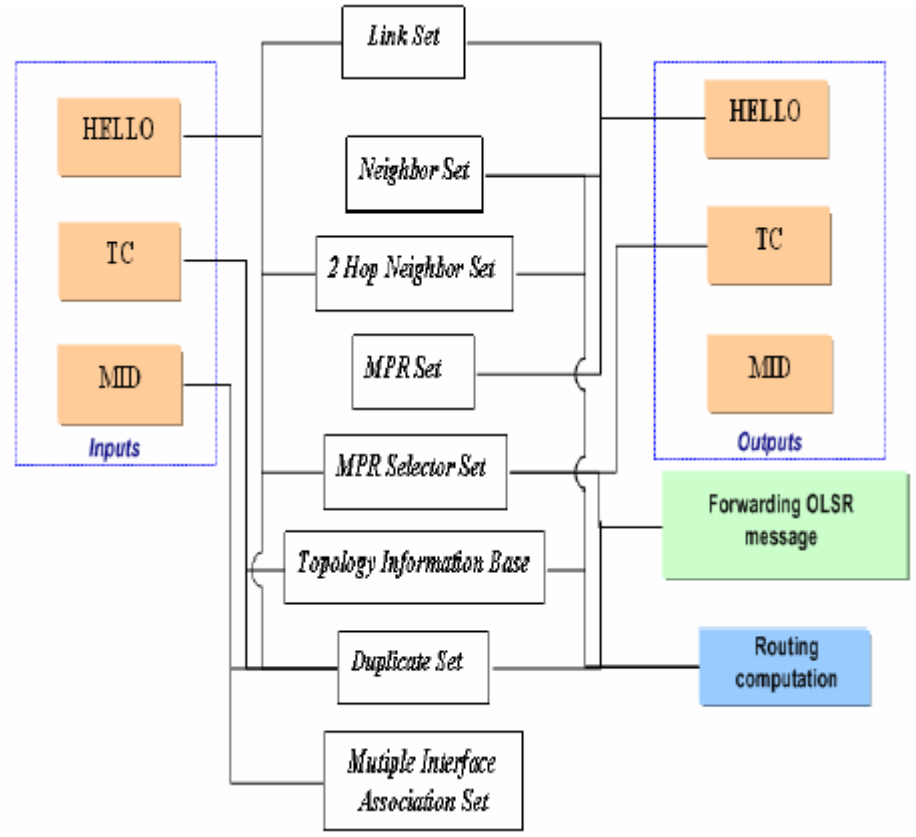

Figure 2.2: OLSR's Architectural design [25]

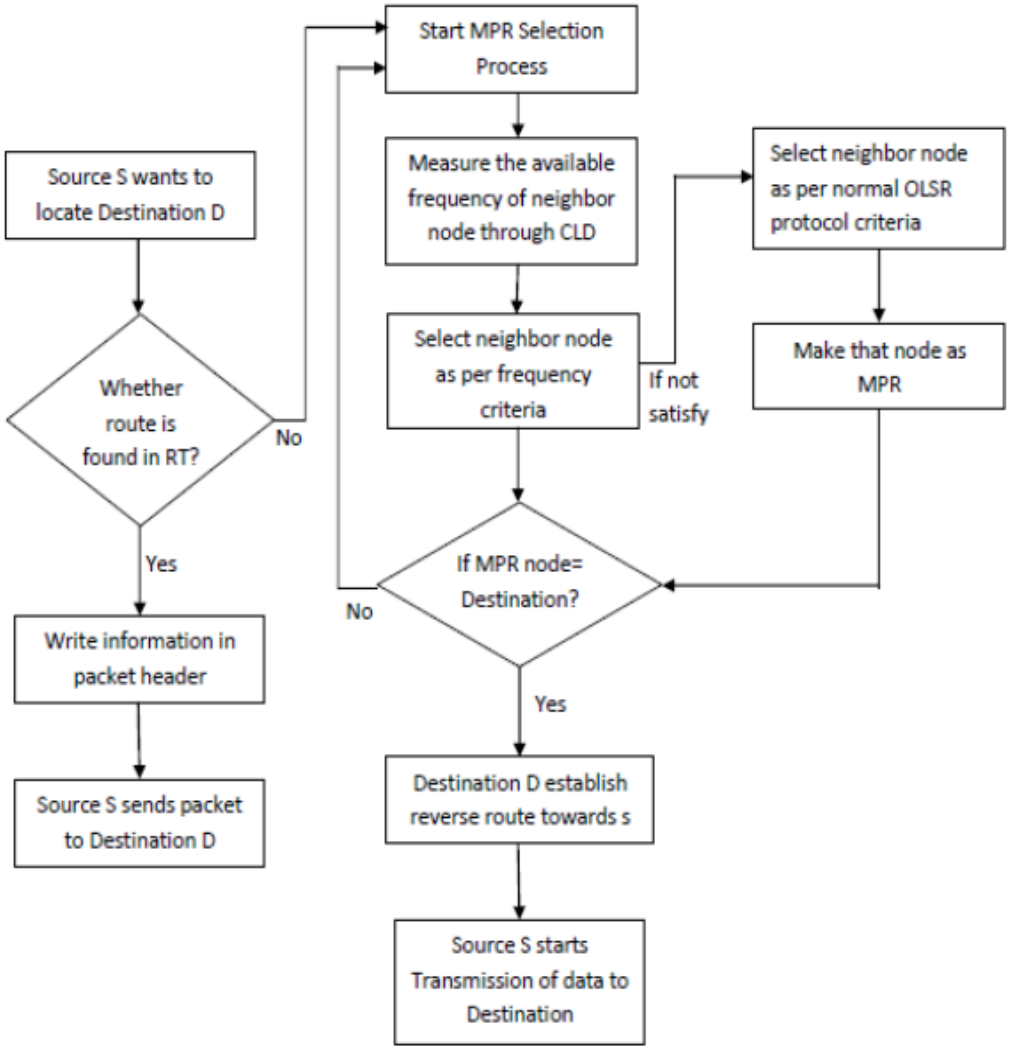

Figure 2.3: OLSR Flow Chart[26]

2.5.2 OLSR attributes

OLSR protocol has the following characteristics:

a) HELLO messages are exchanged [27] among neighbour nodes within a one-hop distance. They may be deployed to encapsulate MPR selection, neighbourhood detection, selection signalling and link sensing. There is a periodic generation of messages containing information on links between network interfaces and neighbour nodes as shown in Figure 2.4.

b) TC messages are periodically generated by MPRs and this according to [28] is done to show other nodes chosen as their MPR. The information is very important for the node's

topological state hence is stored at the topology base of each node in the network and then used for routing table calculations. These messages are then sent to other nodes and ultimately the entire network. The advantage of periodic broadcasting of TC messages makes it easier to distinguish between new and old messages. This is made possible by the sequence number found in each TC message [27].

c) Multi-Interface Declaration (MID) messages are notifications sent by the node to report on the status of network interfaces employed to participate within the network. The report has important information as nodes may have multiple interfaces with diverse addresses [27] participating in the exchange of communication.

d) Neighbour sensing is another phenomenon [29] implanted in the OLSR protocol whereby HELLO messages are broadcasted periodically. There are symmetric and asymmetric links. When a source node receives a HELLO message with an address that link is registered as symmetric and when the address is not found in that HELLO message then the link is said to be asymmetric. The node then sends a HELLO message holding the source node's address. When the source node obtains this message and finds its address it registers that link.

e) Route calculation is implemented through the "shortest path algorithm". Route calculations start when change is detected in either link set, two-hop neighbour set, the neighbour set, topology set, or even the MIAIB (Multiple Interface Association Information Base). The information obtained from the topology set and neighbour set is used to calculate the routing table [28].

OLSR has one major setback which is the constant necessity of maintaining the routing table for all possible routes. This setback may not be severe for scenarios with just a few nodes but may prove to be detrimental for large and dense networks. The overhead control messages may require more bandwidth and thus causing network congestion. This may be one outlined constraint of the protocol but it relatively gives out satisfactory throughput as compared to other MANET protocols [27].

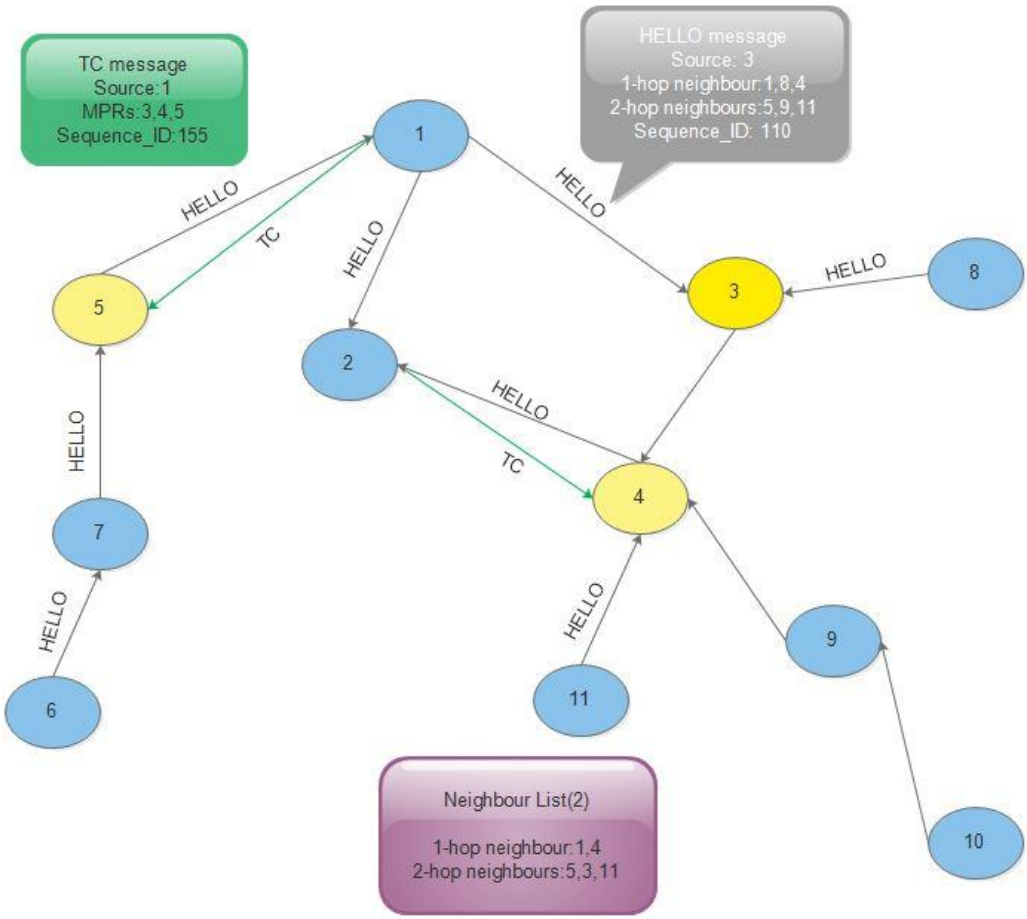

Figure 2.4: Neighbour sensing in OLSR protocol[28]

2.5.3 OLSR Routing performance

OLSR protocol signs routing messages directly [30]. The bandwidth cost of one extended broadcast message is relatively cheaper than having two small messages as they incur significantly higher bandwidth costs. The receiver has to wait for a corresponding message when the routing message and its corresponding signature are sent independently. It is shown that the waiting time at least doubles the per-hop delay for TC messages. To reduce bandwidth consumption, the forwarding nodes should aggregate routing messages into one packet, generate a signature message comprising the whole packet, and send the signature message within this packet. This is done to minimize bandwidth costs. This technique is most suited for larger networks whereby bandwidth

consumption costs are higher and can be thus reduced. In small networks, the probability of aggregating more than one TC message within a reasonable period is small because aggregation delays may increase the per-hop delay to an unacceptable level [30].

To maximize performance, OSLR's cross-layer design seeks to enhance the capacity of wireless networks through the joint optimization of multiple layers within a network. This feature has drawn great interest in protocols that operate through different layer interactions.

2.5.4 Multi-Point Relays (MPRs)

These are nodes that are selected by their one-hop neighbours to retransmit all the broadcast messages they receive from a particular node provided that message is not a duplicate and that the message has a time-to-live field greater than one. Routes are selected by MPRs to avoid data packet transfer problems over uni-directional links. Each node will select its MPR set by using its one-hop neighbours as shown in Figure 2.5. A group of MPR nodes is called an MPR set. The set is chosen such that it covers all symmetric two-hop nodes and a coverage strictly confined within the radio range. An MPR set of node X is denoted as MPR (N)[31].
MPRs minimize the control traffic overhead of OLSR through retransmission of control messages. The technique significantly reduces the rate of transmissions needed to flood a message to all network nodes. The introduction of MPRs is to minimize the overhead of flooding messages and reducing redundant retransmissions in the network[32].

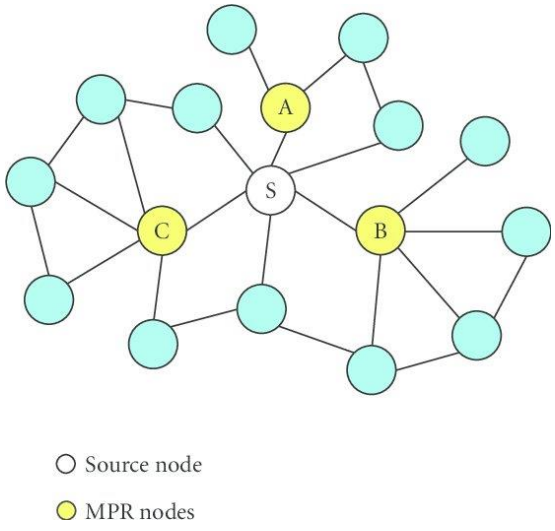

Figure 2.5: Multi-Point Relays[28]

2.6 OLSR core functions

When analyzing the core functions of OLSR, the node's general behaviour, and the interfaces that are equipped with OLSR as a routing protocol are considered. This would mean the messages as they flow within the network and all transmission. The core functionality of the routing protocol, OLSR, is enabled with the following components: link sensing, packet format and forwarding, topology control message,

2.6.1 Link sensing

This is achieved through periodic HELLO messages over the connected interfaces. That would mean that each interface gets a HELLO message. Link sensing describes the links between local and remote interfaces and that is a node and its neighbouring nodes. A local link set is established through link sensing. The local link set facilitates the link between interfaces. In an event whereby sufficient information is obtained from the link-layer, the local link set is used instead of the HELLO message exchange[33].

2.6.2 Packet format and forwarding

The packet specifications are outlined using a packet format and forwarding an optimized flooding appliance serves as a transport mechanism for all OLSR control traffic. All communication relating to OLSR is coordinated by the packet format. This is done for the extensibility of the protocol and to maintain compatibility. It provides storage backing for various types of information into one transmission. All the messages share a common header format so that nodes can easily identify and accept them. The messages are flooded in the entire network but duplicate retransmissions are discarded by MPRs [34].

2.6.3 Topology Control Message

Topology control (TC) messages are sent out to provide each node within the network adequate link-state information for route calculation purposes.

2.6.4 Neighbour detection

A node can intercept the neighbour set directly from the information exchanged during the process of link sensing. A single node will have one main address but with multiple nodes, additional information is required to orderly map the addresses of the interfaces. The additional information is obtained using multiple interface declaration (MID) messages.

2.6.5 Route calculation

The computation of the routing table for each node is made possible using link-state information that is obtained through periodic message exchange and the interface configuration of nodes. Each node uses a routing table for channeling of data to other node destinations. The routing table is computed using information from the local link information base and the topology set. If the information in these repositories changes then the routing table is recalculated and updates the route information about each destination since the nodes are mobile. The format of recording entries in the routing table is in the following format:

1.	*R_dest_addr*	*R_next_addr*	*R_dist*	*R_inface_addr*
2.	"	"	"	"

Whereby *R_dest_addr* is the route destination address, *R_next_addr* is the next-hop node to the *R_dest_addr* and it has to be an asymmetric neighbour reachable through a local address *R_inface_addr*. *R_inface_addr* is the route interface address and *R_dist* is the route destination. Broken or partially known routes to the destination are discarded in the routing table.

2.7 Zone Routing Protocol

Zone Routing Protocol (ZRP) is a hybrid routing protocol [35]. It uses proactive methods to get route information of its nodes but uses reactive methods to get nodes to the destination node outside the routing zone. The routing zone is more or less of a section within the network where all the nodes that are within a distance less than or equal to K hops are within reach. It is mostly used in the recent integrations of Vehicular *Ad-hoc* Networks (VANETs) and the internet of things (IoT). ZRP is mostly implemented for wide-scale networks because of its structural design that combines

proactive and reactive routing approaches hence it is subdivided into two routing protocols namely Intra-zone Routing Protocol (IARP) and Inter-zone Routing Protocol (IERP).

2.7.1 Intra-zone Routing Protocol IARP

IARP is a proactive routing protocol [36] that is based on periodic maintenance and constant updating of routing tables. Route requests handle route queries through the zone border nodes and not by flooding mechanisms like OLSR. Each node in the network uses HELLO messages to gather information about all neighbours within its vicinity (routing zone). These HELLO messages die after one hop and that is after reaching the neighbours. Each node is responsible for managing the routing table so that it can utilize any node in its routing zone for sending out a message called Zone Notification Message (ZNM). Each node that will receive the ZNM and the ZNM itself dies after K hops. Every node in the network that receives this message decreases the hop count by 1 and forwards the message to its neighbouring nodes until the value of K is 0 [37].

2.7.2 Inter-Zone Routing Protocol

The IERP uses reactive mechanisms [36] to find routes between nodes located at distances further than the zone radius. It sends request messages to border nodes every time no route is found in the current zone. This is called the Border-casting method. At first, the source nodes verify whether the destination node is within the current routing zone. If this is true and the destination is known, then the route discovering process is halted. If the destination node is outside the routing zone then the source node forwards a route request to all peripheral nodes [37, 38].

2.8 Quality of Service

The increasing popularity of multimedia applications and the commercial deployment of MANETs has made QoS support an important requirement. QoS support in MANETs is a bit different from that of wired networks as it involves a shared bandwidth with neighbour nodes and they operate in a dynamic topology. This leads to the scramble for nodes to establish reliable routes and procure relevant resources to provide a level of acceptable QoS.

2.8.1 QoS support for MANETS

A lot of work has been done on QoS in wired and wireless systems but it always seems doubtful if any of those proposed mechanisms can equally be applied to MANETs due to their limitations and MANET-specific nature. The support for QoS [39]constantly requires link-state information to be forever present i.e. bandwidth, loss rate, error rate and delay. Supporting the mentioned requirements is often a huge task as the quality of the *ad-hoc* link can abruptly change because nodes are mobile and operate in dynamic environments. Earlier approaches to QoS were based on the virtual circuit model which meant that there should be fixed connection management before communication so that for the duration of that session there would be a guarantee of route and reservation between the source node and the destination node. This was a great approach but it lacked the flexibility needed to dynamically adapt to mobile *ad-hoc* networks where the path and reservation to respond dynamically and in real-time to the ever-changing topology and resource needs was necessary [40].

One of the threats to QoS support is the maintenance of an acceptable service level. By default, QoS support in MANETs is linked to the routing protocol's performance because the flow between the source node to the destination is not always straightforward. There is a high chance that within the lifetime of an ongoing session, the flow gets rerouted and this happens frequently. When this happens, there is also a change in resource provisioning and needs as the flow would be now on a new path. The previous QoS agreement from the initial path may not be valid and so would be the assumption that the route and reservation remain fixed in that duration of a session. This is the reason why many researchers have constantly been coming up with mechanisms and frameworks that can support QoS in MANETs adaptively and can respond to its intrinsic behaviour [39].

2.8.2 QoS metrics

QoS is measured using specific metrics within the network. Some of the evaluated metrics include throughput, delay, jitter, bandwidth and packet loss. Below are some of the few metrics discussed in the literature:

2.8.2.1 Jitter

This is the variation in the end-to-end delay in a network. In most cases, professionals refer to jitter as delay variation. For packet-switched networks, jitter is the distortion in packet arrival times as compared to packet transmission times. It is the time it takes for packets to get from one node to another node during the transmission period.

2.8.2.2 Throughput

It is the total number of routing packets received by the destination node over the total simulation time [15]. In MANETs, the network throughput is measured as a rate of delivered packets over a wireless link or through neighbour nodes.

2.8.2.3 Average End-to-End Delay

This is the average as a measure of how many cycles it takes for the data packets to arrive at their intended destination [15]. The higher the delay, the more insensitive feedback loops that render the protocol less sensitive to short-term dynamic changes in the network. Delay can be analyzed either at each node or on the link between source and destination.

2.8.3 QoS and Trust

Trust and QoS have almost similar metrics to evaluate their efficiency. There have been different approaches to define trust. Trust [18], in general, is meant to be a bi-directional relationship between two entities and plays a huge role in the formation of a relationship with the network. Even though the trust has been formalized as a computational model, it is dynamically applied in the different research communities. QoS [8] in MANETs is a globally growing area of interest. Due to the rapid expansion of multimedia technology, mobile technology and real- time applications like video streaming are strongly compelled to support QoS, and metrics considered are usually throughput, delay, energy consumption, jitter, etc.

Trust and QoS [41]share similar metrics e.g. delay throughput and packet dropping rate. The correlation between these concepts exists not only through shared performance metrics but also through the assurance of good service in the network for the end-user.

QoS and Trust are what stand between the network and the Applications/Users. It is not easy to provide QoS support without having the right QoS requirements. A particular level of trust must be established and that is, only trustworthy nodes that perform as desired will participate. Trust is dynamic [17] just like MANETs which have an ever-changing topology state. This means that the trust value should be based on temporary and yet local information. In addition to this, the trust value is different for similar nodes. This is influenced by the fact that each node goes through different situations in terms of the dynamic topology. Trust is evaluated mostly through a not appropriate amount whilst QoS is of a quantitative measure and is of a precise value. As such, the two have one thing in common and that is to rate performance in networks.

2. 8.4 QoS and video streaming applications

Video streaming is a client- server paradigm that comprises interactive media data of streaming applications such as video- on- demand, video conferencing, e-learning, and many more as depicted in Figure 2.7. End-to-end QoS [6] provision to video-streaming applications over MANETs brings a complex and challenging problem. This is because video streaming is considered to be delay-sensitive and also bandwidth-intensive. To make video streams smaller, they are compressed using video coding formats. Examples of such formats include HEVC, VP8/9, or even H.264. The video streams are then assembled in a container (bit-streams) such as FLV and MP4. The bitstream is transmitted from stream serving to the client [42].

2.8.5 QoS and security

The biggest challenge in addressing issues of security in MANETs is that most proposed security solutions put a strain on the available MANET resources i.e. battery power [23], bandwidth all because they bring heavy traffic load to exchange and verification keys. The design solutions to most of the security issues need to meet or address the resource aspect of the problem. The first step before designing a possible solution for an attack is understanding the form of attack. The QoS and performance get affected whenever an attacker takes control of the network. MANETs are mainly used in extremely important professional fields i.e. military battlefield, medical service, and sensor networks.

2.9 Security goals in MANETs

The goal of every system is to achieve excellent QoS and adherence to security targets to protect client data or information [36, 43].

2.9.1 Availability

Availability is one security target for every system. The authorized user [38] will request the use of the node and in an operable state. The node is therefore supposed to provide its services following its design. Attacks may seek to disrupt the node's operation and also use up some of the node's resources but the node must be able to survive those attacks and be available when requested. Denial of Service (DoS) is one attack that disrupts the availability of a node.

2.9.2 Confidentiality

This security feature ensures the unavailability of certain features to unauthorized entities or users. The information is only made available [44] [45] to authorized personnel. A message that was encrypted at the source will only be decrypted at the destination node. Many cryptographic attacks may try to reveal the message contents. An ideal system will be able to protect the contents of such information from unauthorized users.

2.9.3 Integrity

Integrity is a security feature that ensures that the original contents of the data are maintained and not altered in any way. An attempt to intercept the data being transmitted, either by human beings or malicious nodes is an act against the integrity of the system. Dropping attacks can be launched by a malicious node but the node is compelled to cooperate by the system [44].

2.9.4 Authentication

Authentication ensures user validation and avoiding impersonation. The malicious node could impersonate a legitimate node by using the node's MAC address or even an IP address to obtain authentication and also launch its attack at a higher level.

2.9.5 Anonymity

This feature is for privatizing the true identity of a node to ensure privacy and confidentiality. In most cases, the source of packets is kept private.

2.10 QoS-aware Routing Protocols

To be in a position to address the MANET-IoT relationship, we must first understand the role of QoS-aware routing protocols. QoS-aware protocols are best described according to design mechanisms. A QoS-aware routing protocol has fewer constraints in terms of:

a) Route maintenance
b) Route discovery
c) Bandwidth
d) Route Reservation
e) Route Selection

2.11 Secure routing in IoT and MANETs

MANETs are part of the IoT paradigm as they are now being integrated with WSN for reliable and secure communication [46]. MANETS and WSN are seen to be key technologies in the success of Smart cities because of their self-configuring abilities that enable the deployment of many nodes. MANETs and WSNs show that municipalities will soon adapt them for various reasons e.g. localization, environmental monitoring, and public safety. The Republic of North Korea has invested in such technologies to maximize the wide range of services it offers to its citizens from environmental monitoring to traffic management and also entertainment [47].

2.12 MANET security challenges/vulnerabilities

MANETs are prone to various [48] security vulnerabilities that seek to gain unauthorized access to the user's data. Vulnerabilities of a system may be termed its weaknesses. MANETs are more vulnerable because they rely on wireless technology, unlike traditional wired networks. Below are a few vulnerabilities according to [38].

2.12.1 Availability

The availability of the network is one challenge in MANETs. Availability, in essence, refers to the accessibility of network resources or assets to authorized nodes at predefined times. The dynamic nature of MANETs together with security attacks i.e Blackhole can pose a great threat to the availability of the network [48].

2.12.2 Central Controller

The lack of a centralized controller that could act as a monitoring server is one vulnerability that comes with MANETs. This makes it complex in terms of security provision against attacks as in most instances the network environment is huge and highly dynamic[38].

2.12.2 Dynamic Topology

As mentioned earlier, the network environment in MANETs is dynamic. This may in turn affect the trust relationship among nodes. Malicious nodes that may be compromised within the network are also difficult to spot as the nodes are mobile. Adaptive security mechanisms need to be implemented to improve performance areas of MANETs.

2.12.3 Power limits

Mobile nodes rely solely on battery power and such may pose so many problems. A node may behave maliciously within the network and may be suspected of being an internal attacker but only to discover that it behaves selfishly because of a limited power supply [38].

2.12.4 Resource availability

Resource constraints are the primary reason why some services are not utilized in MANETs. For example, secure communication is needed but most often it is difficult to provide because of the dynamic environment. *Ad-hoc* security mechanisms and architectures are needed to prevent attacks from flooding the network [44].

2.13 Security attacks in MANETs

Attacks in MANETs can be categorically put in two; namely passive and active attacks. The severity of these attacks differ. Wireless networks are most vulnerable to all sorts of attacks internally and externally as compared to [49] traditional networks (wired networks) due to limited physical security, mobile nodes, lack of centralized management, dynamic topology, scalability, and threats emanating from compromised nodes within the network. The vulnerability of the network highlights a weakness in the security architecture or system. Security is very important in mobile *Ad-hoc* networks even though the hostility in the environments they operate in makes it difficult to achieve most security properties as proposed by many authors [43].

2.13.1 Active Attacks

Active attacks are known to alter the normal operation of a network [23]. The attacker actively participates in the interruption of the network's normal operation. The attacker acts as one of the nodes in the network. In this way, it can exploit any other node and uses it to its advantage. It can feed nodes malicious packets or even denial of service (DoS). The active attacker can:

a) Fabricate messages
b) Replay packets
c) Modify packets
d) Drop packets
e) Node Impersonate
f) Insert infected code

Active attacks can be either external or internal. External attacks are normally carried out by nodes that do not belong to the existing network. Internal attacks are carried out through malicious nodes within the network [50]. The severity of the attack is greater from internal attacks than from external attacks. This is because the attacker is already part of the existing network.

2.13.2 Passive Attacks

Passive attacks are characterized by their inability to actively participate in causing harm to the network [23]. The attacker listens to the network to obtain information. They look at what is going

on in the network. They do not just attack the network. At first, they acquire enough information before launching an attack. Once they acquire information, they easily hijack it and launch an attack. They can decrypt weakly encrypted data, acquire passwords, private and public keys, monitor communication routes, and message flow among entities [50]. It may be hard for the user to identify a passive attack as it does not necessarily alter anything regarding user data or traffic.

2.14 MANET-specific threats

2.14.1 Black Hole Attack

A blackhole attack uses an existing node within the network to maliciously invade the network. It uses its routing protocol to falsely advertise itself as having the shortest path to the destination node [50] towards the node it wants to invade. The compromised node falsely advertises the availability of new routes to the rest of the nodes in the network without verification of its routing table. The attacking node [51] is then able to indicate the route availability in response to all route request messages. As shown in Figure 2.6, the attacker is then able to obtain data packets and retain them. In protocols that involve flooding, the attacker's reply will be received by the requesting node first before the reply coming from the actual replying node and in that way, a malicious and misleading route is created. When this route is formed it is then up to the node if it keeps or forwards the packets to an unknown destination of its choice [49].

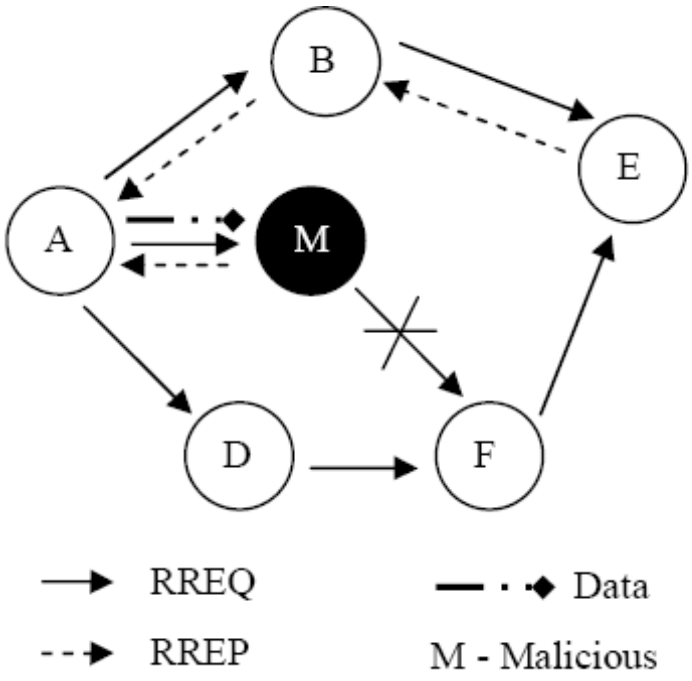

Figure 2.6: Blackhole attack in a MANET [49]

2.14.2 Wormhole Attack

On the other hand, in a wormhole attack, the malicious node records and then tunnels the packets in another parallel location. Many authors have tried security mechanisms to secure MANETs through encryption and node location information. The main challenge in MANETs [49] is a key distribution because there is no centralized controller.

In usual cases, remote regions of malicious nodes become connected using a high-speed link and they may appear to be neighbour nodes but in essence malicious. This attack is often referred to as a tunnelling attack.

2.14.3 Selfish Behavior

This is when a node refuses to relay messages to other nodes to extend its own battery life or to use the channel alone while other nodes remain dormant. The packets carried by selfish nodes are then dropped by that particular node. Existing routing protocols have no mechanism to detect these nodes or predict their behaviour [43].

2.14.4 Gray Hole attack

The Gray Hole attack [43] is almost similar to a black hole but the only difference is that instead of dropping all the packets, it selects packets from one particular node and drops them. It causes a denial of service (DoS) attack against a specified node or group of nodes [45].

2.14.5 Route Error generation

The malicious node fabricates and then sends out route errors to other nodes. It broadcasts messages in the network through multiple routes between other nodes.

2.14.6 Eavesdropping

This is when a node simply observes confidential information. This may be information that may compromise the integrity of that organization [38] e.g. passwords, location, a public key, and critical content. As shown in Figure 2.7, the eavesdropper simply taps the line/link between the source and destination and thus classified information gets exposed [45].

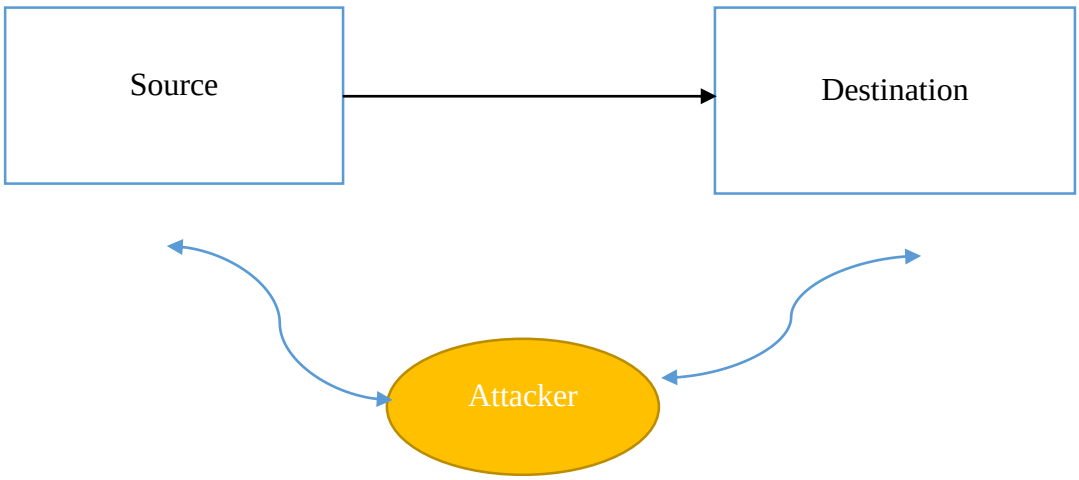

Figure 2.7: Eavesdropping[38]

2.14.7 Blackmail attack

The malicious node sends out broadcast messages reporting that a legitimate node is malicious. It does this for the rest of the network nodes to exclude the legitimate node from the routing tables.

In this way, the legitimate node finds itself isolated as MANETs keep a blacklist for all malicious nodes.

2.14.8 Sinkhole attack

In this attack, the malicious node fabricates and advertises to have an optimal route to the destination to attract traffic from neighbouring nodes. Through the fake [51] broadcasting, the sinkhole node can obtain all the traffic to itself. After receiving the traffic, it alters then drops the data packets. It also increases the energy consumption of the network and destroys it. MPRs are responsible for forwarding packets in the network. In OLSR, a sinkhole [52] attack can disguise itself as an MPR to attract traffic then selectively drop packets. They advertise a false shortest path to the destination node which causes the neighbouring nodes to send packets through the malicious node.

2.14.8.1 Sinkhole attack in Dynamic Source Routing protocol

DSR uses two kinds of messages known as Route Request (RREQ) and Route Reply (RREP) for its route discovery process. The RREQ messages are used only when there is no route available on the route cache to reach the destination node. In this particular instance, DSR initiates the route discovery process by sending an RREQ packet that will be uniquely identified by the source ID, destination ID, and sequence number. A bogus RREQ is usually sent out to carry out a sinkhole attack. If the bogus RREQ has a sequence number higher than the originally sent RREQ from the target node, the other participating nodes in the network will treat the bogus RREQ as the latest request and discard the original one [53].

2.14.8.2 Sinkhole attack in *Ad-hoc* On-Demand Distance Vector protocol

In AODV, the sinkhole attack modifies the sequence number in the RREQ so that it is always higher. If the sequence number is higher, the general assumption is that it contains recent routing information. The malicious node [51] then selects the source and destination pair. The malicious node then observes the source node's sequence number carefully and then generates a bogus RREQ with a higher sequence number than the sequence number of the source node. It then broadcasts the bogus RREQ and nodes in the network assume that the bogus RREQ could be better and is a fresh route than the source node and other routes.

2.14.9 Traffic analysis

The attacking node [38] senses and intercepts the communication path between the source and destination and examining the messages to monitor the patterns in communication. In this way, it can detect the amount of data being transmitted between the sender and receiver. As shown in Figure 2.8, the malicious node only analyzes the network and does not practically participate in the actual network. This means that both encrypted and decrypted data can also be decrypted [49].

2.15 Security Mechanisms

There are existing architectures to combat attacks in MANETs. These include encryption, digital signatures, authentication and access control. Secondary mechanisms include intrusion detection systems (IDS) to monitor route behaviour and to ensure that cooperation amongst nodes is adhered to [38]. They provide a high-security detail. We can safely divide them into preventative mechanisms and reactive mechanisms.

2.15.1 Preventive Mechanisms

When an attack is launched against any system [38], the first line of defence is always a preventative measure to try to counteract that attack. When the network protects itself against physical attacks then such an act is a preventive mechanism. Cryptography, which is either symmetric or asymmetric, is one preventative mechanism. In symmetric cryptography, the message is encrypted and decrypted by the same key from both the sending and receiving end. In asymmetric cryptography, different keys are used (public and private keys). If a private key is used for encryption, then a public key is used for the decryption of the message. If a public key is used for encryption, then a private key is used for decryption. This all happens at the authentication phase. Cryptography has a few weaknesses though most especially the issue of secret keys in an insecure communication channel. If the secret key is not secure the communication is compromised. To authenticate public keys in asymmetric cryptography, digital certificates are used. Digital certificates show the owner of the public key and they are created and signed by a trusted certificate authority (CA).

The trusted CA then verifies the legitimacy of the certificate if it is requested by other users. Data signatures are used to attain data integrity and authentication. They are used to achieve privacy,

non-repudiation, and authentication. Preventive measures alone cannot fully halt most attacks or make the network more secure. Most attacks will still bypass all these mechanisms and cause harm to the entire network. This calls for a second line of defence.

2.15.2 Reactive Mechanisms

This is the second line of defence for MANETs in a case of intrusion and attempts to compromise the security detail of the network. Intrusion detection systems (IDS) are systems specifically designed to detect intrusions through the monitoring and analysis of data. Most of the intrusion detection systems designed for wired networks cannot be implemented the same way as those for *Ad-hoc* wireless networks as these networks have special characteristics like limited resources and topology changes. This research work uses both defence mechanisms to protect the network from malicious activities. Encryption is implemented for packet delivery and an intruder detection system as a second line of defence [38].

2.16 Intrusion Detection Systems (IDS)

An intrusion in the context of computer science can be categorized as an unauthorized form of activity to cause damage to an information system. The intrusion ultimately means that data confidentiality, availability, and integrity will be compromised during the intrusion. An intrusion detection system, however, is software or hardware [54] responsible for detecting various malicious traffic that cannot be detected by a firewall. It is through an IDS that availability, confidentiality, and integrity can be achieved. The Intrusion Detection System can further be dived into two categories: Anomaly-based Intrusion Detection System and Signature-based Intrusion Detection System [55].

2.16.1 Anomaly-based Intrusion Detection System (AIDS)

This kind of IDS uses machine learning, knowledge-based or statistical methods to model the behaviour [37] of a normal functioning computer system. If the behaviour of the system deviates from the modelled behaviour then this is interpreted as an intrusion. The general assumption, in this case, is that malicious behaviour differs from that of a typical user. AIDS operates in two phases namely the training phase and the testing phase. During the training phase, a normal traffic profile is used to learn about the model of normal behaviour, and then during the testing phase, a

new data set is implemented to check the system's ability to detect previously unseen intrusions [54]. One advantage AIDS has is that it can detect zero-day intrusions because it does not rely on signature databases to trigger a danger signal [56].

2.16.2 Signature-based intrusion detection system (SIDS)

Signature-based intrusion detection systems use pattern matching techniques to detect malicious activity. SIDS uses signature matches that already exist in their database to locate an attack. The attack should have previously been recorded before an alarm is triggered. It gives a good detection accuracy for known intrusions but zero-day intrusions would succeed because there is no matching signature that exists in the database until the damage is already done. In modern times, polymorphic variants of malicious attacks have made SIDS less progressive and have further undermined its adequacy in detecting intrusions. The database of signatures needs to constantly be updated. SIDS stands superior to AIDS when it comes to handling known attacks [56].

2.17 Security Frameworks related to the research

A Cross-layer Design (CLD) approach was proposed by authors in [57]. Their proposed strategy was to use Bit Error Rate (BER) in the execution of OLSR. BER is a mix between the association accessibility and breaking point. To upgrade the traditional way of operation in Multi-Point Relays and route computation, they propose using heuristic estimation. This is a technique used to find the propelled route and using the shortest path to produce good-enough solutions in a short space of time. This is, however, an inbound system that already exists in OLSR. It did not consider functions like connection admission control, reservation signalling, and stream classifier which are considered essential information in QoS. This research work incorporates the mentioned functions based on the critical role they play in QoS provisioning.

A trust-based QoS model was proposed by authors in [58]. The model estimated the trust degree between nodes using direct and indirect trust methods. The direct trust was calculated using observations and the indirect trust involved neighbours' recommendations taking into account link delay as the QoS constraint requirement. A contribution was made whereby the authors designed a trust-based QoS algorithm called TQR based on the relationship between link delay and trust degree. The simulations were conducted on NS-2 and the implementation of the algorithm was

based on AODV. A performance comparison between AODV, Watchdog-DSR and QAODV was made and the results showed that the TQR scheme can prevent malicious nodes and improve the performance of the network in terms of end-to-end delay, packet delivery ratio, detection ratio of malicious nodes and routing packet overhead. The authors evaluated the algorithm on reactive protocols which generally have a lower overhead as compared to proactive protocols [59, 60]. The proposed model may be exposed to slander and collusion attacks. A slander attack involves the sending of incorrect recommendations to destroy the reputation of other nodes. The malicious nodes can work in collusion to strengthen the severity of the attack. During the evaluation process, the authors assumed that the attacks do not affect the trust model. This assumption weakens the proposed model because in a practical scenario there are many security vulnerabilities within MANETs that could end up affecting the entire network performance. This work proposes an intruder detection system that detects and removes attacks during the routing phase to improve the QoS of the network. The assumption that there are no known attacks cannot be justified hence a sinkhole attack is simulated and its effect on the network performance is evaluated.

The authors in [61] proposed a trust detection algorithm to increase the detection rate of Black Hole nodes in MANETs and improve the QoS. The proposed framework was modelled to observe the node behaviour of each node using trust metrics. The trust metrics employed include the relationship between sensor nodes, QoS metric trusts, social and service attribute trusts. The mobility behaviour and communication of each node were used to detect the behaviour of the nodes. The ZRP protocol was used for routing and the method was able to avoid black hole nodes. The method was evaluated for different scenarios including those that have trust and those without. The results showed an improved performance in terms of throughput, energy consumption, false acceptance rate, missed detection rate and packet loss. The authors presented social trust in the context of MANETs in their model integration in a probabilistic manner. The integration of different kinds of trust from the authors' perspective provides full immunity to attacks because of the computed risk calculations. This assumption cannot be entirely validated because MANET nodes are constantly moving and the trust levels cannot be fully guaranteed by the network. This work is not based on estimated trust values but on a known security attack that decreases the performance of the network. The model proposed by the authors provides a holistic approach to the realization of trust in a MANET but does not fully cover the security aspect in detail. This

research work is not entirely based on the network level but also considers the cross-layer communication from the other levels like the MAC layer.

A technique for attack analysis and a QoS framework for OLSR was proposed by authors in [62]. They used soft computing techniques to select nodes to provide accurate means of data transmission. The nodes are initially clustered using the Hybrid K-mean algorithm then a path is formed using OLSR. The nodes are optimized during the routing phase using the Hybrid Genetic algorithm. The data transmission was found to be occurring with maximum cluster head selection and data gathering capacity as well as with the shortest path selection in reaching the sink node. The cluster head selection information was gathered from all the nodes because their method did not have cluster head selection. They considered this under their future work as part of optimizing the framework. The authors did not provide a counterattack mechanism but just an analysis. When comparing their proposed technique to others they found it to be effective and provided better results. The implementation was done in NS-3 and results obtained from the performance evaluation only validate some of the conducted studies whereby OLSR outperforms other routing protocols in high node density environments but does not validate the proposed method. This study analyzed the severity of the sinkhole attack and provided a counterattack mechanism that clearly shows an improved QoS when compared to other scenarios using NetSim.

The authors in [63] proposed a secure and proficient QoS framework to prevent various threats and attacks like DoS, replay attack, flooding attack, information disclosure, QoS degradation, over-reservation, over/under-reporting of available bandwidth and stable table starvation. The authors also emphasized the importance of striking an agreement between security and QoS as both impact each other. Their work uses simple rather than complex methods that exist in security. The framework was evaluated in comparison with other existing frameworks like INSIGNIA, PRTMAC, and INORA for throughput, packet delivery ratio, end-to-end delay and jitter where it showed better performance. The authors argue that there are four cross-layer mechanisms found in literature and they do not provide a security mechanism while security is a critical aspect of QoS in MANETs. There is no clear evidence of why the authors evaluate frameworks without security against the one they propose as this does not give a fair comparison. Security attacks have different ways of invading the network and do not have a similar impact on the severity of the attack. It can be argued that if the proposed framework truly represents most of the mentioned attacks, for

example, a black hole attack and a DoS attack are different. The authors also have not provided a counterattack mechanism or even evidence of a specific existing attack in the network. Our study closes that gap by providing a sinkhole attack and a counterattack strategy that also improves QoS delivery.

The authors in [64] proposed a secure ANFIS based QoS multicast routing agent (Secure AQMRA), an extension of the existing protocol AQMRA. The improved protocol aimed to optimize QoS elements of residual power, packet loss, bandwidth, link delay, speed and link utilization for better scalability and reliability. Secure agents were configured for every node in the network. The proposed protocol outperforms the existing one in terms of end-to-end delay and packet delivery ratio. Their proposed framework, however, does not address the aspect of security but rather focuses mostly on optimizing specific elements for better QoS. There is no evidence of any specific attack in the network nor if the implemented algorithms address any specific passive security attack. Our study focuses on both QoS and security respectively.

An energy-efficient *ad-hoc* on-demand routing (EEAODR) algorithm was proposed by authors in [65]. The algorithm focused on used AODV and was designed to improve breakages and steering concerts regarding size reduction and infinitely monitoring routing. It employed two strategies namely TEAODV and MOAODR. TEAODV method was designed to supports unicast and communication. MOAODR was designed to enhance multicast correspondence, reducing overheads, lessening the flooding, and reducing the rate of association breakages. The proposed protocol, cooperation, and reputation in packet delivery were achieved. Their work was centralized to energy efficiency and AODV. This research focuses on OLSR and QoS efficiency in a cross-layer approach. Their focus is not on the other layers of the network but just on the routing protocol.

A novel secure architecture was proposed by authors in [66]. They proposed an access control mechanism that produces a secure address auto-configuration scheme by utilizing identity cryptography and threshold secret sharing concepts for MANET nodes. Their mechanism provided a key management service without any predetermined trust relationship among nodes which effectively addresses the challenge of single-point failure in the traditional public key infrastructure. The proposed mechanism improved the QoS level as compared to existing approaches that try to resolves issues around auto-configuration and access control separately. The two services in their research were combined to make a single comprehensive architecture using

similar security primitives but minimal computation and communication overhead as compared to other existing approaches which at most times have an increased cost because of the different services needed for MANET users. The simulation results gave low membership renewal time and admission time in terms of access control. The overhead was also lowered in comparison to other existing schemes. Their work was evaluated on an NS-2 testbed using the AODV routing protocol in a low node density environment. The authors presented a security framework that does not strike a balance between security and QoS. The focus was mainly on the security aspect of things instead of QoS even though it performed better than other existing schemes. The authors argue that the framework is solely based on cryptographic techniques to control the nodes' access and obtaining a dynamic unique address. The assumption made is that the attackers have bounded computing capabilities. This means that if an attacker were to infiltrate the network and send packets with the goal of depleting network resources then such an attack would be possible as the nodes would spend some resources trying to verify the source of the packets and by this, the attacker would have been successful in launching a local resource consumption attack to the proposed system. The proposed framework was not evaluated for other routing protocols like OLSR which generally has a higher overhead. Our study is evaluated in low, mid, and high node density environments using the OLSR routing protocol.

An ant colony system (ACS)-based algorithm and a situational concept were proposed by authors in [67] to develop a situation-aware multi-constrained QoS (SAMQ) routing algorithm for VAETs. They aimed at computing feasible routes among vehicles subject to multiple QoS constraints and then choose the best-computed route. They did not rule out the possibility of having the best route ending up failing at any moment and to mitigate such a risk they utilized the situational awareness levels and ACS mechanisms to prepare for countermeasures to assure a reliable transmission of data. The simulation results of the proposed scheme demonstrated superior levels of reliable data transmission as compared to existing QoS routing algorithms in a highly dynamic network topology. The simulations were run on the OMNET++4.6 network simulator for 20 runs and an average of the runs was obtained. The investigation contributed to the study of VANETs as the simulations were conducted in a highly dynamic and constrained environment but their scheme provided continuous and stable data transmission. They did not, however, consider the possibility of any malicious activity challenging the effectiveness of the scheme hence the absence of a security mechanism. The authors did not rule out the possibility of data transmission being

interrupted due to a low reception rate or link breakage. When the breakage occurs, the intermediate node issues a SARE message to the precursor node to switch to a backup route at this point, the data transmission would have been interrupted because of an unreachable destination until a new route is established. The results also showed an improved ratio of packet delivery to dropped packets. The breakage of links could affect the end-to-end delay for video streaming applications. This work utilizes video streaming as the preferred application and it is delay-sensitive.

Pushpalatha *et al.* [68] presented a novel framework disruption tolerant secure opportunistic routing (DTSOR) method that ensures reliable and secure communication during emergencies in high mobility nodes. The authors used cluster-based address autoconfiguration and trace analysis as methods to select an optimal path for forwarding packets. The selected optimal path would then further ensure security by using a puzzle-based data anonymization method. The proposed framework outperformed the state of the art approaches in terms of throughput, network overhead and packet delivery ratio. The results also show that data anonymity using a context-based puzzle greatly affects the performance without much effort. There is insufficient evidence to prove that the framework is effective towards attacks that are already in the network but its design does address the communication aspect between nodes. Our research work is evaluated for sinkhole attacks whereby the attack is initiated in the network to get a realistic result instead of assuming the attack does not exist or rather assuming the framework is generalized for all attacks as the authors have proposed in their study.

Tripathy *et al.* [46] proposed a unified framework for MANETs and WSN in the IoT environment for efficient and secure communication between intelligent nodes. Their framework is comprised of different phases such as the collection of data using sensors then sending it to data processing centres using mobile access points and lastly forwarding the data to data centres for analysis, decision-making, and reporting purposes. The study was evaluated on NS-2 and achieved efficient throughput on MANET-WSN integration using three ZigBee devices and there was more packet loss when the node was out of range and a new path is established with the remaining nodes. On an overall scale, the integration mechanism proved efficient in managing communication in the smart environment. This framework only addresses the communication and integration aspect of the networks and nothing on security. The integration of the two systems makes it difficult to come

up with appropriate metrics as the authors used traditional metrics like throughput and delay. The integration also opened room for the design of specific routing protocols that are more applicable to diversified applications. Alternatively, the existing routing protocols could be extended and tested for efficacy.

The authors in [69] proposed an Automatic Neuro-Fuzzy Inference System (ANFIS) and the Kalman Filter are used to optimize the learning and discovery of the dynamic QoS parametric values intelligently and automatically. The hybrid approach was implemented to automate and provide better QoS. The Neuro-Fuzzy approach gave speed and intelligence to MANETs and the Kalman filtration's multi-heuristic and stochastic approach provided the peak optimality. The parameter values were checked and analyzed during simulations conducted and the overall bandwidth consumption had improved, queue size decreased in all scenarios and QoS increased. Their work assumes the network to be free from attacks and thus no security architecture design to tackle any form of passive or active attack that may affect the QoS delivery. The work focuses on speed and intelligence rather than security to optimize QoS. This study focuses on security to optimize QoS.

The authors in [70] proposed a novel algorithm called DMN (Detection of Malicious Nodes in VANETs) to isolate nodes showing malicious behaviour and also enhancing network performance. The proposed algorithm is an extension of the DMV algorithm. DMN was designed to optimize the selection of the DMV, an existing algorithm, which selects nodes as verifiers with a distrust value less than the monitored vehicle. The verifiers are the load, distrust value and distance. The decision value is then evaluated and compared to the verifier selection threshold and thus the optimal verifier improves network utilization and performance. The results indicated a higher throughput, a reduced end-to-end delay, and a better packet delivery ratio as compared to the DMV algorithm. The authors suggested that the work could further be developed or extended to accommodate a technique for malicious nodes and also deployment of real-time traffic to test and analyze performance under realistic conditions. Our research work is based on real-time traffic and covers the two suggested future implementations.

The authors in [71] proposed a novel cross-layer cooperative scheme for detecting blackhole attacks targeting the QoS of OLSR in VANETs. They implement this using QoS-OLSR, a protocol that relies mainly on MPRs that are responsible for all routing in the network. Blackhole attacks

target MPRs where packets are intentionally dropped to cause DoS. The authors proposed two mechanisms that allow the exchange of information between layers. The first scheme utilized information among network and physical layers while the second scheme was linked to the physical, MAC and network layers to enable reliable and efficient detection. In the physical layer, the detection technique assigned each legitimate user a signature that is multiplied by the message, and each node used a maximum likelihood approach to detect the legitimacy of the message. In the MAC layer, the detection technique monitored the number of RTS/CTS (Request to send/clear to send) requests among neighbour nodes. The cooperative watchdog technique is implemented at the network layer to monitor the exchanged packets exchanged among neighbours. The cross-layer technique gave enhanced detection results and minimized the false alarm rate as compared to other state-of-the-art detection schemes. The proposed scheme was implemented for blackhole attacks and there is no evidence of its effectiveness on other attacks in MANETs. Our study implements an intruder detection system that includes a Pathrater and a Watchdog between MAC and network layers. At the application layer, the AES algorithm is implemented as an additional security measure.

The authors in [72] proposed an Enhanced (Medium Access Control) MAC with Delay Aware Multipath Routing (EMAC-DAMR) Mechanism to enhance QoS. The enhancements were facilitated through the MAC in terms of packet transmission delay, remaining time and usage of the Erling factor. NS-3 was used as a testbed and the evaluated metrics include packet delivery ratio, throughput, packet drop, and overhead. EMAC-DAMR was compared with other solutions and it performed better in terms of throughput and packet delivery ratio. It also reduced the packet delay, overhead packets and the number of packets dropped. The authors made major contributions to the QoS study in AODV. Other routing protocols like OLSR and DSR were not evaluated by the framework.

The authors in [73] proposed the Secure Optimized Link State Routing (SOLSR) protocol to prevent security attacks and also find an optimal path from source to destination by integrating a message authentication code that acts as a global secret key in both Topology Control (TC) Messages and HELLO messages. Their study suggests that the timestamp mechanism is used to avoid replay attacks because authentication and encryption cannot prevent replay attacks. This is the reason why our study does not only rely on AES but also involves an intruder detection system

to avoid sinkhole attacks. Their proposed SOLSR was compared with the traditional OLSR protocol on NS-2 over different network scenarios and it achieved better results in terms of data overhead, packet loss and delay. Their approach to QoS was not on a cross-layered approach and there is no evidence of the framework's effectiveness to other forms of attacks except replay attacks. The simulation utilized video as an application hence the authors used both UDP and TCP. Video streaming has complex requirements in terms of transport protocol as retransmission may not always be ideal for streams.

The authors in [74] presented a QoS Management in Mobile *Ad-hoc* Networks (QoSMMANET) framework to offer QoS in high mobility environments. The elements of the proposed framework included a QoS management protocol, a queuing discipline, and a polynomial probabilistic flooding protocol. The flooding protocol was responsible for minimizing congestion and increasing network coverage during high node mobility conditions. The queuing discipline provided different class services by providing packet prioritization and packet differentiation. The QoS management protocol was responsible for balancing traffic flows and avoiding bottlenecks. The experiments were conducted on an NS-2 simulator with multiple-source multicasting in low and high mobility scenarios. The results showed a reasonable packet delivery ratio in high mobility and the best performance in terms of collisions and packet duplicates. The drawback of the framework was with the deadline miss ratio being lower. The conclusion was that the framework only offers soft real-time support and not hard real-time support and that in the future it could be extended to address issues like security, load balancing, battery consumption, jitter, and scalability. This study does not focus on mobility but the density of the nodes and it also addresses security in the network.

Jhaveri *et al.* in [75] proposed an extension to their previously registered framework called a trusted routing scheme with pattern discovery (TRS-PD). The initial design and implementation of the framework were based on identifying packet dropping adversaries in advance by observing and analyzing their behaviour during route discovery. The extension of the framework integrates sensitivity analysis by varying different parameters in distinct network scenarios. There were three packet dropping attacks initiated in these scenarios. The work also summarized the attack-pattern discovery mechanism, routing mechanism and trust model to prevent attack patterns and their malicious behaviour. The investigation was conducted on NS-2 and the results depicted that the

distrust threshold is an important feature for any designed trust-based scheme which should be set conservatively to maximize the rate at which the scheme detects malicious activity. The authors designed this trust-centric framework for MANETs in Industrial IoT using AODV and an extended version of AODV called TRS-PD, a protocol that uses a two-tier approach in which the attack pattern discovery mechanism is the main element of the trust model. The traffic implemented was CBR-UDP and not video streaming. This study extended the traditional OLSR protocol to add a sinkhole attack.

The authors in [76] designed a framework to prevent potential security threats that threaten the privacy of users and overcoming the limitations that compromise reliability and data integrity. The contribution to the work included the development of a secure neighbourhood awareness service and shared localization information management protocol. The framework was developed through different components including a local verification protocol to secure location information by providing a non-line-of-site (NLOS) verification protocol to remove obstacles. The second component was the privacy-preserving location information management responsible for the detection of data inconsistency and provide a recovery process whilst also preventing attackers from spoofing individual vehicles. The trust model evaluation mechanism was the third component and responsible for neighbourhood awareness. The last component was the adaptive beacon protocol that reduces the number of messages and provides QoS control. The authors also proposed a security evaluation model for the localization service and to also evaluate integrated security measures provided by the different components. The enhanced models proposed by the authors bring significant contributions to the security domain in VANET but not entirely on the QoS domain. The authors do not specify as to which attacks are targeted because a good security measure is designed as per known security threat to the network.

Gurumoorthi *et.al* in [77] looked into the implications of using the traditional SIP architecture in MANETs. Their study also examined the performance limitations within the SIP service measurable when centralized registers are used by MANET nodes then present an alternative approach to provide those services in an internet-connected MANET to reduce the performance limitations. To enhance SIP signalling performance over MANETs, the authors noted the importance of qualifying the SIP signalling behaviour to conform to the mechanical nature of MANET systems when comparing the two. MANETs do not have a central controller and SIP

architectures have registers that act as servers responsible for updating local information into the database with the user's contact information. There are also redirect servers responsible for mapping the SIP address of the called party into a new address then returning them to the client. The authors presented various performance methods that could be applied but in the end, concluded that the best approach to enhancing QoS in VoIP applications over MANETs would be to have SIP signalling behaviour synchronized or mapped to that of MANETs. The presented work did not offer any design or implementation suggestion as to how this would be achieved but different methods were simulated and analyzed.

Maamar *et al*. [78] proposed a mechanism to enhance the AODV routing strategy in the maintenance phase to provide better QoS. The strategy involved adding a mechanism to predict link failure based on the signal strength. The intention was to create a mechanism responsible for link management during breakages especially when the signal strength is low due to neighbour remoteness then a discovery part of road rescue would be established as an alternative instead of complete disconnection. The simulations were conducted on NS-2 to measure packet delivery ratio and traffic control on both the original and modified protocols. The proposed protocol outperformed the traditional AODV and improved the QoS for applications in MANETs. This mechanism did not address any security issues that may be present during the maintenance and handoff phase especially when there is a conflict caused by more than one procedure launch.

The authors in [79] proposed a cross-layered approach that involves the mSCTP transport protocol, the MIH standard and a custom-designed routing protocol called QA-AOMDV. The proposed framework performed context-aware routing and mobility management through LTE networks and MANETs. The simulation results showed that the proposed scheme reduced the ping-pong effect which is known for generally degrading the network performance in mobile nodes. The drawback of the scheme was that it solely depended on the RSSI parameter which was not well optimized to be adapted for a different kind of network other than LTE. This study was simulated for 10 nodes with CBR application traffic and there were no security considerations made other than the need to provide an enhanced QoS.

Xia *et al*. in [80] proposed a lightweight trust quantification framework to address malicious attacks and denial of cooperation. The framework was further divided into trust assessment and trust prediction. They used the node's historical behaviour to do node trust assessment then the

SCGM(1,1)-weighted Markov stochastic chain was used to predict the node's trust and its possible future decision-making capabilities. The authors also improved an existing protocol called On-Demand Multicast Routing Protocol (ODMRP) to propose a new one called On-Demand Trust-Based Multicast Routing protocol (ODTMRP). The experiment was conducted on NS-2 and the results showed better performance by the proposed protocol as compared to the other protocols evaluated. The historical behaviour cannot be truly used as a true quantifying measure of a node's trust hence the applied SCGM(1,1)-weighted Markov stochastic chain method can be deemed effective when the node is not compromised.

The authors in [25] presented the use of two differing QoS level schemes: Best Effort level and High Effort level for the development of streams. Their proposed framework involves a centre point that isolates streams with different level needs by dispensing them towards a similar goal. This was done using a procedure with two tables Dedicated Routing Table (DRT) and Standard Routing Table (SRT) responsible for the movement of data streams. The authors proposed Quality Optimized Link State Routing (QOLSR) as the basic OLSR protocol that was considered for development. The authors compared their two protocols QOLSR and OLSR in a neutral environment. The assumption was that there was no infiltration of an attack or malicious behaviour. The robustness of their method implemented can only be validated for neutral environments but this is not a practical case for MANETs as they operate in diverse environments. This work goes beyond protocol modification for QoS by also making it secure to malicious behaviour by bringing in an intruder detection system.

The authors in [81] propose an infiltrator detection system for node isolation attacks (DoS) against OLSR. As expected, the first line of attack for the attacker is to position itself as an MPR so that it sends fake information with a HELLO message. The proposed solution is based on identifying the false information with a HELLO message using the infiltrator identification system. The performance of the proposed algorithm is tested in terms of network throughput and packet delivery ratio. The proposed algorithm outperformed the standard version of OLSR on the tested parameters. OLSR is a protocol known for its good performance in dense networks. This algorithm was only tested for three mobile node-scenario for two QoS measures and thus its effectiveness cannot be judged based on three nodes analysis in a network scenario. This study is evaluated in different node density environments to validate its robustness.

Borkae *et al.* [82] present a routing protocol-oriented security framework for MANETs. They firstly conceptualized a secure adjacent position trust verification model. Secondly, they extended the normal *Ad-hoc* on Demand Multi-path Vector protocol (AODV) into a multi-path protocol which is trust enhanced meaning the addition of the trust model within the standard protocol. They named it AOMDV-SAPTV [82]. Their experiments were run to ascertain the effectiveness of the new protocol. Quality of Service (QoS) aware routing was implemented to get a feasible path from source to destination which was perceived to satisfy more than two end to end QoS impairments. An algorithm called the DE algorithm was implemented in this work to get the best and optimal path for routing. The results showed that their proposed technique enhanced the quality of routing and had found the best path using the DE algorithm [82]. The authors, similarly to this study, presented a QoS-aware strategy for routing and security. The evaluation of the presented framework was based on AODV and its efficiency cannot be guaranteed on other routing protocols. The main contribution to QoS was the formulation of an extended version of the AODV protocol.

Khan *et al.* [11] proposed a multi-attribute trust framework for MANETs (MATF). The framework was designed to minimize bootstrapping time and also to tackle selective behaviour. It was also designed to enhance the security of MANETs by enabling a node to identify and then quarantine malicious nodes from the routing paths. The proposed security framework was not only designed to detect malicious nodes and in time but also to decrease the number of false positives. They used multi-watchdogs as a measure to achieve this [11]. Their work was based on a reactive routing protocol, AODV, for the detection of wormhole attacks and not OLSR. The aspect of QoS was not a factor in their evaluation process and it is central to our research work.

Khan *et al.* [83], also proposed a Detection and Prevention System (DPS) to detect and also prevent compromised nodes in MANETs. To achieve their purpose, they created special nodes called DPS nodes which were used to continuously monitor patterns of behaviour in other nodes that are deployed in the network. When the special nodes found a node showing behavioural signs of suspicion, it would declare that particular node as a wormhole node through a broadcast message. All data and control messages that may emanate from the compromised node are then discarded by the network. Their experiments were conducted on an NS-2 Simulator. The DPS showed to have considerably reduced the rate at which packets were dropped by compromised nodes with a

low false positives rate [83]. The DPS nodes operate similarly as MPRs in the routing protocol OLSR which is the chosen algorithm for this research work.

Hurley-Smith *et al.* [84] proposed a novel security protocol called Security Using Pre-Existing Routing for Mobile *Ad-hoc* Networks (SUPERMAN). The initial protocol design was to address issues like node authentication, secure communication and network access control for MANETs through existing routing protocols. SUPERMAN brings together communication security and routing at the network layer. Their security protocol is unique from others in the sense that they combine routing and communication security at the network layer. This is in contrast to existing approaches that may require additional protocols to protect the network. The SUPERMAN proved to provide lower-cost security than SAODV and SOLSR for their respective routing protocols. SUPERMAN was created to give security to all data communicated over a MANET only, not every type of network. One efficient method it employed was protecting application data and routing, ensuring that MANET provides confidential, trustworthy communication, and reliability to all true nodes [84]. The further development of the OLSR and AODV routing protocol to improve QoS was a major contribution to MANET studies. The evaluation, however, is only centric to the network layer. It only protects application data and its routing not link state information and protection from existing attacks in the network. A part of this research work simulates an existing attack within the network.

Sharma *et al.* [85] used an Intrusion Detection System, AODV, and a Watchdog AODV scheme to simulate a Blackhole attack in MANETs and its effects. The emphasis is given to security in terms of it being a combination of procedures, processes and systems used to ensure authentication, integrity, access control, nonrepudiation and confidentiality. Their focus on security was based on the growing demand to support live streaming traffic for military and civilian applications. AODV had a decreased performance in simulation with a blackhole attack. The scheme infuses an element of trust management. The results obtained conclude that the Watchdog IDS trust management solution for preventing the black hole attack was better than other proposed solutions. The Watchdog had the advantage of listening to the nodes within the network transmission range and constantly monitoring packet delivery amongst the nodes. If a packet was not delivered within a specific time range or is forwarded but now altered by its neighbour then it deemed that neighbour as misbehaving and the watchdog would declare that node as a grey hole node and exclude it from

the path of sending packets [85]. Similarly, to this research work, the authors implemented an IDS but the only difference is that they based theirs on Trust, and this work used network state information. The effectiveness of the IDS was not tested for sinkhole attacks but blackhole attacks.

Kaur *et al*. [10] analyzed security threats MANETs face and presented the security objective to be achieved. Their security objective called CBDS was successfully implemented on the black hole and grey hole attacks which proved to be equally effective in the case of DoS attacks and Sleep deprivation attacks. Their simulation results showed increased detection for CBDS and an enhanced response [10]. Their work made significant strides to effectively detect black hole and grey hole attacks. There is, however, no evidence of its effectiveness on sinkhole attacks in MANETs.

Monica *et al.* [86] in their work analyzed, simulated and performed a comparative study of three different attacks based on many parameters. These attacks were denial of service (DoS), black hole attack, and wormhole for wireless *Ad-hoc* networks based on performance. The comparison made was for their Packet Delivery Ratio (PDR), End to End Delay and throughput. The attacks had different properties and different behavioural patterns of attacks in different environments primarily because of their different designs. This gave an idea on how to tackle such attacks based on how they performed and the various methodologies that could be invoked to counteract them [86]. This again would mean, a present opportunity for future research and improvements on the protocol. They concluded that the rate of data loss is much less in wormhole as compared to the other two attacks [86]. The best way to counterattack is to first know the behavioural patterns and properties of the attack itself and this is what the authors did. This work, however, was not evaluated for sinkhole attacks. There are a few studies that concentrate on sinkhole attacks in MANETs. This is the reason our research shifts from focusing on AODV or even the other well-known attacks to look at sinkhole attacks while using OLSR.

2.18 Cross-Layer Architecture

The cross-layer design of the OSI Model brings a shift to the traditional assumption by network architectures that communication functions are organized according to their protocol layers and that protocol headers from each layer are responsible for packet deliveries. The layers are modular and for every layer, there is a specific service that is implemented. The flow and communication

between layers are direct between adjacent layers that are using standard interfaces. As shown in Figure 2.8, the Cross-layer design violates this reference model by allowing interactions and also the exchange of state information among non-adjacent levels of the protocol stack. This is done to obtain performance gains including energy conservation and security[87, 88]

2.18.1 Security in Cross-layer design

The nodes in a MANET communicate openly in a shared broadcast channel and this makes them more vulnerable to attacks. It is easy for an attacker to infiltrate the channel because it is open. The use of multi-hops for communication is another weakness that implies that the entire network depends on the cooperation and solutions from individual nodes and such opens room for attacks. Cross-layer design of MAC, routing, and transport protocols guides how the design in security should be detailed. For example, an attack on the transport layer needs a particular counter attack solution for that layer and the same applies to the other layers. In a cross-layer design solution, all the layers are considered during the stages of protocol design[89].

2.18.2 Energy reservation in Cross-layer design

Energy efficiency is an important factor in the deployment of MANETs. It can be a limiting factor as the nodes are mobile relying on portable and limited power sources. The reservation of energy, in this case, is a challenging task because of the multi-hop routing of packets for other nodes and to guarantee connectivity in the overall network [87].

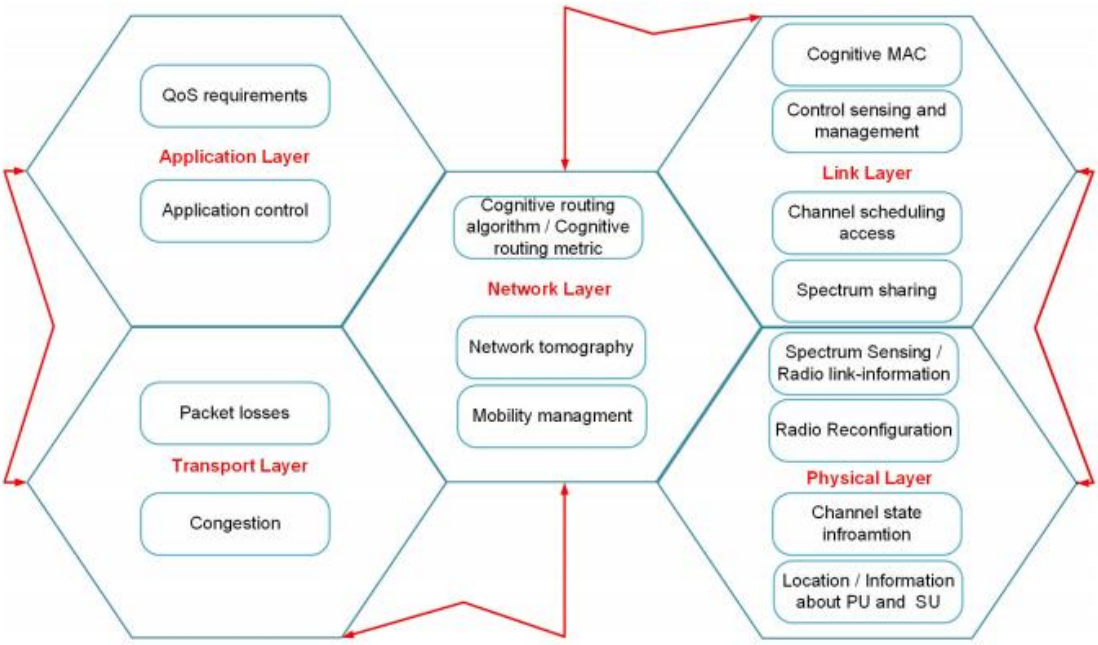

Figure 2.8: Cross-layer Model[87]

2.19 Chapter Summary

This chapter addressed **RO1** and **RQ1** as highlighted in Chapter 1.An extensive review of work done by network developers and researchers around security issues in MANETs with consideration for QoS was conducted. It also brought in some of the key critical elements, models and concepts related to the study including the strengths and weaknesses of previous studies conducted. Literature has proved a great deal of background in terms of how much work has been done in terms of security but most articles focus on specific attacks and protocols within MANETs. The area is huge and there is still more room to improve the security of *Ad-hoc* networks despite their dynamic topologies. QoS assurance in a network involves the exchange of information in the various layers. Most designed models in literature skip aspects like resource reservation, reservation signalling, connection admission control, reservation of bandwidth, mobility and scalability during the design phase of frameworks. The conducted literature also brought in areas of controversy that reveal gaps within the area of MANETs especially when it comes to the OLSR

protocol. Most of the reviewed frameworks focused on similar attacks being blackhole, grey hole, and DoS. The approach to security also matters. The cross-layer approach this research work is implementing is a holistic one because it ensures that no opportunistic form of attack attaches itself to the network. This is the reason why this work utilizes a cross-layer approach so that these important aspects of QoS are addressed. Chapter 3 will explain the methodology implemented in this study.

CHAPTER 3

RESEARCH METHODOLOGY AND EXPERIMENTAL SETUP

3.1 Chapter Overview

This chapter addresses the methods and models used to realize the goal of this study. The framework formulated in this thesis entailed the evaluation of QoS with optimum parameters set. The framework is termed "node-based" because MANETs do not have a central controller but every node possesses a sense of independence and acts as its router. This chapter also focuses on introducing a few concepts that are pivotal in the modelling, testing, and implementation of the framework. The first part of this chapter established the methodology and the related methods used. The second part brings about the actual simulation setup and conceptual framework overview. The implementation of the framework is done on NetSim, a versatile network simulation tool used for defence applications, protocol modelling and simulation. NetSim provides a good enviroment with unmatched depth, flexibility and power as it allows the designer to modify the routing protocol for performance purposes. The tool's versatility is that it allows the designer to also interface with external software packages like Wireshark and Matlab.

3.2 Introduction

QoS is not a command that works single-handedly but a series of mechanisms that work together towards the realization of efficient network performance. The OSI Model is used in this study as it provides a holistic approach to network configuration and management. The proposed framework is evaluated from the application link layer to the physical layer as most of the attacks investigated are domiciled within those layers. The applications layer was also utilized as custom traffic (High-Resolution video) would be generated to test the effectiveness of the proposed model using different scenarios. The work was implemented using a computer simulation methodological approach. According to authors in [90] simulation is defined as "the use of mathematical modelling, performed on a computer, which is designed to predict the behaviour of a real-world or physical computer program". Experiments are executed using the discrete event simulation approach to achieve this work's research objectives and the main goal of this investigation. Discrete event simulation involves the modelling of a system as a set of entities evolving according

to the availability of resources and the triggering of events[90-92]. Discrete event simulations are discrete, stochastic and dynamic. A system is considered stochastic when its behaviour is determined by one or more random variables. Dynamic simulation involves and integrates design, operation and control in the same environment. A discrete system is one where the state variables only change a limited number of times at specific instances in time. The following sub-sections present the techniques used to fulfil the research goal:

3.3 Research Methodology and Research methods

A research methodology is a systematic way of solving an existing problem. It is the science of studying how a research goal is to be met and how the general research is to be carried out. Researchers use different ways to describe, explain, and predict phenomena in their work and this is called research methodology. The research methodology is sometimes referred to as the study of methods at which knowledge is obtained by giving a clear work plan of the research work including means of addressing the research problem [93, 94].

Research methods can be categorized as the different procedures, algorithms or schemes used in research. They are a myriad of tools that researchers employ whilst doing any form of research. Research methods are planned,value-neutral and scientific. They include numerical schemes, statistical approaches, experimental studies and theoretical procedures. The collection of samples and data is possible through various scientific research methods that call for explanations based on measurements, collected facts, observation and not on self-reasoning explanations. The only accepted explanations are those which can be verified by experiments. The selected method has to be appropriate for the task, is commensurate with the researcher's skill set, and is one with more strengths than weaknesses than other methods that could have been used [93, 94].

3.3.1 Quantitative Research Methods

Lakshman *et al*. in [95] expressed that quantitative research methods examine the effects of an independent variable on a dependant variable in ways that can be expressed numerically. In this method, causal inferences are drawn from true experiments (direct observation) or associations drawn through statistical analysis. Survey research and questionnaire forms dominate socio-economic investigations for evaluations particularly in developing countries and are standardized

for increased reliability and precision of data collected. Chen and Murray in [96] defended the use of surveys in light of criticism from the research community through a descriptive definition *" A rural third world survey is the careful collection, tabulation and analysis of wild guesses, half-truths, and outright lies meticulously recorded by gullible outsiders during interviews with suspicious, intimidated, but outwardly compliant villagers".* The research output from quantitative research can be explanatory, predictive and confirming.

3.4 Methodology used in this thesis

To successfully achieve this work's objectives and the main goal, the following research techniques were applied.

3.4.1 A comprehensive literature review

This was done to establish the research gap from literature and to better understand the efforts and other works thus far studied. The methodology used in this part of the research is the systematic review methodology used as part of the methodologies in this thesis as according to Linares-Espinos *et al*.[97]. This was the critical and verifiable summary arising from the various publications that address the aim of this research done by various researchers globally related to the field of study and various literature consulted.

3.4.2 Simulation

The simulation approach involves the creation of an artificial environment within which relevant data can be generated. This is done in a controlled environment and it permits the observation of a system's dynamic behaviours. One of the purposes of simulation is to perform real decision making or diagnostic tasks. Decisions taken through analytical formulations are less preferred compared to simulations because they do not provide certainty [90, 98]. Network simulators best describe and represent the state of the network. These include nodes, links, switches, hubs and routers. In modern-day simulators, they are either Graphical User Interface (GUI) driven or Command Line Interface (CLI) driven. Simulation is mainly used in performance analysis, comparison, or even management and also for determining how a network would behave in a real-life situation. The generated result of the simulation helps in identifying the performance attributes of the network. The flair to the entire simulation software is that there is a wide range of tools that

ensure the generation of excellent results (GUI-based). There is a choice of network traffic selection, programming environments, projection and statistical data representation. These are all part of the package of the simulation tool.

3.4.2.1 Purpose of simulation

a) Prediction

Simulation for prediction is being used as a substitute for experimentation and intervention in cases where it would be dangerous, inconvenient or untimely to conduct that experiment on an actual system. This often is the case with human beings as subjects and to large scale elements of the organization e.g supply chain and information systems because of scalability challenges. Simulation simply takes a structured model with rules that govern that particular model and then produces output based on observed behaviour. Researchers can infer what would happen in a real-life situation through a comparison of different outputs obtained from the different structures and governing rules. The validity of the predictions made depends on the validity of the model.

b) Performance

Simulation is also used to perform real tasks such as diagnosis and decision-making through a calibrated and validated model. There is always the natural context of uncertainty and randomness of the system. Simulations are used to mimic that uncertainty in the form of a Monte Carlo simulation. Monte Carlo simulation predicts the probability of different outcomes when the intervention of random variables is present by explaining the impact of risk and uncertainty in forecasting and prediction models. It performs risk analysis by building models of possible results by changing a range of values. Unlike discrete event simulation, Monte Carlo simulation does not emphasize the variable of time.

c) Training

In the military, aviation and nuclear industry, simulations are used for training purposes. Simulation makes it easy, quick and safe for users to make up decisions that are similar to decisions they would have made in reality. Crises are expensive, dangerous and too difficult to set up in a physical context hence computer simulation is used together with virtual reality. In this way, the

organizations can anticipate ways to make their products more stable and capable to work in a real-life situation. There are many elements considered within the context of this type of simulation e.g how people around will react, environmental impact, the stress associated with situations and timing required for decision making. All these elements provide relevant data that is then used for learning purposes and making the output product safer and performance-ready.

d) Education

In the education sector, simulation provides users with a metaphorical and deeper conceptual understanding of how complex systems work. This is different from simulation for training used in military and aviation. In this case, users gain knowledge in mechanisms like reciprocity, noise, nonlinearity, feedback, etc.

3.4.2.2 Simulation implemented in this research

Discrete event simulation and experimentation using Netsim are implemented to adequately characterize variables, corresponding states and events that change the value of these variable states in some rule-oriented but stochastic manner. The entities are the different components in the proposed framework. A network design [1] that refers to the planning of the implementation of scenarios is also carried out. Figure 3.1 shows elements of a discrete event simulation as characterized by Law *et al*. [99].

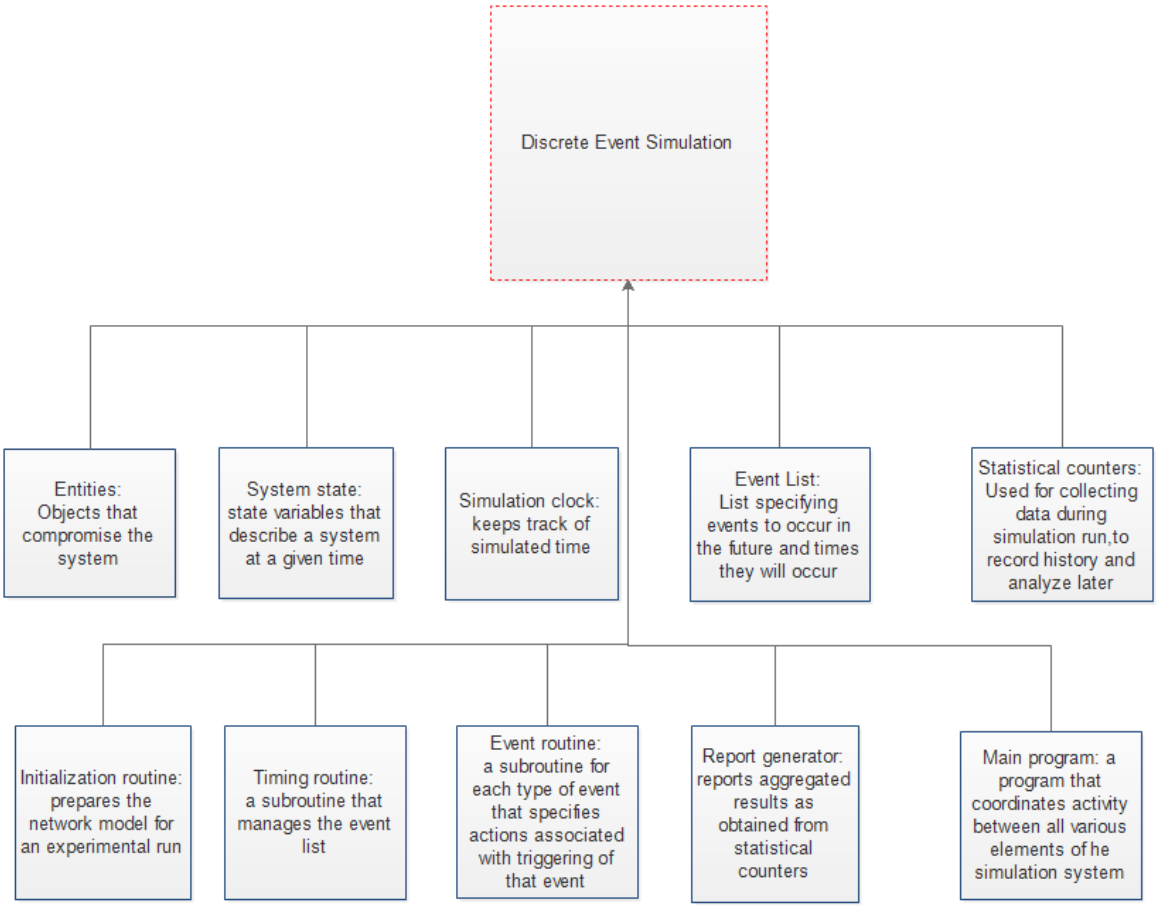

Figure 3.1: Elements of discrete-event simulation [99]

3.4.2.3 Implementation of the simulation model

This is the most important aspect of a simulation study. The implementation of the simulation model is highlighted in Figure 3.2 [9].

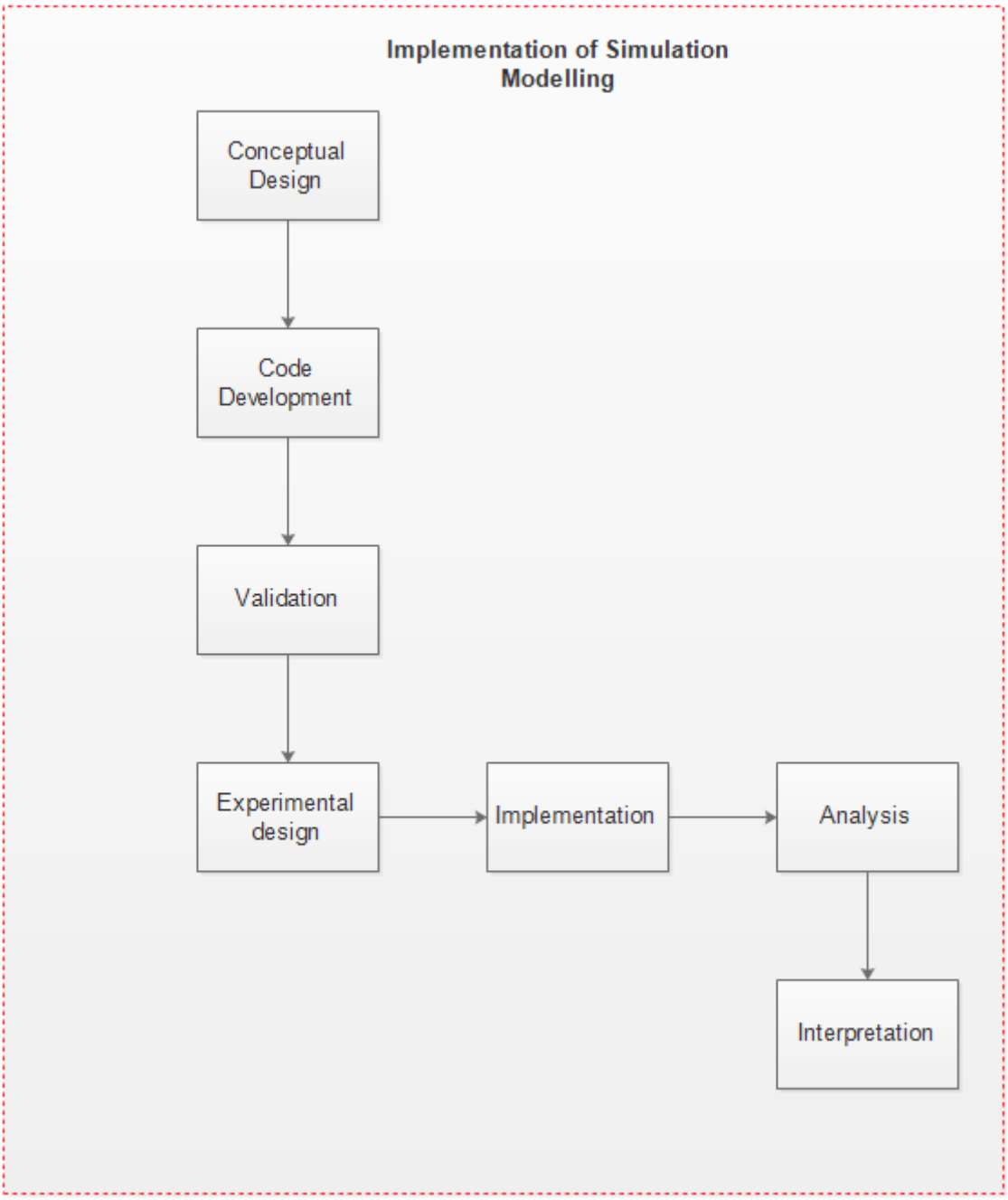

Figure 3.2: Implementation of a simulation model [9]

a) Conceptual Design

At the implementation stage, the user determines what is to be modelled, who will be using the model, the requirements and the research questions to be asked via such a model. A simple conceptual design over a complex one is always commendable unless the complexity added to it

is well justified hence preference is given to more parsimonious designs. In the conceptual design, there must be cohesion between theoretical propositions expressed and the model that is developed.

b) Code Development

The computational language is built into the simulation software and the user can write statements that are written in higher-level computer languages like C++ or C. Software packages like NetSim enable the user to construct a simulation model using drag, drop and click on the screen. The more user-friendly the modelling environment, the more complexity is hidden from the frontend but NetSim gives out source code for development and modification.

c) Validation

The results obtained in a simulation may not necessarily have an error but still have no validity associated with it. In some cases, the results are open to interpretation, and only an expert can tell whether the results make sense. In such a case, there needs to be a historic quantitative behaviour that can be compared to. The computational model has to meet the quantitive system history distributionally or pattern-wise. If it is a distributional fit it means that the variable displays the same mean and covariance structure that can be closely related to one in reality. A pattern-wise fit would mean the variables are related to each other in a valid manner but having numerical values that have no relation to real behaviour.

d) Experimental design

The user sets up a series of experiments showing the initial and runtime parameter values that will be used to provide answers to some of the questions posed. There is a huge gap between research questions linked to the goal of the research and theoretical propositions, and operationalization of that in an experimental design. Questions related to the experimental design include those that have to do with model configuration, length of runs, how the output should be interpreted and analyzed, effective ways to make runs and the frequency of the runs. A good experiment makes the analysis and discovery process more efficient.

e) Implementation

The experimental design is executed using replicates. A replicate can be defined as multiple runs of one experimental condition achieved by implementing a different stream of randomized numbers. During the implementation phase, the simulation model is initialized to some starting conditions. The data obtained during the initial configuration period should be ignored unless the user is interested in transient behaviour. After some time, the simulation behaviour changes to some "steady state" and at this particular point in time the further observation gives new information in contrast to the initial one.

f) Analysis

The replicates implemented are averaged over an experimental condition for further analysis and the user can check the data under the model being hypothesized. Graphs that represent time-oriented behaviour for the various experimental scenarios are drawn.

g) Interpretation

The results are discussed through the observation made during the analysis phase. The results need to be clear for the user not to attempt to link the simulation results back to the theory of interest by an overcompensation of facts.

3.4.2.2 Simulation Environment

The simulations are carried out on a Windows 10 Professional Operating system (Desktop) with 3.40 GHz processor speed, 4.00 GB RAM, and 64-bit Operating System.These are ideal specifications for the simulation tool.The minimum RAM required is 2 GB and 2.44 GHz processr speed.

3.4.2.4 Software for simulation and justification for selection

NetSim is developed by an Indian organization called TETCOS and was initially developed by BOSON. The connection is made possible by a NetSim cloud server that remotely connects client nodes. Our workstation connects through the server's VPN. NetSim enables users to virtually create a network along with its components such as devices, links and applications, etc. One other

feature NetSim provides is a graphical editor interface that is used to build models for different network entities. NetSim is known to provide a great simulating environment to implement new customized ideas and solutions and thus give the user an extended platform to re-create a complete network. This one advantage is backed by lower costs. It has an extensively wide range of protocols in its library and also a huge library of ready models that belong to fixed network hardware that is commercially available.

In this study, vehicles were represented by mobile wireless LAN workstations that move randomly. The software is very efficient in the study of application-based protocols. Its potential to generate several traffic types adds on as another advantage. Its C source code enables us to modify the protocol code by adding features that are unique to this thesis and thus extend knowledge in the study of sinkhole attacks. The results dashboard is easy to read and analyze with graphs and tables. NetSim allows packet animation which enables users to visually understand the protocol being implemented. Our target within the tool is its ability to redefine the source code of the protocol for improved performance and security. It also allows the programmer to study and choose between different network technologies and adjusting application settings. NetSim has an open architecture with its protocol written in the language C (source code) to code, link and also debug.

3.4.3 Experimental Workflow of the simulation in NetSim

The experiment workflow is shown in Figure 3.3.

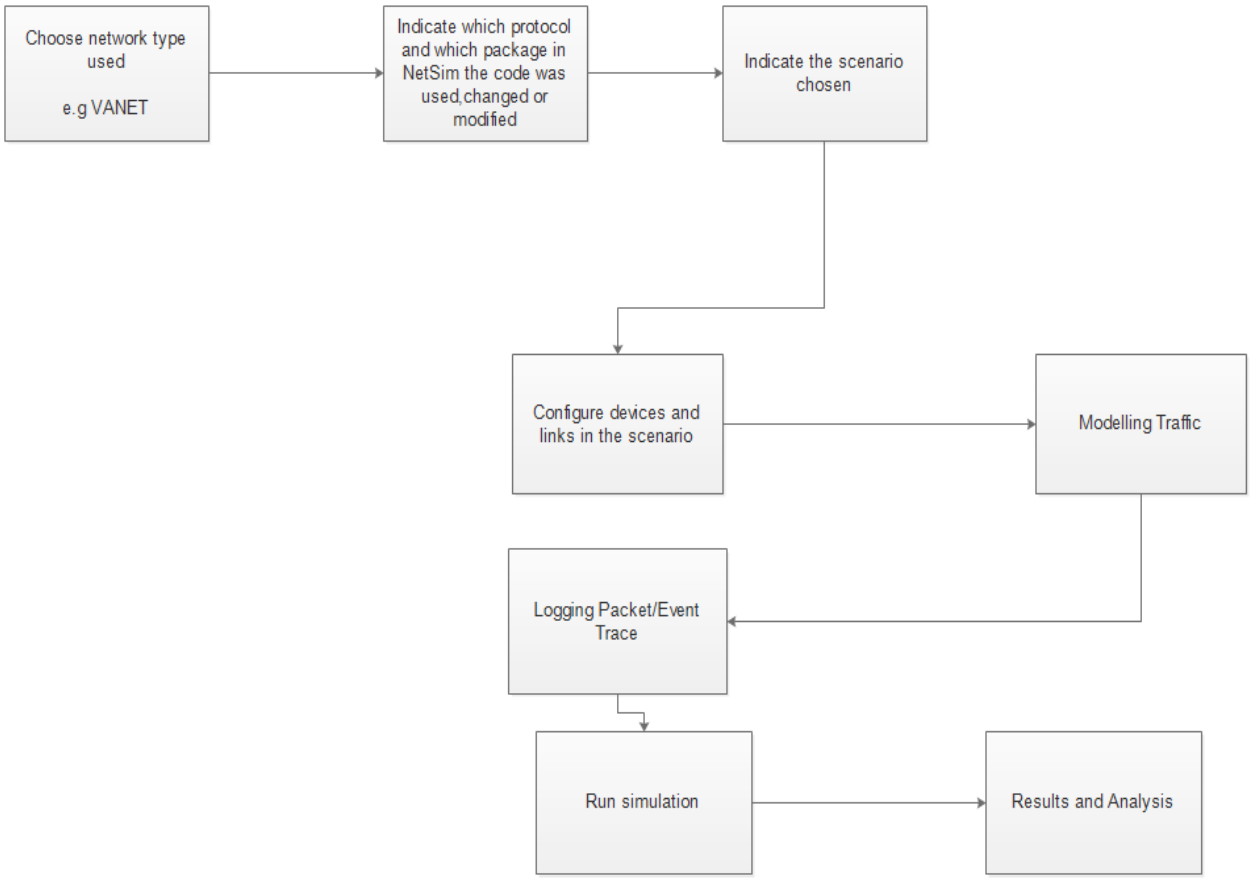

Figure 3.3: Experimental workflow

3.4.3.1 Choose a Network

There is a variety of networking areas to select from including Wireless Sensor Networks, Cognitive Radio Networks, VANETs, Slotted Aloha, Internetworks, MANETs, Pure Aloha, etc. The selection of a network is then followed by the creation of a workspace at which different scenarios can be created. The next step is to set grid settings and a device placement strategy.

3.4.3.2 Write own code

The existing algorithms were then modified by altering NetSim's source C code. A customized version of the protocol OLSR was created using NetSim's application programming interface (API). A dynamic link library (DLL) was created for the custom code. A DLL is a shared library concept implemented by Microsoft. The file is distinguishable by its extension (.dll). The file is not directly executable but a compiler like Visual Studio is used to build DLLs. In this thesis, the

compiler used is Microsoft Visual Studio 2019. Lastly, the new code was debugged and variables were synced with the simulation. The modification to the code was done in two phases. The first phase involved adding a sinkhole attack into the network by adding an extension to the existing OLSR source code. This new *OLSR Malicious.c* code was integrated with the main ZRP/OLSR project folder which contains the OLSR header files. Figure 3.7 shows malicious code functions that were added to the ZRP/OLSR project together with its header dependencies. Project dependencies include IP, application, firewall and battery mode. One node in the program is defined and configured to introduce a sinkhole attack in the network. After changes were implemented to the project, it was important to "Rebuild" the project to update the DLL library file. The implemented changes also reflect on during the simulation. The last phase of the modification is explained in Chapter 4 with the framework formulation. NetSim allows us to modify the source code at any stage of the simulation lifecycle to readjust functionalities.

3.4.3.3 Create and design network scenario

The first step towards the realization of the experiment is to first design the overall network architecture. NetSim provides a GUI or XML configuration file. Devices, links and applications are clicked then dropped in the environment using NetSim's GUI. The next step is setting up properties with a click to edit the layered parameters. This thesis focuses on the design of four different network models representing different node densities.

3.4.3.4 Configure devices and links in the scenario

The second step was to configure the devices in terms of application status, movement, protocol parameters and IP addresses. It is also very important to configure links by right-clicking to adjust their parameters. The mobility of the nodes is set to Random Waypoint to allow the random movement of mobile users as their location, velocity and acceleration changes over time.

3.4.3.5 Modelling traffic

By using the application icon, traffic was then configured and modelled. There are various choices among video, voice, database, email, FTP or any custom traffic. The nodes are individually set as MANETs and thus have no central controller as they act as their routers. The dialogue for node

configuration with the Application parameters includes the Encryption standard to select from and QoS scheduling service types (QoS classes).

3.4.3.6 Run Simulation

This is the actual running of the experiment with customized settings. The discrete event simulation (DES) was run to capture an ordered sequence of events in time. An event in this context would be a particular change in the state of the system at a particular time. The running of the simulation can either be done on the GUI or command-line interface (CLI).

3.4.3.7 Logging Packet / Event Trace

This functionality is to trace the packets as they move within the network. It can be visualized as it moves from hop to hop. The event trace files are used for detailed simulation analysis. In our work, event trace files were enabled to retrieve the various statistics about the performance of the network for further analysis.

3.4.3.8 Results and analysis

After running the simulation, the results are viewed. The first thing was to examine the output performance metrics from various levels of the OSI model i.e. network and application, then looked at the various metrics such as jitter, link utilization, throughput, delay, etc. Lastly, the metrics were interpreted as per trace files using graphs and plots to create pivotal tables and charts for proper visualization and analysis. The results are discussed in Chapter 5 of this thesis.

3.4.4 Simulation parameters

The simulation environment was prepared to create different scenarios and obtained the best possible results.

Table 3.1: Simulation parameters

Parameters	**Value(s)**
Simulator	NetSim Standard v12.1

Application Protocols	OLSR
Grid length	1000 m*1000 m
Simulation time	100 seconds
Traffic type	Video
QoS Class	ertPS (extended-real-time Polling Service)
Node movement model	Random Waypoint
Trajectory	Random
Transport Protocol	UDP
Speed	50 km/h
Refresh interval	2 s
Encryption Algorithm	Advanced Encryption Standard
Node density	16,49,100
Congestion Control Algorithm	New Reno

3.5 QoS measures

In this study, the QoS measures used to ensure that quality is optimized in the network are throughput, delay, and jitter.

Throughput: It represents the number of bits forwarded from the Medium Access Control (MAC) to higher layers in all nodes in the network. It is measured in bits per second. The throughput may also be referred to as the average number of packets successfully transmitted or received per second. This work focused on the application throughput which is the total user data sent to the intended destination per second. The formula to calculate throughput is given as:

$$\text{Throughput} = \frac{P_D}{T} \tag{3.1}$$

Where P_D is the number of packets delivered and T is the time in seconds.

Delay: This is normally the time taken for one packet to be transmitted from the source node to the destination node. This performance metric evaluates the routing protocol's effectiveness in the use of network resources. Delay may be caused by several obstacles including transfer time, buffering during discovery latency, interference queue, and propagation. The formula to calculate delay is given as:

$$\text{Delay} = T_{rx} - T_{st} \tag{3.2}$$

Where T_{rx} is the time the packet is received and T_{st} is the time the packet is sent.

Jitter: This is the variation in time between route changes and data packets arriving. The variation may be caused by internal sources like data transmission errors, the presence of a malicious node, and network congestion. It usually affects the audio quality of the video if its level is high. The formula to calculate jitter is given as:

$$\text{Jitter} = D_t - D_P \tag{3.3}$$

Where D_t is the transmission delay of the current packet and D_P transmission delay of the previous packet.

3.6 QoS-aware security Architecture

The architecture presented in this work is a conceptual model of the proposed framework. The overall framework has an adaptation of a few layers of the OSI model as shown in Figure 3.2. The layers exhibited are the application layer, transport layer, network layer, data link layer and physical layer.

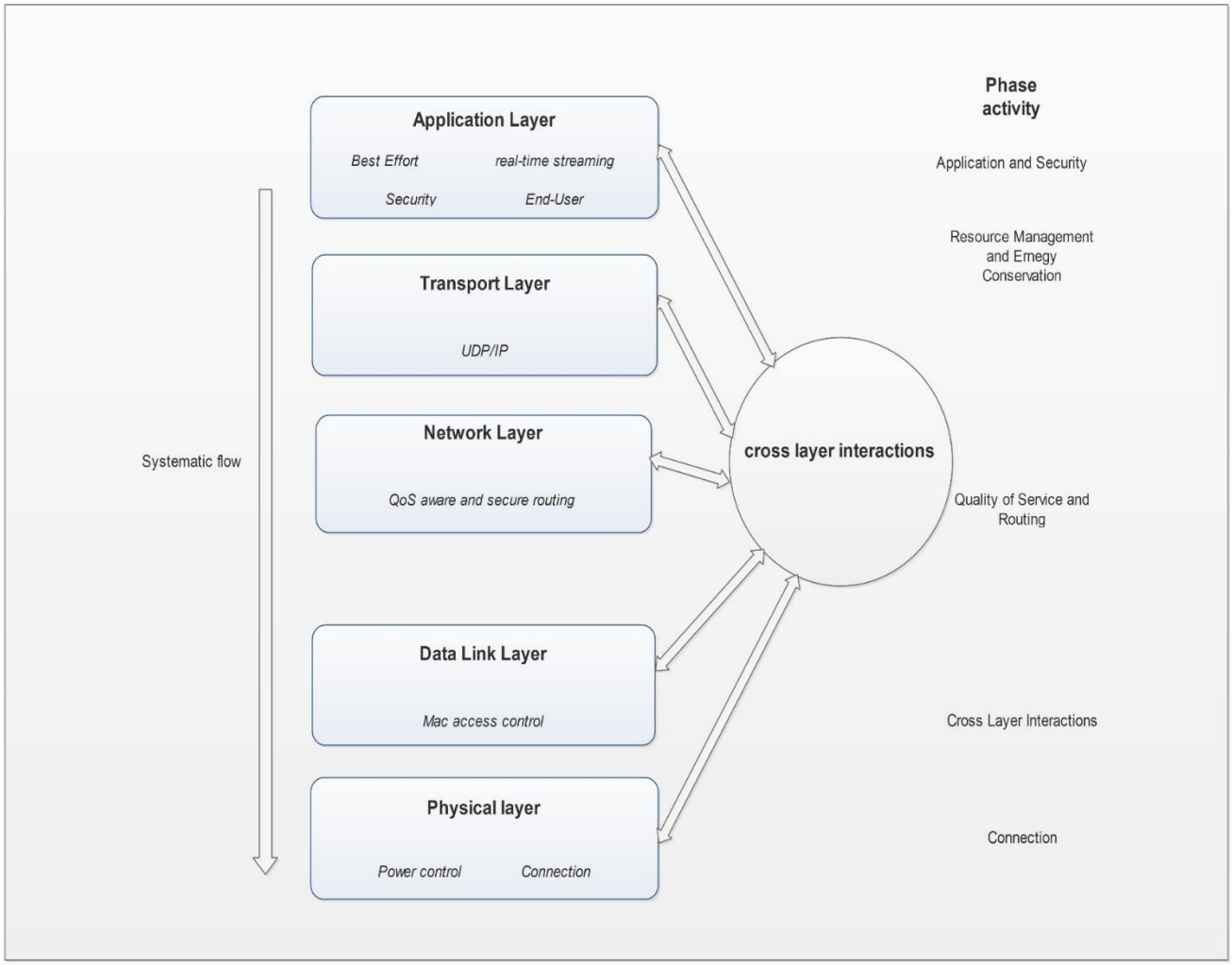

Fig 3.4: Conceptual QoS-aware security architecture [100]

3.6.1 Application Layer

The application layer is mainly in charge of generating relevant traffic i.e. CBR, video, voice, email and HTTP. To fully ascertain QoS, a resource reservation scheme is used. A resource reservation scheme will ensure that QoS for high priority sessions is guaranteed and that sufficient bandwidth is administered throughout the transmission phase to fulfil the fundamental requirements of QoS. The assigned resource reservation scheme will guarantee QoS performance as it decreases the chances of high priority session collisions while using the bandwidth. The real-time Polling Service (rtPS) algorithm is a QoS scheduling scheme that supports real-time service flows and reservation of resources like bandwidth requests during transmission. Application security threats like malicious nodes, worms and viruses are dealt with in this layer through keying.

The end goal of the yet to be formulated framework is the assurance of QoS within the network. This can be achieved through:

a) QoS Medium Access Control (MAC) scheduling
b) Admission Control
c) QoS-aware routing
d) Traffic policing

These four mechanisms are covered throughout the framework at different layers in terms of implementation. The proposed QoS-aware routing protocol solution will be based on:

a) OLSR protocol and QoS
b) OLSR protocol and MAC protocol

3.6.2 Transport layer

The transport layer involves the movement of packets from the application layer to the network layer. It focuses on end-to-end communication during data encryption. This can be done using the Transport Control Protocol (TCP) or User Datagram Protocol (UDP). The key activities in the layer are as follows:

a) Congestion Control Algorithm: this is used to control the congestion within the network. In our work, we implement New Reno, an algorithm that improves retransmission during the quick-recovery phase of TCP Reno. New Reno consists of various mechanisms including Fast Retransmit, Fast Recovery, Congestion Avoidance and Slow Start. The Slow Start and Congestion Avoidance are responsible for increasing the congestion window in the absence of losses to use up the available bandwidth. Congestion in a network is determined by packet losses which are seen through duplicate acknowledgements and timeouts. Fast Retransmit and Fast Reconnect are known for mild congestion and this is achieved by reducing the congestion window by half instead of one packet after a timeout [101].

b) Transport protocol: a transport protocol has the prime responsibility of establishing and facilitating the movement of data from one node to the other. In this work, the UDP protocol is used because of the live streaming data traffic. UDP enables the sending of packets with multi-cast than to each connected client which would increase the load and overhead in the network.

Using UDP greatly reduces the amount of upstream bandwidth utilized. The video would have to be stopped with TCP while it resends packets in an attempt to stream mathematically perfect packets. This causes a delay in the video stream as it would have to pick up from where it left off after the missing packet makes it to the destination [102].

c) Encryption: this is one of the most effective ways of achieving data security. A secret key will then be used to have access to an encrypted file in a process called decryption. This work uses AES (Advanced Encryption Standard) algorithm. AES is a symmetric-key block cipher algorithm used to encrypt and decrypt data. The same key is used for both the encryption and decryption process. Encryption involves the encoding of data to prevent intruders from accessing the original data easily. The plaintext data is converted to an unreadable format called Ciphertext. Decryption involves converting the ciphertext back to plain text without missing or skipping any words from the original message. AES is sometimes referred to as the Rijndael algorithm. The Rijndael algorithm is based on the substitution-permutation network which involves a series of linked mathematical operations with a fixed block size of 128 bits and key sizes of 128,192 and 256 bits [103]. This research acknowledges that encryption is not the only viable solution when looking at the threat model the framework seeks to address hence the introduction of an intrusion detection system [104-107].

3.6.3 Network Layer

At the network layer, routing takes place. The network layer is more central to the realization of a fully functional system enabling cross-layer communication between the transport layer and the network layer. This includes:

a) **Adaptive routing (QoS-aware):** a process of determining the most efficient path in which a data packet can be used in a network to reach its destination. OLSR is used to read and respond to the changes in the network topology. It is sometimes referred to as dynamic routing because when a packet arrives at a node, that node uses information shared among the nodes to calculate the most suitable path which in this case would be MPRs. If the chosen path happens to be congested then the packet is sent using a different path and the information is shared among MANET nodes. In this way, packet delivery failure is avoided, network congestion is minimized and performance is improved.

b) **Routing protocol:** determines how MANET nodes should communicate with each other and enables them in selecting optimal routes between any two nodes within the network.

c) **ICMP status:** The Internet Control Message Protocol (ICMP) relays messages about the status of our IP address e.g. Destination Unreachable, Time exceeded, and Trace Route. It is different from transport protocols like UDP and TCP in that it is only used for diagnostic or control purposes in response to IP errors not the exchange of data between systems. It is a network layer protocol whose messages are processed differently from the normal IP even though ICMP messages are found in every IP packet. In this thesis, the status of the ICMP is active so that control messages about node availability and packet delivery is available to the end-user.

d) **Packet scheduler and buffer:** this contains the actual memory that is used to store packets. Additionally, the scheduler automatically builds a protective front (firewall) against hostile nodes and thus protecting network resources from saturation.

e) **Admission control:** this is a very important component in the proposed framework. It is key in the provisioning of QoS in the MANET because it determines the extent to which network resources are utilized and whether or not QoS characteristics are delivered. Admission control is more of a validation process where a check is done before the establishment of a connection to calculate the sufficiency of network resources for a proposed connection.

3.6.4 Data Link layer

At the data link layer, the IEE 802.11 standard is utilized. The MAC plays a huge role in linking the network and physical layer. The MAC address handles queues during routing and bandwidth estimation during cross-layer interactions between the network to the data link layer and from the data link to the physical layer. The responsibility of the link layer in this framework is towards the MAC access control and it performs packet queuing, priority classification, bandwidth classification, MAC differentiation, packet scheduling and forwarding.

3.7 Chapter summary

This chapter introduced the simulation environment and the justification of the proposed tool. It also presented the methodology that would be used for the research. Simulation is a powerful research method that enables the user to look into the future and also attempt to make enhancements to performance. This method is generally safe and there are no risks or environmental concerns since it is laboratory-based and is just utilized as a theory testing mechanism. . The network design, modelling and simulation parameters were presented. The chapter also presented the simulation modelling from the conceptual design, code development through to validation stages on the implementation of a simulation model. Discrete event simulation and the execution processes were also highlighted from the experimental design and implementation while analysis of output is the same as the interpretation of observation or result. The experimental workflow was presented where the user was given the choice to design a network model, write code, design a network scenario, configure devices, model traffic, run the simulation, and analyze the results obtained from that simulation. The conceptual framework presented a platform to explain the key features of the proposed framework presented in Chapter 4. In the next chapter, a breakdown of the framework from its formation to operations and its implementation is presented.

CHAPTER 4

FRAMEWORK FORMULATION, DESIGN AND IMPLEMENTATION

4.1 Chapter Overview

In this chapter, the proposed QoS-aware security framework is formulated. The processes that lead to the formulation of the framework were discussed and the underlying components that attribute to it. This chapter presented a QoS evaluation of the routing protocol, OLSR, implemented in aiming to achieve good QoS as compared to normal OLSR. The QoS-aware security framework implementation presented answers to research questions RQ3 and RQ4 along with the research objectives RO3 and RO4 as stated in chapter one of the study. Also presented are key QoS measures/parameters and how the protocols behave during the simulation. Also presented in this Chapter is the proposed security keying mechanism and how it links up to the entire framework.

4.3 Framework overview

The conceptual view of the framework presented in Chapter 3 demonstrated some of the key stages in the actual prototype. The prototype in Figure 4.1 follows the same design mechanism like the one in Figure 3.6. The TCP/IP model is at the core of the design and implementation of the framework. At the upper level is the application layer which contains video traffic generation, admission control and a resource reservation scheme that links up with the keying system at the transport layer. The keying system is responsible for security encryption and decryption to prevent attackers from having access to information passed on from the source to the destination. The utilized encryption method is the AES standard as shown in Figure 3.7. A proper data handover architecture was designed to fully support MANET environments. The alternative route in the security structure was designed to maintain data integrity and prevent passive attacks which may phish for information in the channel and observe activity within the channel. The transport classifier then takes over from the transport protocol which in this instance is UDP. This protocol is chosen over TCP based on its excellent delivery in video streaming traffic. Cross-layer interactions between layers (application and transport) continue because of the transport classifier which links up with the application.

The network layer is important in the framework because it directly deals with the communication and routing of packets using nodes. It is also responsible for choosing the best available path for packet transfer by closely working with the transport and MAC layer. It is primarily responsible for QoS aware routing and that involves a technique called adaptive routing. Adaptive routing determines the best/optimal path a data packet should follow in the network to reach its intended destination. This is achieved by using the shortest path algorithm which takes the data packet to the destination with minimal congestion and thus efficiently using network resources. This algorithm allows nodes to calculate routes in a given network topology and thus saving time and minimizes overhead size. Adaptive routing improves network performance as routes adjust automatically in response to dynamic network topology. The nodes exchange route information and updates during adaptive routing. This is necessary to fully adhere to QoS requirements. The routing protocol, OLSR, facilitates routing in the network layer. It is the highest decision making entity that is responsible for directing all packets from source to destination. The routing table is constantly updated as the node moves randomly within the network. Most of the operations that happen in our framework depend on the network level. The ICMP status is for monitoring the IP connection of the network.

The data link (MAC) layer provides more of a link address (MAC) to associate with the IP address shown at the ICMP status. The framework also has a MAC differentiator, packet queue management and available bandwidth estimator. These addresses work to ensure that packets are properly scheduled, forwarded and allocated appropriate resources (bandwidth) as well as that each node's unique identifier (MAC address) is linked with an IP address. There are many cross-layer interactions within this layer as the physical layer is also involved. When packets are forwarded to the physical layer then the process of moving from the physical layer to the application layer begins. The packets will move from the MAC differentiator to the QoS aware mechanism, link up with the transport classifier, and lastly the application, where it all started. The functions within the framework depend on a series of interconnected layers. The inclusion of a lightweight Watchdog and Pathrater is imperative in the detection and response to malicious traffic.

4.4 Framework Design and Implementation

The proposed framework follows a cross-layered structure. Cross-layer design in the context of MANETs refers to the sharing of information among the different layers for the efficient use of

available network resources and promoting adaptability. The main reason for cross-layer design is to sustain the functionalities found in the original layers and additionally allowing interaction, joint optimization and coordination of cross-layer protocols. Each layer in the design is characterized by having unique parameters to help them determine the best adaptation rules and control knobs for current network status. The optimization aspect of the design comes with the customization of variables and constraints from other multiple layers. The design speaks not only towards an isolated problem but also other parts of the system that may encounter unintended effects while the problem is being resolved. This is the reason why security and QoS come at a balanced scale because the complexity of the design might affect other elements of the system then resulting in poor transmission of packets or a weakened system. The highlight of the model is the intrusion detection system and is embedded in the different layers of the model. The uniqueness of the proposed framework is that it incorporates QoS and security implementation in more than one functional layer in a proportional way.

4.4.1 Motivation for Framework design

The key motivation behind the design outlook of the framework is due to the rise of new problems in MANETs and these include the possibility of opportunistic communication by malicious devices. The chosen design can accommodate new modalities over the network. This research work introduces new interfaces like the lightweight IDS in the cross-layer framework which is something that would have been otherwise close to impossible within the traditional layered architecture. Another motivation is the introduction of an upward-downward and downward-upward flow within layers as seen in Figure 4.1. The key design features of the proposed framework are towards fostering a method in which the layers can communicate with each other without compromising the quality and integrity of the data. According to Thota [108], the cross-layer design is known for its substantial gain in QoS, output and efficiency. The traditional layered architecture does not allow direct communication between layers but with this design adjacent layers can communicate and every layer works on its variables. The cross-layer design of the framework also allows new abstractions within the communication infrastructure between layers. It allows for the cooperation of new protocols between the different blocks for flexibility and optimistic designing during run time.

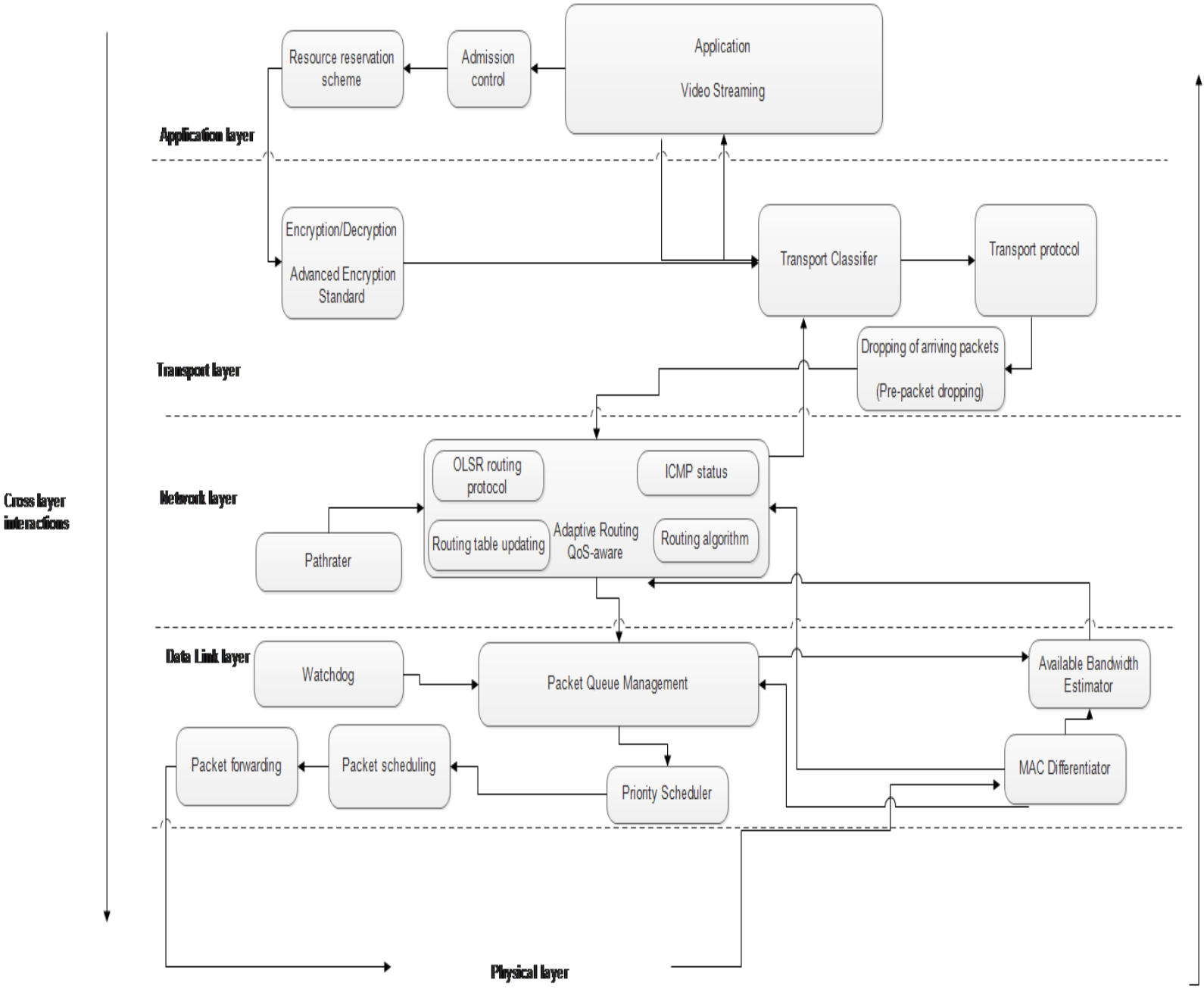

Figure 4.1 QoS-aware security framework

4.4.2 Security keying mechanism

As outlined in Chapter 3, the security keying system used in the framework is the AES Standard for both encryption and decryption. AES [105] is chosen on the basis that it consumes fewer resources (e.g. battery power), is fast and is a most effective algorithms. AES is sometimes referred to as the Rijndael block cipher operating on matrix blocks with 8-bit entries of size:

$$4 \times N_b \quad (4.1)$$

Whereby $4 \leq N_b \leq 8$ is the block length. N_b represents the number of bits.

Encryption scrambles the message and outputs it as unrecognizable data. Decryption takes the encrypted data which would be in the form of unrecognizable data and adds a source key to output the original message. The AES is proposed because of its less resource constraint, lightweight, and less computationally demanding features when it comes to routing. The AES is made up of a couple of initialization steps. The first step is the key expansion where the key is broken down into multiple subkeys and the other step is the initial round which involves substitution and transposition. The keys can be broken down into the following:

Table 4.1 AES algorithm processing properties[109]

Bits	**Cycles**	**Rating**
128-bits	10	Fastest
192-bits	12	Slower
256-bits	14	Slowest but most secure

In this work, the 128-bit version of key management is implemented. A series of rounds are performed using the multiple sub-keys made during the expansion phase. The number of times the rounds are made depends on the size of the key we select as highlighted in Table 4.1. As part of the contribution of this work, we introduced an intermediate trusted node between the source and the destination. This unique node can relay the same message packets issued from the source node to the destination. It provides an alternate route to secure the integrity of the message. This is done to ensure that at the destination, all packets are delivered. In an event whereby Node A, for example, cannot get all the packets to Node D, then Node X acts as surety for complete packet delivery depending on the route table as repetitive packets are discarded. This would mean that node X is equipped with encrypted relay capabilities to Node D.

Node X acts as a Multi-Point Relay (MPR) node. MPR's are trusted nodes within the network to relay routing information to the intended destination. This would practically mean that node A

selects node X as an MPR then retransmits control packets from node A. In the network, each transmitting node could have one or more of these MPR nodes. These are nodes that are selected by their one-hop neighbours to retransmit all the broadcast messages they receive from a particular node provided that message is not a duplicate and that the message has a "Time to Live" field greater than one. Routes are selected by MPRs to avoid data packet transfer problems over uni-directional links. Each node will select its MPR set by using its one-hop neighbours. A group of MPR nodes is called an MPR set. The set is chosen such that it covers all symmetric two-hop nodes and a coverage strictly confined within the radio range. An MPR set of Node X is denoted as MPR (X). The other nodes within the network not selected as MPR process control packets as it is a broadcast environment but do not forward the packets. If Node A has selected node X and Y as its MPRs then it is safe to say:

$$\text{Node A: MPR(A)} = (X,Y) \tag{4.2}$$

Where X and Y represent node X and node Y, MPR (A) is the set of MPR nodes belonging to node A. The MPR nodes are selected based on a neighbour basis to the transmitting node. Each transmitting node uses HELLO messages to determine its MPR set. These HELLO messages are periodically broadcast to one-hop neighbours and not forwarded. Through the neighbour list in the HELLO messages received, nodes can determine two-hop neighbours and an MPR set. This MPR set is assigned a sequence number and the sequence number is incremented each time a new set is calculated. An MPR set is re-calculated when there is a change in one-hop or two-hop neighbourhood detected.

MPR nodes are the only ones allowed to generate and propagate Topology Control (TC) messages. The advertisement before sending TC messages is not sent to all links in the network. MPRs minimize the control traffic overhead of OLSR through retransmission of control messages. The technique significantly reduces the rate of transmissions needed to flood a message to all network nodes. The introduction of MPRs is to minimize the overhead of flooding messages and reducing redundant retransmissions in the network.

The encryption and decryption [109] methods may appear similar but in essence function differently and need to be separated. Rounds are several repeated AESs at a set number of times. Encryption has the following steps from round 0 till 9:

a) Byte substitution
b) Shift rows
c) Mix columns
d) Add Round key

In pseudo C notation, the above is derived as:

```
Round (State, RoundKey)
{
ByteSub(State);
ShiftRow(State);
MixColumn(State);
AddRoundKey (State, RoundKey);
}
```

For the last round execution (round set 10) for AES Encryption is presented in the following order as shown below:

a) Sub byte
b) Shift Row
c) Add Round Key

In pseudo C notation, we can represent it as:

```
FinalRound(State,RoundKey)
{
ByteSub(State) ;
ShiftRow(State) ;
AddRoundKey(State,RoundKey);
}
```

Decryption has the following steps for rounds 11 till 15:

a) Add Round key
b) Mix Columns

c) Shift columns
d) Byte substitution

For the last round of operation AES decryption has the following steps:

a) Inverse Shift Rows
b) Inverse Sub bytes
c) Add Round Key

Reducing the number of rounds performed may reduce power consumption but would harm the security of the protocol making it less secure. The 10 rounds of key expansion imposed on this work are done to strengthen the security of the protocol. Naturally, seven rounds or more can be considered fairly secure and energy-efficient.

4.4.2.1 AES Encryption

Figure 4.2 clearly shows the operations necessary for the encryption of data using AES. As mentioned earlier, the AES block cipher handles the computational aspect of the encryption process. A key and message come together to build scrambled data. The computation is lightweight and does not require the usage of a lot of network resources. The two most important steps for encryption are substitution and transposition.

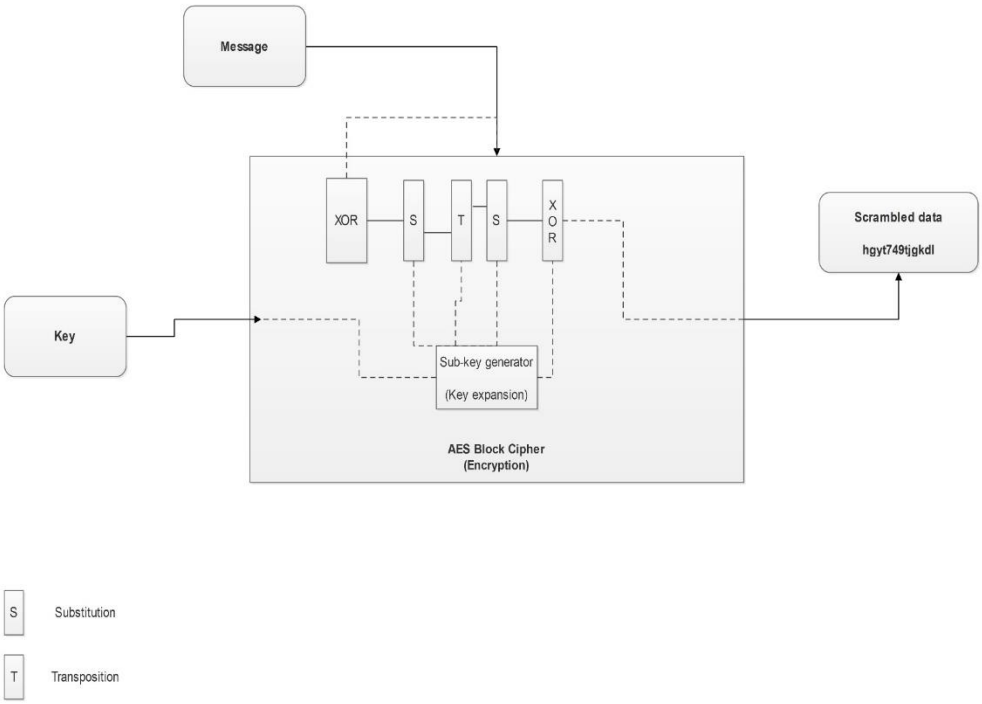

Figure 4.2: AES encryption [109]

4.4.2.2 AES Block Cipher

In an AES block cipher, there are many steps involved before the final realization of readable data/encrypted data. The first is key expansion whereby the key is subdivided into 44 subkeys. Each subkey has a size of 32 bits/4bytes/1 word. Each round has 4 subkeys. There are four stages inside the block cipher, namely the byte substitution, shift rows, mix columns and add round keys.

a) Byte Substitution: Each byte of the state is replaced with another byte depending on the key. This is normally guided by the Rijndael S-box lookup table. The Rijndael S-box is a substitution box used as the fundamental base of the AES architecture.

b) Shift Rows: The rows are shifted to the left. The general rule is that the first row is not shifted. The second row is shifted by one byte to the left. The third row is shifted by 2 bytes to the left and the last row is shifted by three bytes to the left respectively. As the bytes are shifted to the left, they appear on the right.

c) Mix Columns: the four bytes of each state are combined in a step by step manner. This step relies on the Shift Rows step and they both provide diffusion within the AES cipher. Each

column is multiplied by the known cipher using the XOR. The Mix Column is not performed on the last round though.

d) Add Round Key: It has two steps. There is the whitening step and the last step is the inner loop. Add Round key involves adding the State and the Round Key using special arithmetic. This arithmetic is called Addition Mod 2. The State and Round Key are both added as a Galois field. Additional Mod 2 happens to match an operation by the CPU. It performs arithmetic faster.



Figure 4.3: Step-wise AES encryption [110]

4.4.2.3 AES Decryption

Decryption undergoes similar steps as encryption but follows a different execution. In this instance, scrambled data is put into the cipher box together with the key. The key still gets to be broken down into 44 subkeys and computed. The output of this mechanism is the original message before it was encrypted. Decryption happens at the destination of the packets. In our case, it happens with

the trusted node and the destination node. The private key used for encryption is the same key used for decryption. The last round of execution involves an inverse shift of rows and inverse sub byte.

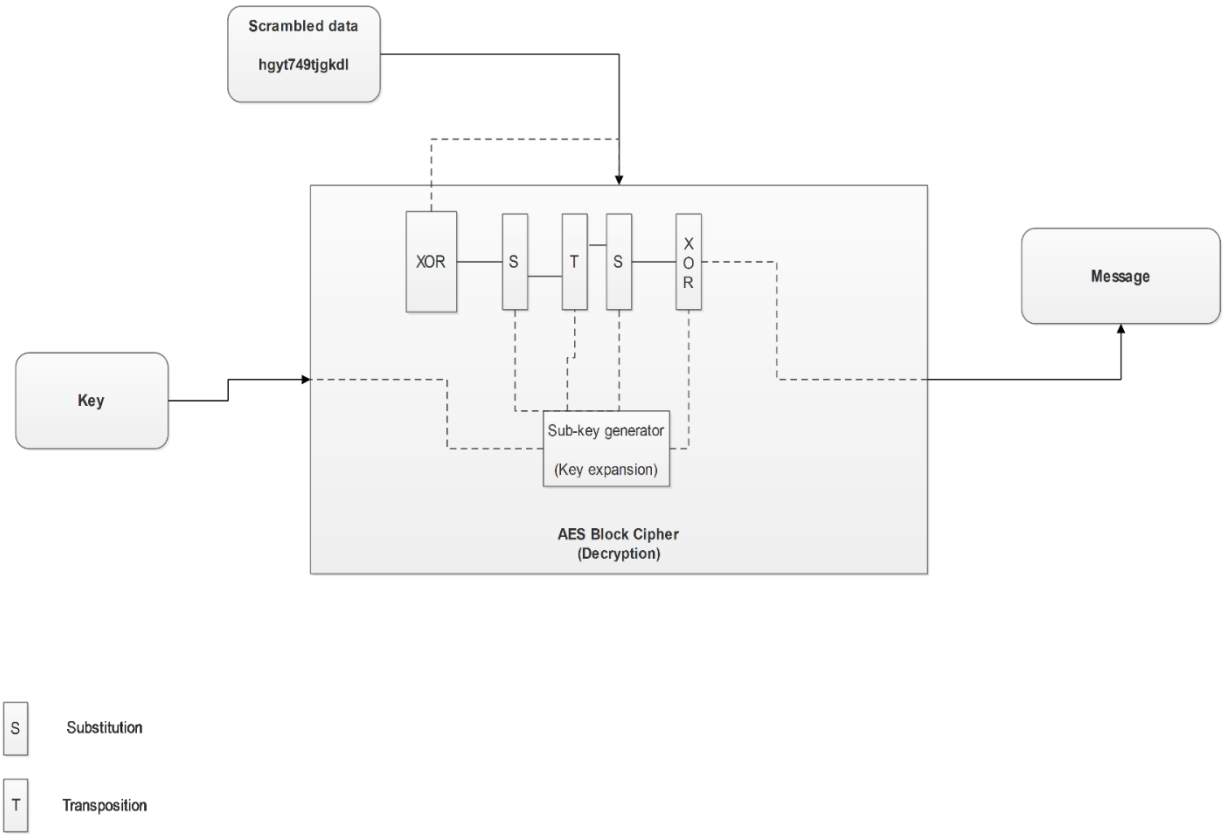

Figure 4.4: Decryption system [103]

4.4.3 Admission Control

The admission control mechanism deals with QoS violations as a result of false admissions, changes in node neighbourhood, node mobility or signal propagation conditions. This component is responsible for admitting data sessions with QoS requirements that can be satisfied without violating previous sessions. The collection of information about available network resources is done to make admission decisions. The admission control protocol establishes if there are any nodes or node pairs with the necessary resources available. This action is completed during route

discovery or even after the routes have been discovered. If no route completes the application's QoS requirements, then that data session is rejected. This is the reason why it is important during this phase, that information about the resources available with neighbouring nodes is collected. The key responsibility of this component is the management of data sessions with QoS-assurance violations and having those sessions re-admitted in an event when there was an interruption during its packet sending phase and maintenance of state information [111, 112].

4.4.4 Resource Reservation Scheme

In MANETs, the bandwidth is shared by nodes among neighbouring links, and its usage is managed by the routing protocol implemented. The Resource Reservation Scheme (RRS) can be implemented through QoS routing means or QoS MAC strategies. QoS routing schemes can be proactive or reactive. These are the known routing schemes like AODV, DSR and OLSR. The RRS is responsible for providing an acceptable level of QoS for high-priority sessions during routing by reserving resource usage at all immediate nodes along with the source to destination route. This is done so that applications like video or any form of multimedia have enough bandwidth during the transmission phase to satisfy an acceptable level of QoS requirements. The routing session's QoS is dependent on the bandwidth shared among the neighbouring links under the same medium and nodes within the interference range of the route. This is the reason why an RRS is implemented is so that state information on existing transmissions is protected within the route. In OLSR, the available resources are detected in advance as part of its proactive routing process, and all state information is constantly updated at the routing table. A secondary route is selected in an event where the primary route cannot satisfy QoS requirements because of link failure or insufficient resources. RRS is implemented to alleviate the impact of latency that comes with the routing process through the transmission of data from any source node because all the possible routes would have been identified proactively [113].

4.4.5 Transport Protocol

The framework is designed to accommodate both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic. The video streaming implemented in this work uses UDP as the preferred transport protocol. UDP uses the Internet Protocol (IP) to obtain datagrams from one node to another. It simplifies the need to have an established connection before data transmission.

Even although TCP is reliable and has assured delivery of packets, it is not favourable to video streaming and live conferencing applications since UDP embraces high performance and allows packets to be dropped instead of trying to process the delayed packets. Uneven delays between sections of a received message are not tolerated by real-time applications. UDP also saves bandwidth usage and is not latency-prone because there is no error checking. The overhead of a UDP packet is 8-bytes and that of TCP may vary from 20-60 bytes. It is implemented in this work because it also supports multicasting and packet switching. The application can trace route, record route and time stamp through UDP.

4.4.6 QoS-aware adaptive routing

The routing protocol, OLSR, facilitates all routing in the network. OLSR works using a link-state algorithm that constantly works with the routing table. The routing table is flexible to adaptive routing as the topology is dynamic and it needs constant updating. The packets will use an optimal path as directed by the MPRs to the destination. Nodes can exchange updates and route table information. Adaptive routing allows the routing path to change over time as the topology in which the nodes operate is ever-changing. Another role player/component in our framework is packet switching. Packet switching is regarded as a higher-level decision-making entity responsible for driving packets from source to destination.

The routing protocol, OLSR's flowchart in the proposed framework is shown in Figure 4.5 where we propose a few changes towards some of the traditional operations of the protocol to accommodate QoS and increase efficiency in terms of performance. Figure 4.5 shows the flowchart of our modified OLSR protocol. MPRs still play a critical role in terms of propagating TC messages through to the routing table. An un-authenticated node will be regarded as a malicious node and will be isolated from the network. At first, the node will be sent a fake HELLO message then selected as an MPR then blacklisted at the routing table. After routing, packets are sent to the data link layer where there is the packet queue management, MAC differentiator, and available bandwidth estimator. Under packet queue management, there is packet scheduling, priority scheduling, and packet forwarding.

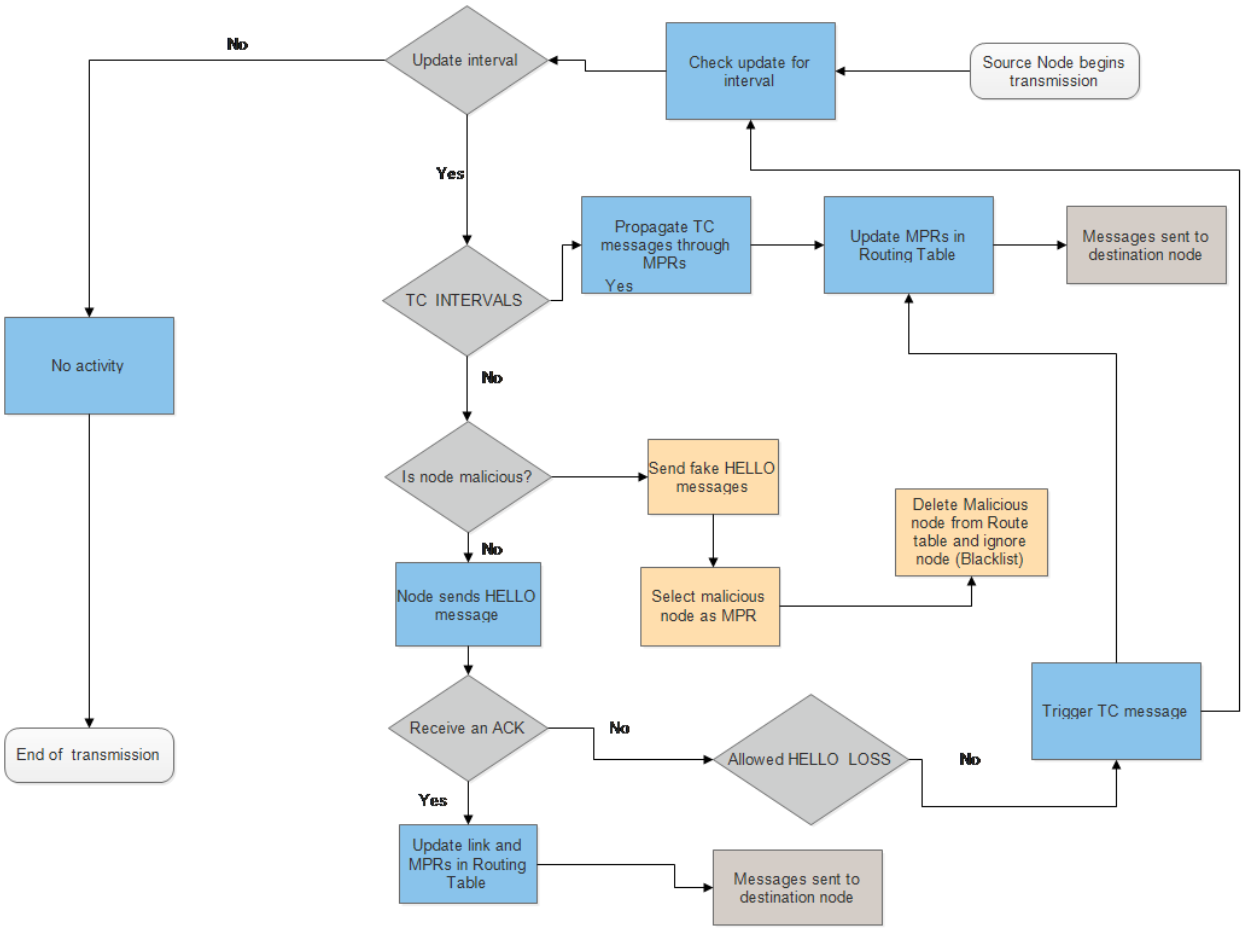

Figure 4.5: Modified OLSR flow chart [114]

4.4.7 Packet Queue Management

Admission control is responsible for preventing network overload to guarantee QoS for already admitted sessions and is deployed where a new session is either admitted when resources are available or rejected. Packet queue management is responsible for anticipating signal congestion before it occurs using scheduling means to determine the order or flow of different packets. The packet scheduler is responsible for providing the actual memory used for storing packets and providing a firewall used against malicious nodes whose intent is to selfishly use up network resources. The packet scheduler is responsible for determining the order in which packets are processed at the node level and transmitted over a link. This framework utilizes absolute priority queuing to assist in packet queue management. In absolute priority queuing, the packet scheduler scans and rates queues according to high, medium, and low priority. High priority is most preferred during the forwarding of packets from queues and this means that packets with lower priority get

processed as soon as higher priority queues are empty. In terms of resources, packets in high priority queues will receive the best service, and queues with a lower priority will have to be satisfied with the remaining resources. Once the packet is sent then the scheduler restarts the process again with high priority first. This mechanism is very important in terms of delay critical data and bandwidth usage for the type of traffic used which is video traffic in this instance. The packets are forwarded with minimal delay as long as the high priority packets do not exceed the outgoing traffic.

4.4.8 Sinkhole Attack

A sinkhole attack is orchestrated by modifying S. K. Suman's [115] source code. The modified sections are highlighted in all figures and algorithms respectively. This section makes a contribution to the study of sinkhole attacks and RFC 3626 OLSR as defined by the IETF Standards Process of November 10, 2008. This attack is launched against the network to disorganize the routing flow and modify packet information or even drop packets just to make the network inefficient in terms of performance. This work implements a sinkhole attack on the OLSR routing protocol. An *OLSR MALICIOUS.c* file containing a series of functions is added to the ZRP/OLSR project. The malicious node intercepts HELLO PACKETS sent by the source node by disguising itself as an MPR then adds a fake route entry and thus flooding the network with fake link-state information that is critical in determining the next hop to the destination. The malicious file added to the ZRP project has these notable source headers and functions as shown here:

```
#include "main.h"
#include "ZRP.h"
#include "OLSR.h"
#define MALICIOUS_NODE1 5
int fn_NetSim_OLSR_MaliciousNode ();
int fn_NetSim_Malicious1_OLSR_PopulateMPRSelectorSet ();
int fn_NetSim_Malicious1_OLSR_PacketProcessing ();
int fn_NetSim_Malicious1_OLSR_PacketForwarding ();
int fn_NetSim_Malicious1_OLSR_2HopTupleExpire ();
int fn_NetSim_OLSR_MaliciousRouteAddToCache(NetSim_EVENTDETAILS*);
int fn_NetSim_OLSR_MaliciousProcessSourceRouteOption(NetSim_EVENTDETAILS*);
```

4.4.8.1 fn_NetSim_OLSR_MaliciousNode ()

This function is placed to check if the defined node is malicious or not. If the node is malicious, then the function invokes malicious behaviour. Any node or device ID within the network can be set as malicious with the function:

```
int fn_NetSim_OLSR_MaliciousNode (NetSim_EVENTDETAILS* pstruEventDetails)
{
	if (pstruEventDetails->nDeviceId == MALICIOUS_NODE1)
	{
		return 1;
	}
	return 0;
}
```

4.4.8.2 fn_NetSim_Malicious1_OLSR_PopulateMPRSelectorSet ()

The function of an MPR selector set of a node, *n*, is populated by the main addresses of the nodes which have selected *n* as MPR. The selection of MPRs is sent through by HELLO messages intercepted by the malicious node in the function:

```
NODE_OLSR* olsr = GetOLSRData(pstruEventDetails->nDeviceId);
	NetSim_PACKET* packet = pstruEventDetails->pPacket++;
	OLSR_HEADER* header = (OLSR_HEADER*) packet->pstruNetworkData->Packet_RoutingProtocol;
	OLSR_HELLO_PACKET* hello = (OLSR_HELLO_PACKET*) header->message->MESSAGE;
	OLSR_HELLO_LINK* linkInfo = hello->link;
	double validityTime = olsrConvertMEToDouble(header->message->Vtime);
	while (linkInfo)
	{
		if (linkInfo->LinkCode.neighTypes == MPR_NEIGH)
		{
			OLSR_MPR_SELECTION_SET* set = olsrFindMPRSelectorSet(olsr->mprSelectionSet,
header->message->OriginatorAddress);
			if (!set)
			{
				set = MPR_SELECTION_SET_ALLOC ();
				LIST_ADD_LAST((void**) &olsr->mprSelectionSet, set);
				set->MS_main_addr = IP_COPY(header->message->OriginatorAddress);
			}
```

```
                    set->MS_time = pstruEventDetails->dEventTime + validityTime;
                }
                linkInfo = linkInfo->next;
        }
        return 0;
}
```

4.4.8.3 fn_NetSim_OLSR_MaliciousRouteAddToCache ()

This function is used by the malicious node that has selected itself as an MPR to add a fake route entry into the route cache before flooding the network with fake link-state information.

```
int fn_NetSim_OLSR_MaliciousRouteAddToCache (NetSim_EVENTDETAILS* pstruEventDetails)
{
        NODE_OLSR* olsr = GetOLSRData(pstruEventDetails->nDeviceId);
        NETSIM_IPAddress subnet = STR_TO_IP4("255.255.255.255");
        OLSR_NEIGHBOUR_SET* neighbour;
        OLSR_2HOP_NEIGHBOUR_SET* two_hop_neighbour;
        OLSR_TOPOLOGY_INFORMATION_BASE* topology;

OLSR_HEADER*header=
(OLSR_HEADER*)pstruEventDetails->pPacket->pstruNetworkData->Packet_RoutingProtocol;

NETSIM_ID interfaceId = 5;
iptable_add(olsr->ipWrapper,two_hop_neighbour->N_2hop_addr,subnet,0,two_hop_neighbour->N_neighbour_mai
n_addr,1,
        &olsr->mainAddress,
        &interfaceId,
        2,
        "OLSR");

        return 1;
}
```

4.4.8.4 fn_NetSim_OLSR_MaliciousProcessSourceRouteOption ()

This function is used to drop packets instead of the node forwarding them to the next hop:

```
int fn_NetSim_OLSR_MaliciousProcessSourceRouteOption (NetSim_EVENTDETAILS* pstruEventDetails)
{
	NetSim_PACKET* packet = pstruEventDetails->pPacket;
	//update the metrics
	OLSR_DEV_VAR(pstruEventDetails->nDeviceId)->OLSRMetrics.packetReceived++;
	fn_NetSim_Packet_FreePacket(pstruEventDetails->pPacket);
	return 0;
}
```

4.4.9 Intrusion Detection System

An intrusion detection system is implemented by modifying S. K. Suman's [115] source code to detect and remove a sinkhole attack from the network. The modified sections are explained and highlighted in all figures. According to Kumat and Venugopalan [116], intrusion detection is the process of capturing and analyzing significant events in a system for its possible presence. Intrusion detection systems aim to prevent unauthorized access in the network and also responding to the malicious activities that may be caused by the intruder. Researchers have proposed various approaches to implementing an IDS based on the idea of wired networks that have routers and switches. This makes it difficult for MANETs as they do not have all the mentioned devices but all the devices are connected through an open wireless medium.

The implemented standalone intrusion detection system is customized to function in the two layers of the TCP/IP model namely the network and data link layer running the OLSR routing protocol. It uses two mechanisms to achieve its goal and is intended to work for UDP traffic because TCP requires ACKs from the destination. The Watchdog analyzes every node that is in transmission proximity. It compares two variables (threshold value and counter) before detecting that the node is showing malicious behaviour. If a node withholds a packet, the watchdog algorithm counts the downtime for the delay of the node to forward the message and compares this with the threshold value. The delay has to be less than the threshold value for the node to be considered cooperative or else the node will be considered malicious. The watchdog will also check the packet itself for alterations and possible modifications. If any node only accepts the watchdog node and does not forward, then the watchdog declares it as a sinkhole node and isolates that node from the path of packet transmission. A sinkhole node has an infinite ratio of receiving and sending data packets[117]. Figure 4.6 shows the flow chart of the intruder detection system:

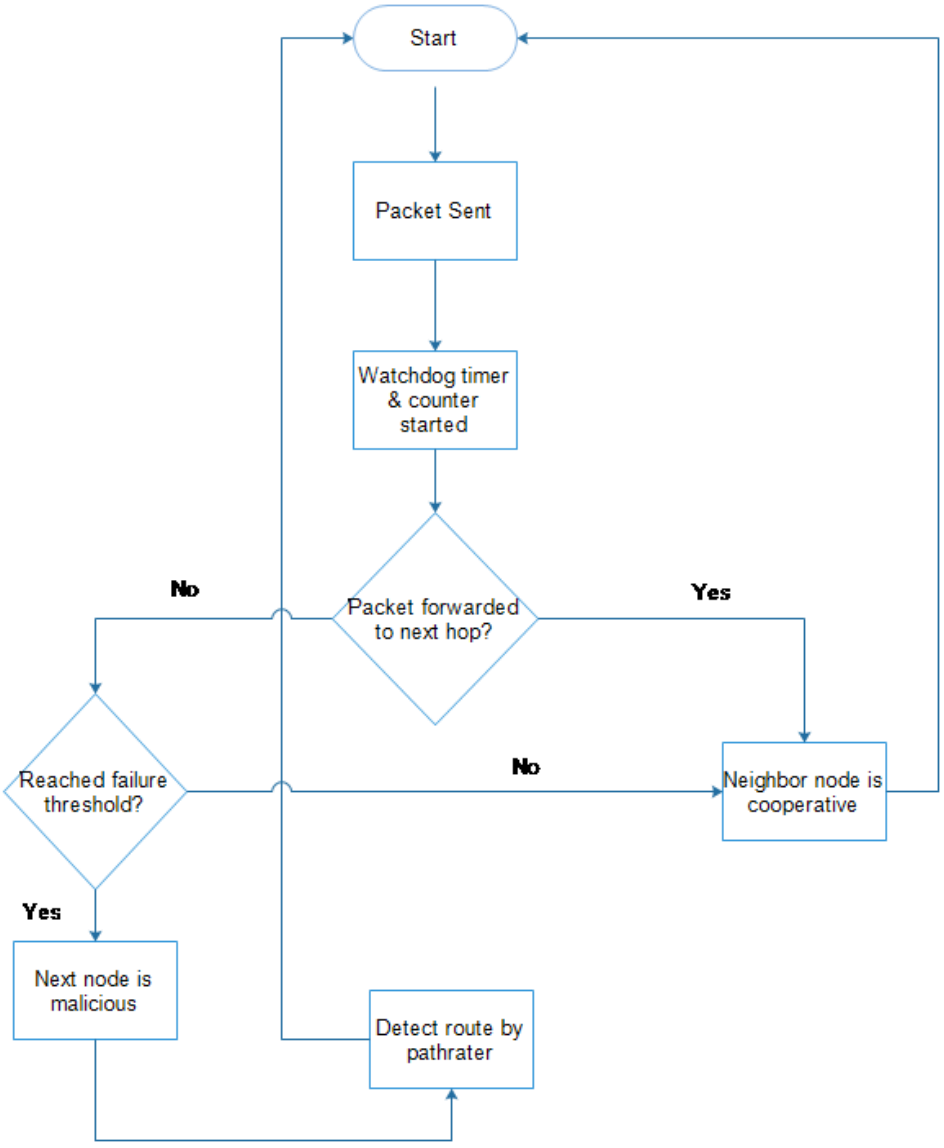

Figure 4.6: IDS Workflow [117]

4.4.9.1 Watchdog

The Watchdog mechanism is implemented in the WLAN IEEE802_11 project folder and is a key method for the detection of intrusions in the network. It is considered simplistic and lightweight because it works with less overhead by using embedded local information handling capabilities. In this code, (*WATCHDOG_TIME*) a watchdog timer is set for 2 seconds. Users can modify it for any routing protocol. The nodes wait for the duration of this timer to detect if the next-hop node is malicious or not by checking if it broadcasts the packet further to the destination node. The failure

threshold is set at 5. The failure threshold is the number of times a packet is resent before declaring it malicious.

4.4.9.2 Watchdog code flow

Once the _NETSIM_WATCHDOG _ is defined, the watchdog timer begins. When a packet is sent to a neighbouring hop, the current node checks the watchdog timer duration to verify if the packet is forwarded to its destination. If the node is malicious, then it will not forward the packets it receives until the watchdog timer in the node expires. Once a packet is forwarded to the next-hop node, the current node checks for watchdog timer duration if the packet is getting forwarded further to the destination node or not. The packet is then sent until the watchdog timer expires and the failure threshold is reached. There is a counter that measures the frequency at which the watchdog timer expires and once the counter's value equates to the failure threshold then the next hop is marked by the current node as malicious. As highlighted in the QoS framework malicious nodes are blacklisted and removed to maintain the integrity of the network. The function *add_to_blacklist* (*NETSIM_ID, NETSIM_IPAddress*) is responsible for adding the intruder's IP address to the blacklist. The function *find_ip_in_blacklist* (*NETSIM_ID,NETSIM_IPAddress*) returns true if the node is marked as blacklisted. The function *verify_route_reply* (*NETSIM_ID, OLSR_HELLO_PACKET**) verifies the IP address obtained from the control packet to check if it is blacklisted or not. If the reply is not from the malicious node, then it returns true. The function *blacklist_found*(*NETSIM_ID ,NETSIM_ID*) deletes the node's route entry from the cache when it is blacklisted [115].

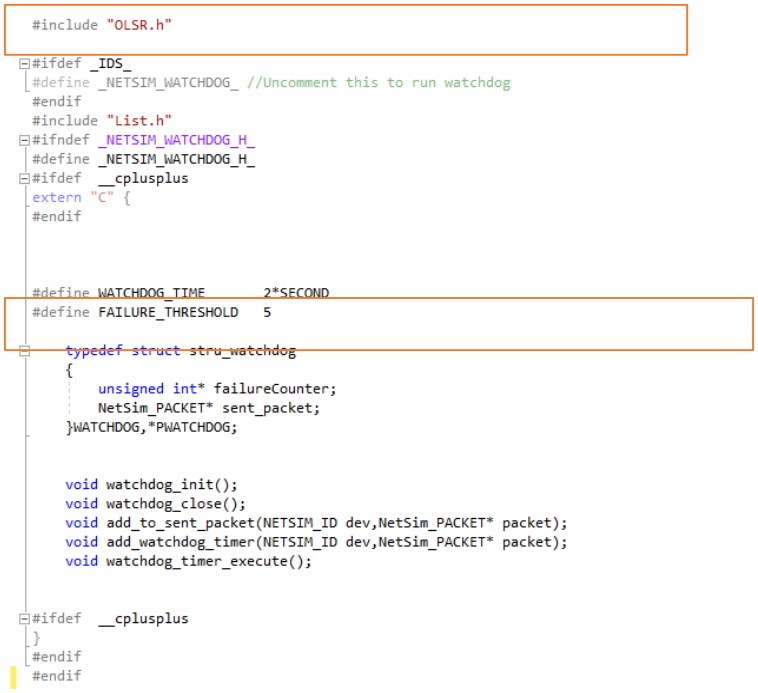

Figure 4.7: Modified Watchdog declaration code [115]

a) Watchdog Initiation

```
void watchdog_init()
{
	NETSIM_ID i;
	watchdog = (PWATCHDOG*)calloc(NETWORK->nDeviceCount+1, sizeof* watchdog);
	for(i=1;i<=NETWORK->nDeviceCount;i++)
	{
		watchdog[i] = (PWATCHDOG)calloc(1,sizeof* watchdog[i]);
		watchdog[i]->failureCounter = (unsigned int*)calloc(NETWORK->nDeviceCount+1,sizeof* watchdog[i]->failureCounter);
	}
}
```

b) Watchdog Execution

```
void watchdog_timer_execute()
{
	NetSim_PACKET* packet = pstruEventDetails->pPacket;
	NETSIM_ID dev = pstruEventDetails->nDeviceId;
	NETSIM_ID rcv = packet->nReceiverId;
	if(!check_in_sent_list(dev,packet))
	{
```

```
                PWATCHDOG dog=watchdog[dev];
                dog->failureCounter[rcv]++;
                if(dog->failureCounter[rcv] >= FAILURE_THRESHOLD)
                {

                        dog->failureCounter[rcv] = 0;
                }
        }
        else
        {
                PWATCHDOG dog=watchdog[dev];
                dog->failureCounter[rcv]=0;
        }
        fn_NetSim_Packet_FreePacket(packet);
}
```

4.4.10 Pathrater

The Pathrater file is added in the ZRP/OLSR project folder and its function is to validate routes. It runs in the network layer hence it is added to this particular project folder. In an event where a HELLO packet is processed, the function verifies the response in the route cache to check for the blacklisted node. If a blacklisted node is found, that route entry is deleted from the cache. Figure 4.8 shows Pathrater code definitions under the file *Pathrater. h* and Figure 4.9 show the execution code integration with the OLSR routing protocol under the file *Pathrater. h.*

```
#ifdef _IDS_
#define _NETSIM_PATHRATER_ //Uncomment this to run pathrater
#endif


#include "List.h"
#ifndef _NETSIM_PATHRATER_H_
#define _NETSIM_PATHRATER_H_
#ifdef  __cplusplus
extern "C" {
#endif

    typedef struct stru_blacklist
    {
        NETSIM_IPAddress ip;
        _ele* ele;
    }BLACKLIST,*PBLACKLIST;
#define BLACKLIST_ALLOC() (PBLACKLIST)list_alloc(sizeof(BLACKLIST),offsetof(BLACKLIST,ele))
#define BLACKLIST_NEXT(_blacklist) _blacklist=(PBLACKLIST)LIST_NEXT(_blacklist)
#define BLACKLIST_FREE_ALL(_blacklist) while(_blacklist) LIST_FREE((void**)&_blacklist,_blacklist)
#define BLACKLIST_ADD(_list,_black) LIST_ADD_LAST((void**)&_list,_black)
#define OLSR_DELETE_ENTRY_CACHE

    void pathrater_init();


    void pathrater_close();


    void add_to_blacklist(NETSIM_ID dev_id,NETSIM_IPAddress ip);

    bool verify_route_reply(NETSIM_ID dev_id);

    bool is_malicious(NETSIM_ID dev,double time);
    int Malicious_AddRouteToCache(NetSim_EVENTDETAILS* pstruEventDetails);
    int Malicious_DropPacket(NetSim_EVENTDETAILS* pstruEventDetails);
    int cache;

#ifdef  __cplusplus
}
#endif
#endif
```

Figure 4.8: Modified Pathrater file declaration [115]

```
#include "main.h"
#include "Pathrater.h"
#include "OLSR.h"

PBLACKLIST* blacklist;
//Initializes the pathrater.
void pathrater_init() { ... }
//Closes the pathrater.
void pathrater_close() { ... }
//Add malicious node to black list
void add_to_blacklist(NETSIM_ID dev_id,NETSIM_IPAddress ip) { ... }
//Check if the device is blacklisted
bool find_ip_in_blacklist(NETSIM_ID dev_id,NETSIM_IPAddress ip)
{
    PBLACKLIST black=blacklist[dev_id];
    while(black) { ... }
    return false;
}
//Verify route reply
bool verify_route_reply(NETSIM_ID dev_id,OLSR_HELLO_PACKET* hello) { ... }
//If node is blacklisted, delete route entry
_declspec(dllexport) void blacklist_found(NETSIM_ID dev_id,NETSIM_ID black_id) { ... }
```

Figure 4.9: Modified Pathrater file functions for OLSR [115]

4.10.2.1 add_to_blacklist (NETSIM_ID, NETSIM_IPAddress)

This function is responsible for adding the IP address of the sinkhole node to a blacklist and it is noted as a possible intruder.

```
void add_to_blacklist(NETSIM_ID dev_id,NETSIM_IPAddress ip)
{
        PBLACKLIST black = BLACKLIST_ALLOC();
        black->ip=IP_COPY(ip);
        BLACKLIST_ADD(blacklist[dev_id],black);
}
```

4.10.2.2 find_ip_in_blacklist(NETSIM_ID,NETSIM_IPAddress)

This function returns true if the suspected node is blacklisted:

```
bool find_ip_in_blacklist (NETSIM_ID dev_id,NETSIM_IPAddress ip)
```

```
{
        PBLACKLIST black=blacklist[dev_id];
        while(black)
        {
                if(!IP_COMPARE(black->ip,ip))
                        return true;
                BLACKLIST_NEXT(black);
        }
        return false;
}
```

4.10.2.3 add_to_blacklist (NETSIM_ID, NETSIM_IPAddress)

This function verifies the IP address obtained from the OLSR control packet. This takes account of whether it is blacklisted or not. It returns true if the OLSR packet is not from the sinkhole node as shown in Figure 4.10.

```
//Verify route reply
bool verify_route_reply(NETSIM_ID dev_id,OLSR_HELLO_PACKET* hello)
{
    struct Address;
    int len = OLSR_CONTROL_PACKET(hello);
    int i;
    bool ret = false;
    for(i=0;i<len;i++)
    {
        ret = find_ip_in_blacklist(dev_id,hello);
        if(ret)
            return false;
    }
    return true;
}
```

Figure 4.10: Modified route reply [115]

4.10.2.4 blacklist_found(NETSIM_ID ,NETSIM_ID)

This function removes the sinkhole node's route entry from the cache:

```
//If node is blacklisted, delete route entry
_declspec(dllexport) void blacklist_found(NETSIM_ID dev_id,NETSIM_ID black_id)
{
    int n = 0;
    NETSIM_IPAddress black_ip = (black_id);
    NETSIM_IPAddress my_ip = (dev_id);

    //Add to blacklist
    add_to_blacklist(dev_id,black_ip);
#ifdef _IDS_
    for (n = 0; n<(mal_dev_count); n++)
    {
        if (mal_devid[n] == dev_id+1 && mal_det_time[n] == 0)
            mal_det_time[n] = pstruEventDetails->dEventTime;//-mal_start_time[n];
    }
#endif

    //Delete route from route cache
    OLSR_DELETE_ENTRY_CACHE(cache,my_ip,black_ip);
}
```

Figure 4.10: Modified deletion of the route from cache [115]

4.5 Chapter Summary

This chapter addressed some of the operational choices made in this work and a stepwise approach to our framework designed to improve QoS and detect sinkhole attacks. It also provided solutions to RQ3, RQ4 and ultimately met RO3, RO4, and RO5. This framework also heightened the importance of QoS and security in mobile wireless systems. The framework also contributed to the evaluation of sinkhole attacks in MANETs as literature does not specifically address the extent to which sinkhole attacks affect OLSR routing protocol. A further extension of the AES algorithm was brought in as the security aspect of our work. The AES algorithm matches the design requirement of a lightweight architecture but at the same time is effective. A customized flow chart of the OLSR algorithm was effected as part of this work's contribution to making a secure and adaptive routing protocol. A contribution was also brought in through the introduction of a sinkhole attack in OLSR where a malicious code was integrated. The last contribution that

addresses our framework implementation was a custom intrusion detection mechanism designed to detect and remove sinkhole attacks. This was done to improve the flow of packets in different node environments. This detection system was implemented at the network and data link layer using a Watchdog and Pathrater. The next chapter gives detailed findings from the conducted evaluation experiments.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Chapter Overview

In this chapter, the results of the simulations from the sinkhole attack and the proposed framework to fulfil RO4 are discussed. The generated results are a direct product of the evaluation set-up discussed in Chapter 4. The results firstly, present the network model for each simulation before the evaluation of performance metrics. This is followed by a comparison of performance metrics with the sinkhole attack and the framework for throughput, delay and jitter. The simulations were broken down into different scenarios. The scenarios were set according to different node densities at a period of 100 seconds each. Each scenario has four experiments running with 16, 49, 64 and 100 nodes. These different nodes represent performance in low, mid, and high node density environments. For all the simulations, NetSim Standard was used. NetSim Standard is an advanced version of NetSim that allows users to manipulate the source and recreate custom protocols and functions. These results were then compared for efficiency and to justify the use of the proposed framework. At the end of this chapter, the results we obtained were presented in the form of properly labelled graphs.

5.2 Comparison of Results

5.2.1 Simulation 1: Performance evaluation under low-density node scenario (16 nodes)

The first simulation results showed the performance scenarios with 16 nodes exposed to a sinkhole attack and with the proposed framework. Figure 5.1 shows the network model used in different scenarios. The first scenario conducted was without an attack or the framework. This was done to have a complete perspective as to the extent of the attack in the network. The results of the scenarios clearly showed that the proposed framework outperforms malicious scenarios by a huge margin in low node density environments. The three metrics considered in the performance level of the framework are discussed:

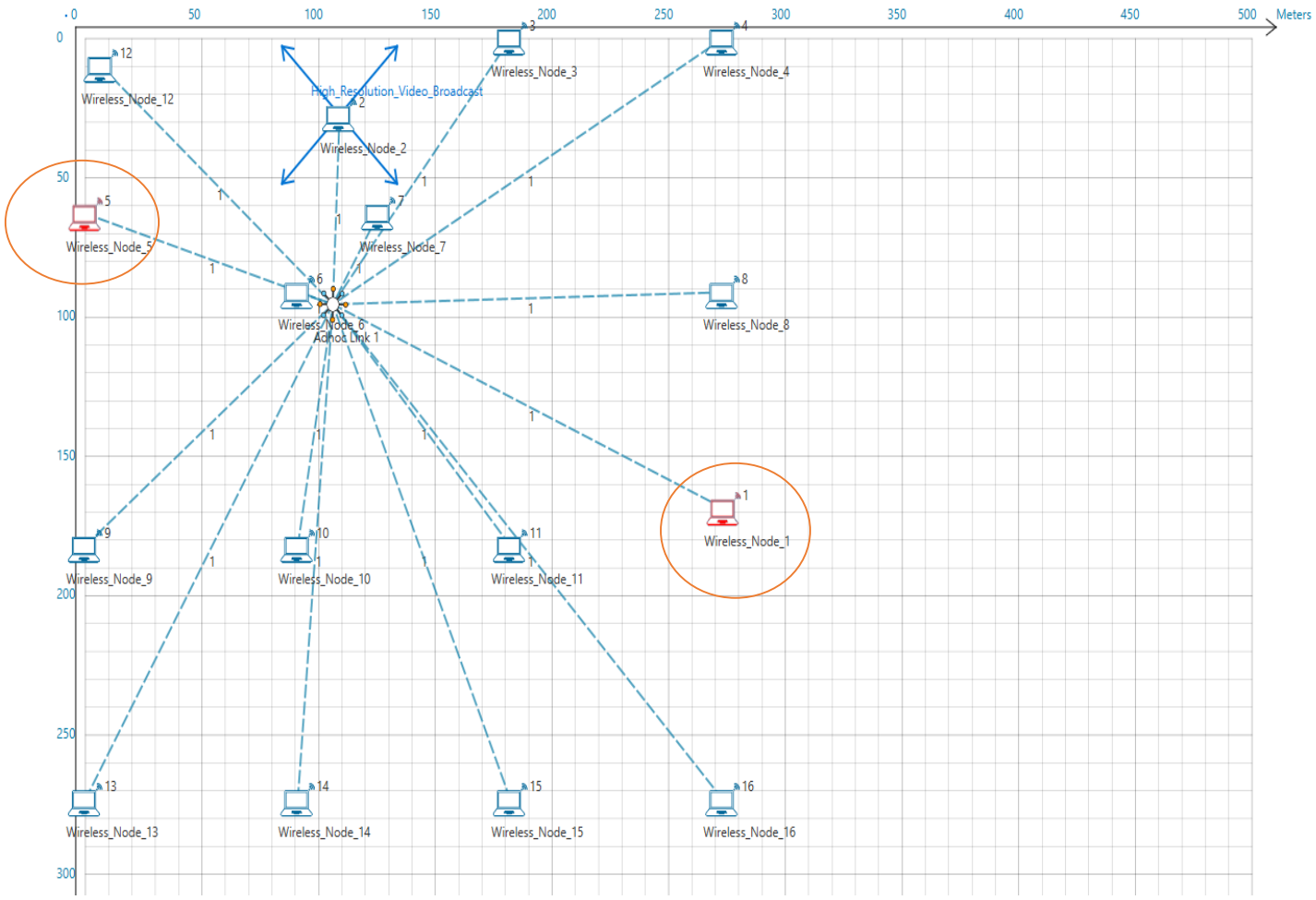

Figure 5.1: Network model with sinkhole nodes

5.2.1.1 Throughput performance analysis

Figure 5.2 shows the throughput for both sinkhole and framework scenarios. The Y-axis shows the throughput in Megabits per second (Mbps) and X-axis shows the time in milliseconds (ms). In the sinkhole scenario, there was a 2000 ms lag time at which the application starts. The next 14000 ms showed a positive hike up to 0.74 Mbps in throughput after which there was a steady increase until the end of the simulation, unlike the framework scenario which grew with time. The highest network throughput achieved by the framework was 0.77 Mbps at 68806.37 ms of simulation time. OLSR is known to be a routing protocol most suited for dense network environments.

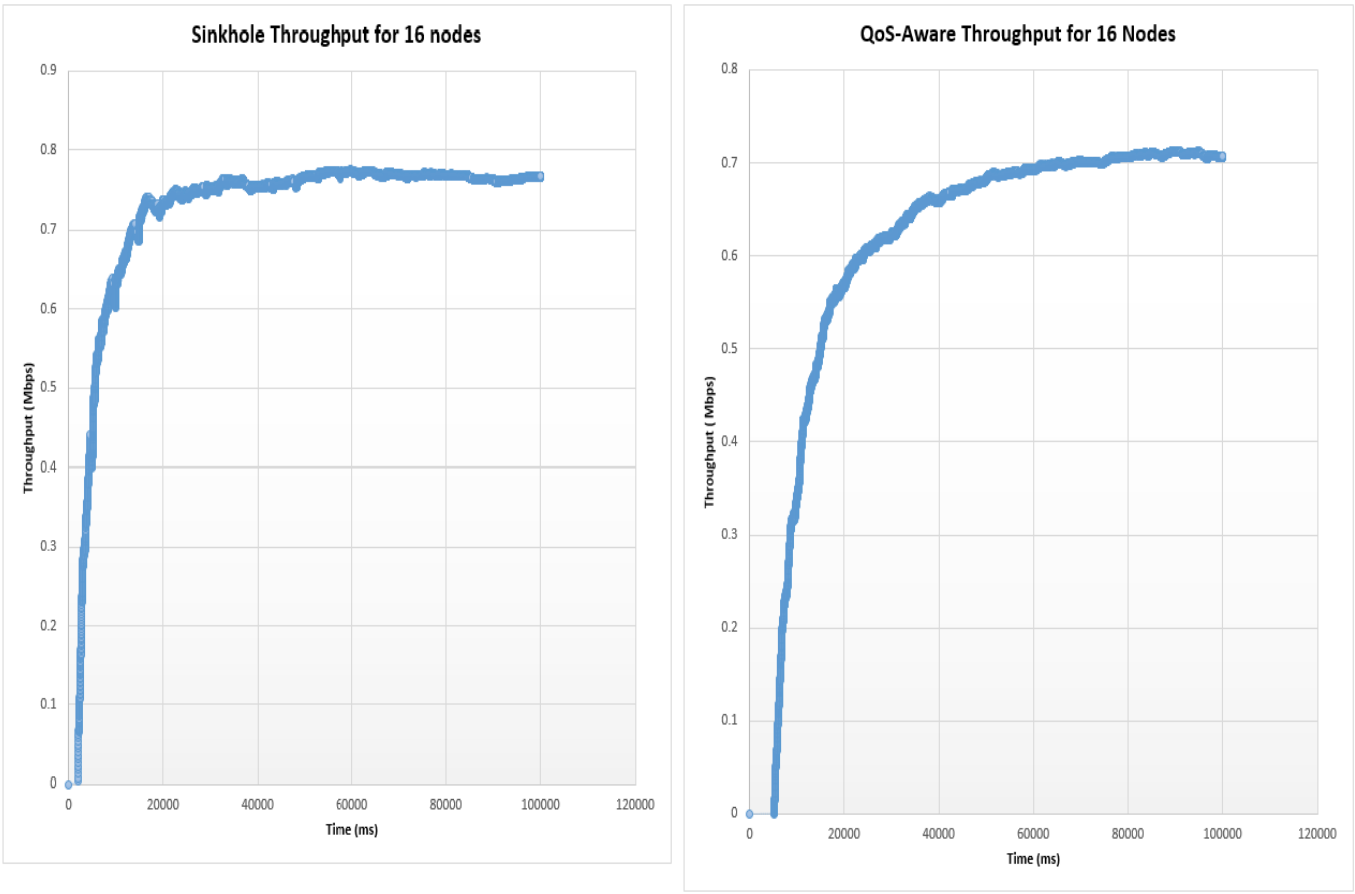

Figure 5.2: Throughput for sinkhole and framework scenarios

5.2.1.2 Delay performance analysis

The proposed QoS framework had the lowest recorded value in terms of network delay and as expected the malicious scenario produced the highest level of delay for any 16-node scenario as shown in Figure 5.3. The delay in the sinkhole scenario was caused by the time the attack exceeds the failure threshold value before forwarding the packets to the rest of the network. The framework significantly reversed this action by a huge margin meaning that the intrusion was identified, blacklisted, and alternative routes that guarantee an acceptable level of QoS were chosen. In the framework scenario, the first node had a delay of 12369.96 microseconds. Node 3 had an elevated delay value of 13739.96 microseconds. The delay slightly fluctuated between 13663.037 and 13739.96 microseconds for Nodes 7 and 12. The rest of the nodes maintained a constant delay of 12017.41 microseconds. The delay was relatively low as compared to that of the sinkhole scenario.

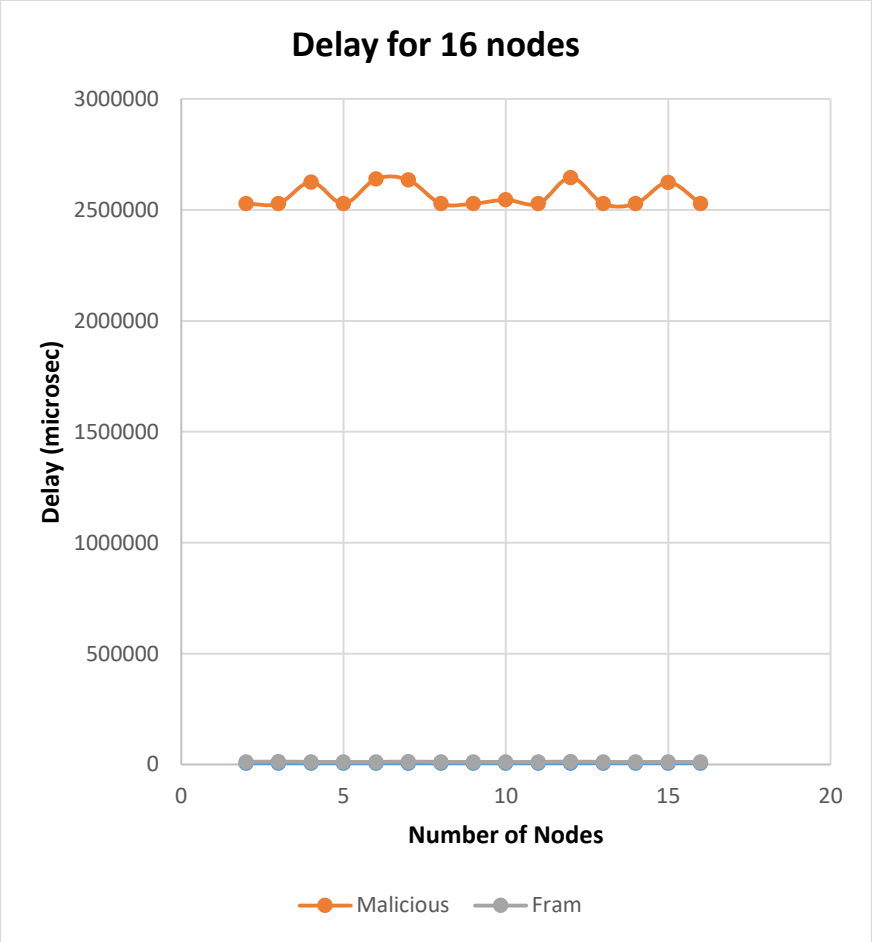

Figure 5.3: Delay for 100 nodes

5.2.1.3 Jitter performance analysis

The proposed QoS framework outperformed the malicious scenario again in terms of jitter recording an average of about 6500 microseconds as compared to the malicious scenario which had margins of about 35000 microseconds. In the sinkhole scenario, the first node had a delay of 26543.9 microseconds. Node 2 to Node 5 showed a constant jitter of 26468.72 microseconds but Node 6 and Node 12 both had 26400.87 microseconds. This was an expected margin based on the malicious activity within this broadcast session. As shown in Figure 5.4, the framework significantly lowered the levels of jitter among all the nodes. The framework proved to be effective in low-density node scenarios.

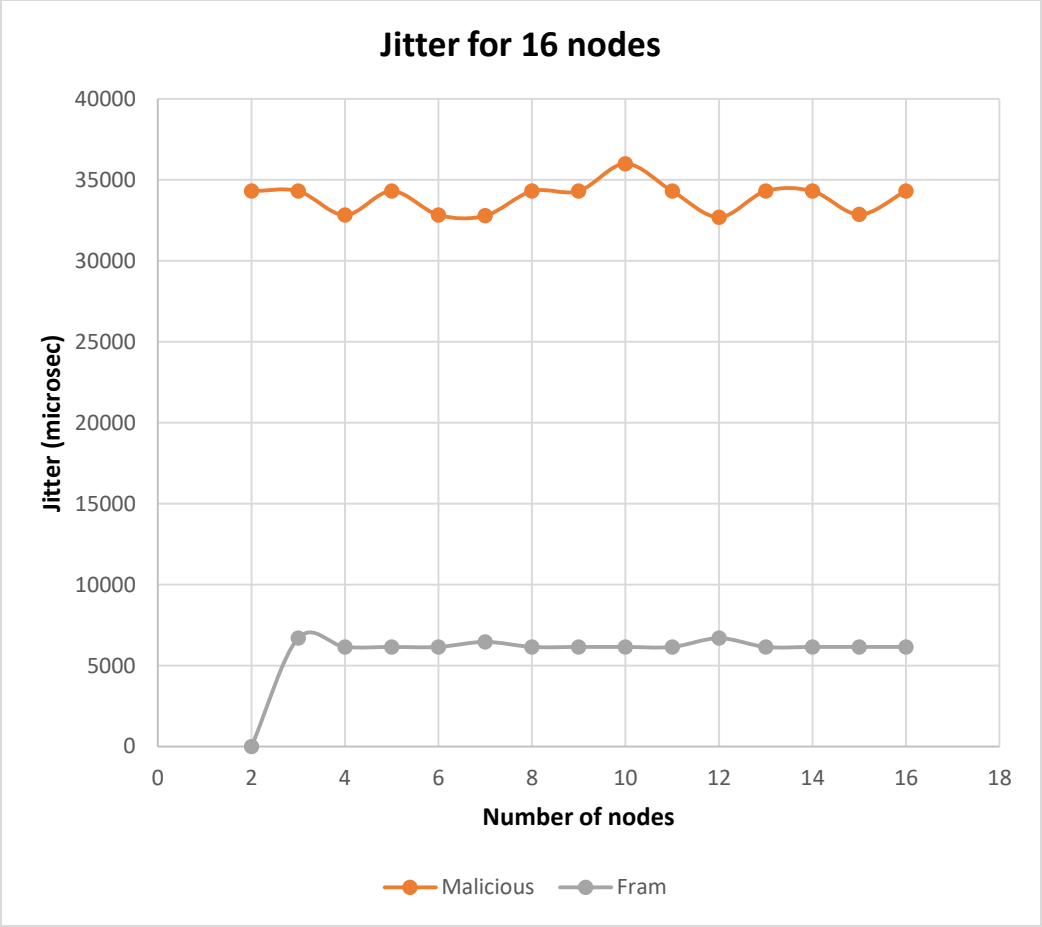

Figure 5.4: Jitter for 16 nodes

5.2.2 Simulation 2: Performance evaluation under average density nodes scenario (49 nodes)

The second simulation results showed the performance of the network with 49 nodes exposed to a sinkhole attack and with the proposed framework. Again, there was the involvement of a normal scenario to determine the extent of the malicious behaviour and also the counterattack measures that were taken. The proposed framework still outperformed the malicious node scenarios based on the given throughput but the result was not satisfactory as there were high margins in terms of delay and jitter. Figure 5.5 shows the implemented network model with 49 mobile nodes and two marked malicious nodes. The experiment was run 10 times for validation of findings and in all instances, the results in Figure 5.6-5.8 were obtained.

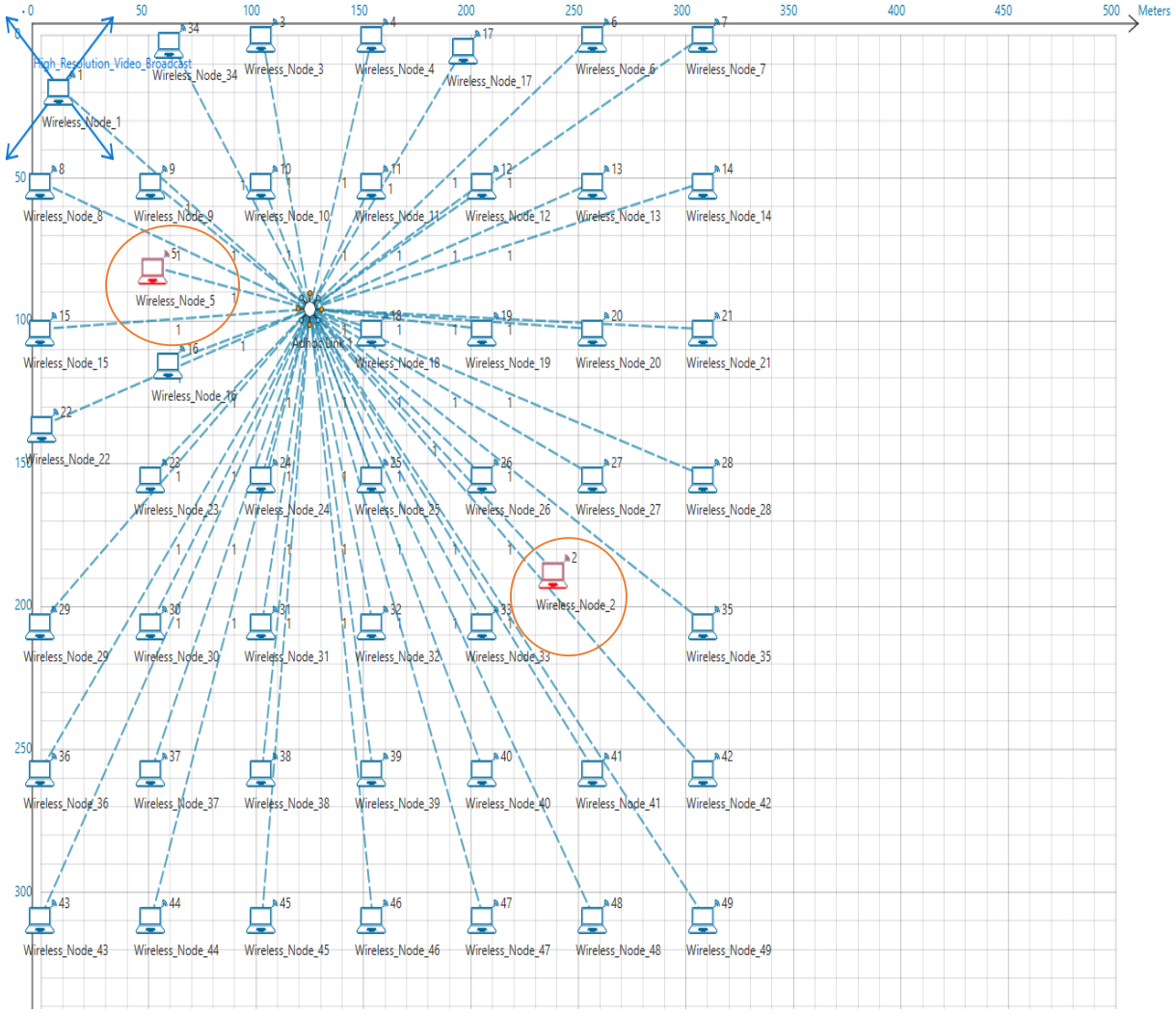

Figure 5.5: Network model with sinkhole nodes

5.2.2.1 Throughput performance analysis

Figure 5.6 shows the throughput for both sinkhole and framework scenarios as the first 5000 milliseconds were set to be the time at which the application starts. In the sinkhole scenario, the next 8833 milliseconds showed positive results up to 1.522 Mbps in throughput followed by a steady decrease to 0.88 Megabits/sec. The highest network throughput achieved was 1.525Mbps at 13604.24 ms of simulation time. The scenario's output significantly dropped after 10000 ms. The framework on the other hand had a positive hike in figures throughout with a steady increase

until the end of the simulation. The highest network throughput achieved was 2.37Mbps at 94282.78ms. The framework scenario's throughput grew over time, unlike the sinkhole scenario which dropped performance just after a tenth of the simulation time.

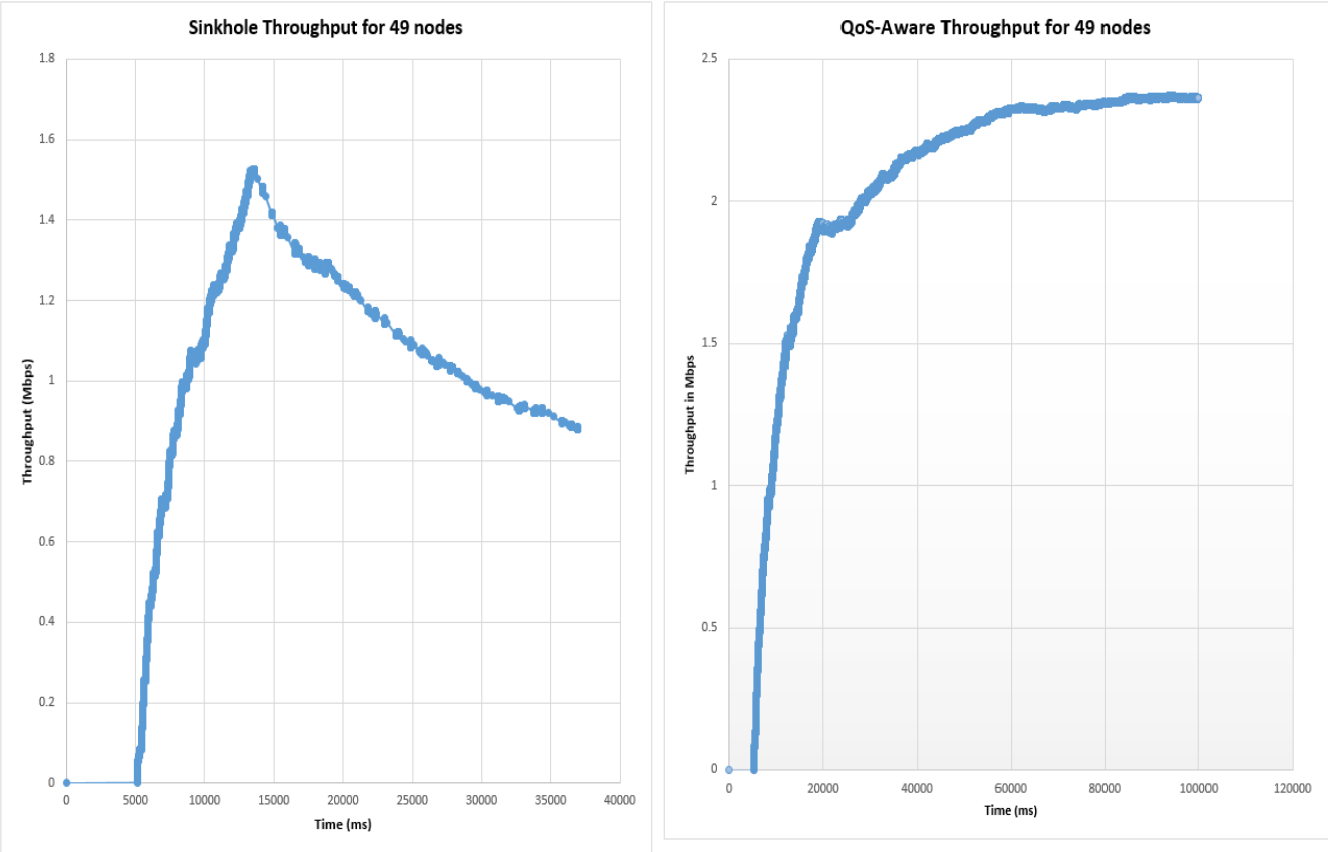

Figure 5.6: Throughput for sinkhole and framework scenarios

5.2.2.2 Delay performance analysis

The results of the scenarios show a relatively high delay value for the framework as compared to the other scenarios in Figure 5.7. This margin is proportional to the throughput of the network which is higher than the other scenario. The presented result proved that the employed practices were very effective. In the malicious scenario, all the other nodes show a constant delay of 2137687.396 microseconds except Node 14 which has a delay of 2771719.28 microseconds. The margin shown in Figure 5.7 is extremely high compared to the framework whose delay for almost

all the nodes was around 40000 microseconds. The delay increases as many nodes were included in the network but the framework could sustain such effects to provide an acceptable level of QoS. The two scenarios have an opposite outlook in terms of their delay propagation. The growth in the delay is, however, normal as the network grows but the optimization of OLSR and the cross-layer interaction between the network, physical, and MAC layers play a huge role in making the network more delay-sensitive and efficient.

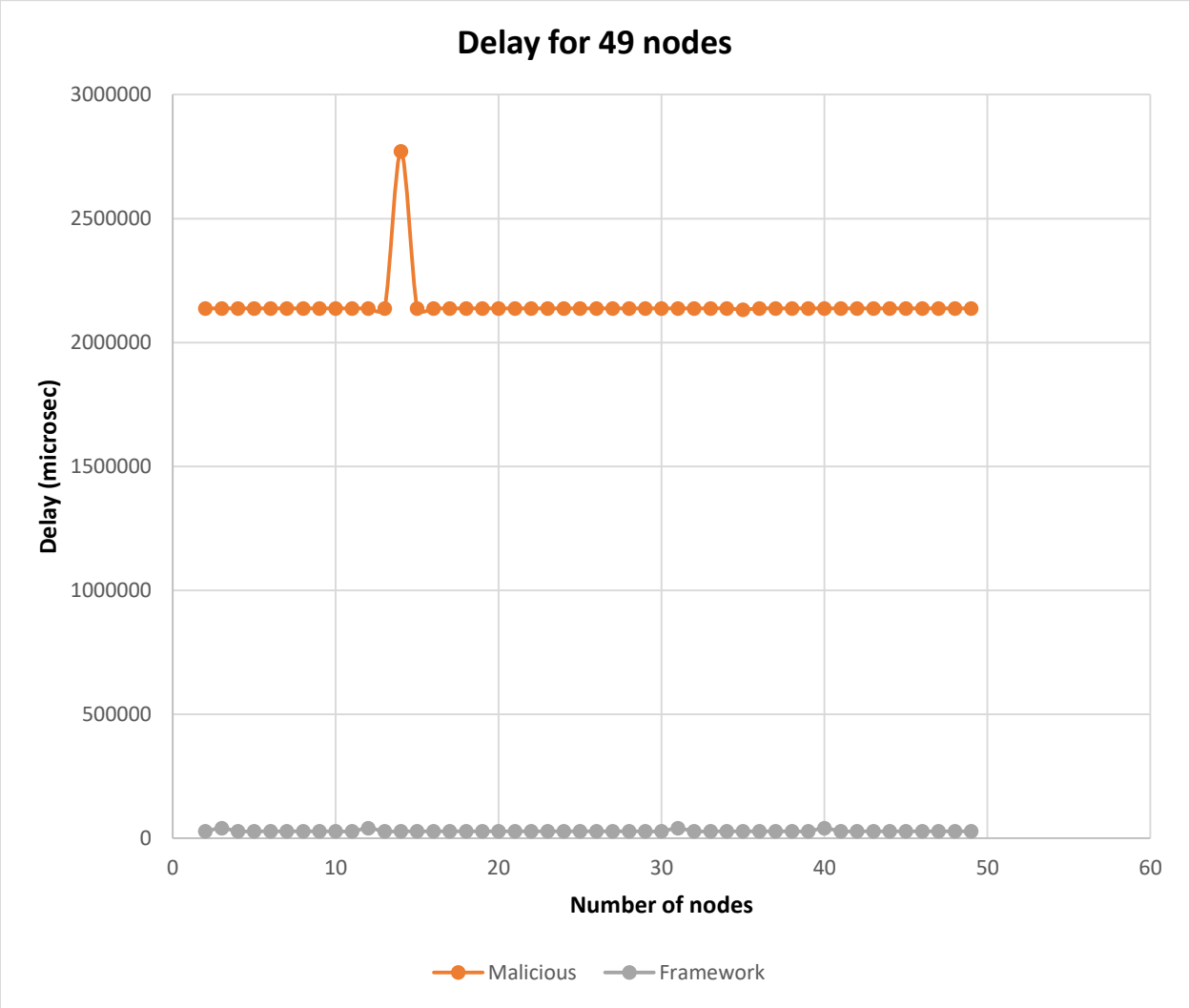

Figure 5.7: Delay for 49 nodes

5.2.2.3 Discussion for Jitter

The results of the scenarios in figure 5.8 show a rise in jitter levels for the two scenarios as compared to when there were 49 nodes in the network. The malicious scenario had relatively much higher levels of jitter than in the proposed framework. This result is strongly attributed to the framework's key components like admission control, resource reservation, and the IDS. In the normal scenario, the first two nodes show a steady hike in jitter from 0 microseconds until it reaches 101938.83 microseconds. The 11 nodes maintain a constant level of 101938.83 microseconds but node 14 dropped to 78282.15 microseconds. The rest of the nodes maintain this level until the end of the application. The highest recorded jitter level remained at 101938.83 microseconds which was fairly normal based on the number of nodes in the network. This scenario was not part of the comparison between the sinkhole and framework scenarios but rather acted as a litmus to the two scenarios.

In the malicious scenario, the first two nodes show a steady hike in jitter from 0 microseconds until it reached 119335.24 microseconds. The rest of the nodes maintained this level until the end of the application except node 14 which had a jitter level of 78282.16 microseconds. The framework scenario outperformed the sinkhole scenario and by reducing the jitter levels to almost four times lower than when the sinkhole nodes were in action. The framework's route selection mechanism was well equipped to deal with various changes in the route or queuing network. The prioritization of packets by the framework further helped in reducing network congestion.

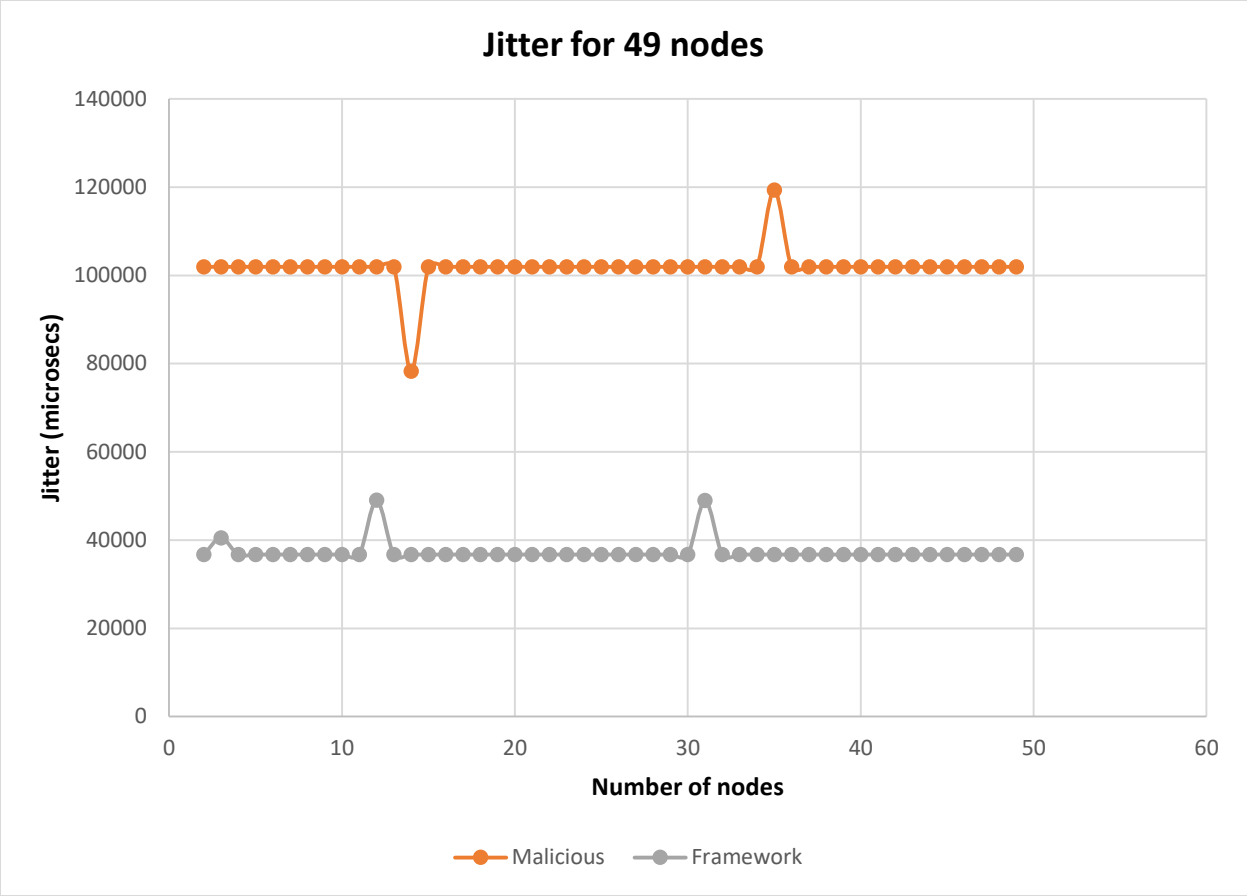

Figure 5.8: Jitter for 49 nodes

5.2.3 Simulation 3: Performance evaluation under high-density nodes scenario (100 nodes)

The proposed framework still outperforms the malicious node scenarios based on the stated performance metrics. There were high margins of delay and jitter for the malicious scenario relative to the density of the nodes due to a lack of timely access to resources. Figure 5.9 presents the implemented network model of 100 uniformly placed mobile nodes with sinkhole nodes present.

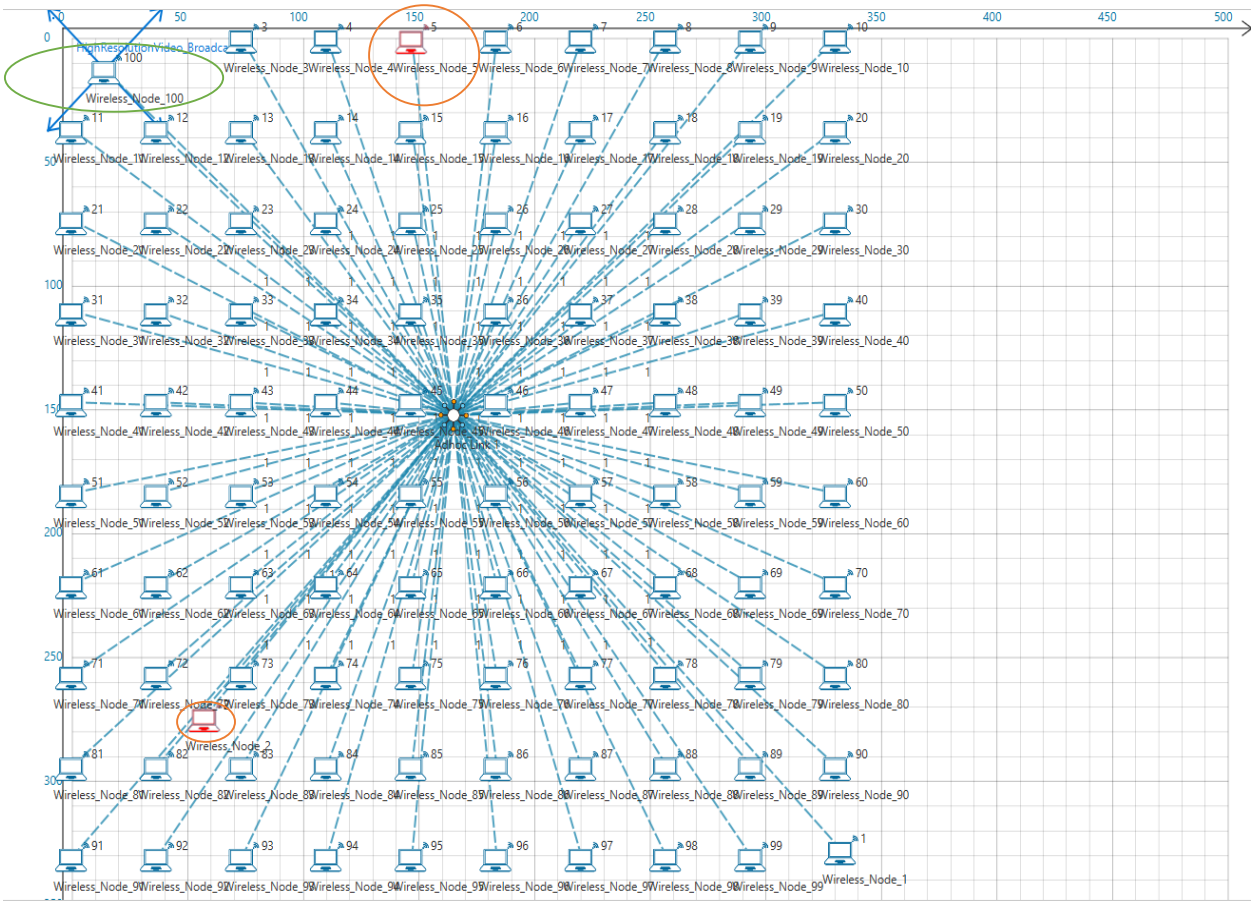

Figure 5. 9: Network model with sinkhole nodes

5.2.3.1 Throughput performance analysis

Figure 5.10 shows the throughput of 100 nodes as the first 5000 milliseconds represents the time at which the application starts. In the malicious scenario, the next 4972 milliseconds showed a positive hike up to 1.78 Megabits/sec in throughput followed by a steady decrease to 0.391 Megabits/sec at 68851.81 milliseconds. The framework scenario showed a positive hike to about 3.68 Megabits/sec in throughput followed by a steady increase up to 4.79 Megabits/sec at 99727.63 milliseconds. This is the highest recorded throughput throughout all the simulations conducted despite the high node density. The high jitter levels contributed to the late arrival of packets and low levels of throughput in the sinkhole scenario. The high node density environment means that many nodes compete for access to resources and the unavailability of queuing mechanisms would mean higher traffic congestion in the network. This does not mean having fewer nodes in the network might be the solution as literature reveals that OLSR performs better in high node

densities. The detection and isolation of the sinkhole attack in the network is the best solution to network performance as done with the framework scenario. The QoS mechanisms implemented optimize the use of the available bandwidth and traffic prioritization was implemented.

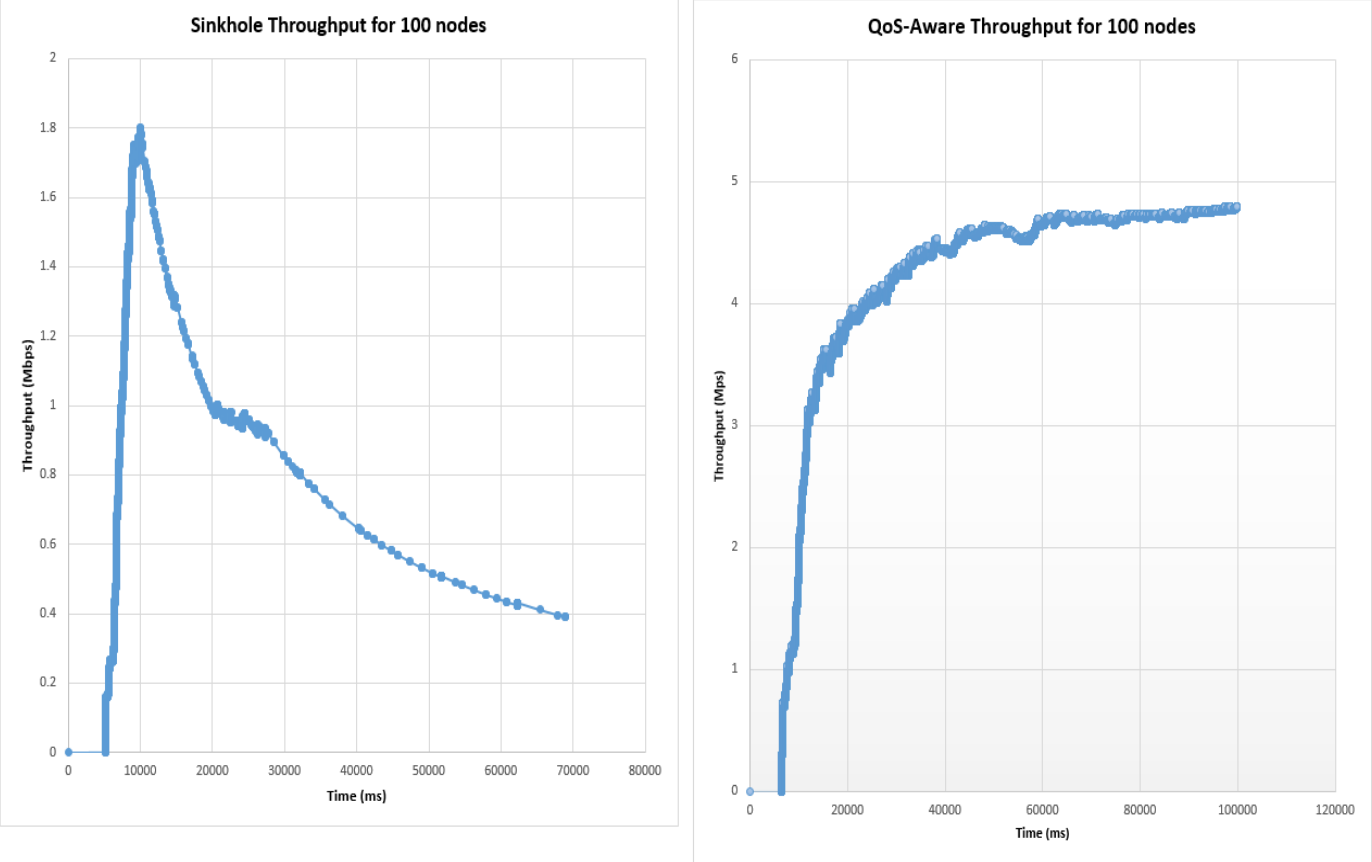

Figure 5.11: Throughput for sinkhole and framework scenarios

5.2.3.2 Delay performance analysis

The results of the scenarios in Figure 5.12 show a high delay margin for the malicious scenario as compared to the framework scenario. In the sinkhole scenario, the highest recorded delay value was 6152112.929 microseconds for node 63 as it was the first computed node like Node 29, 53, and 63. These levels are too high as compared to the framework scenario. The framework scenario at Node 1 had a delay of 581591.68 microseconds. All the other nodes produce a constant delay of 429470.72. This was a significant drop in delay as compared to the other scenario. The effectiveness of the framework is clearly shown in the result presented in Figure 5.12 where the

delay was reduced to about four times as compared to that of the sinkhole scenario. The framework aimed to maximize QoS offering and this was evident as it outperformed the normal scenario as well. It is important to note that the delay caused by the malicious nodes was corrected significantly by the framework running the OLSR protocol.

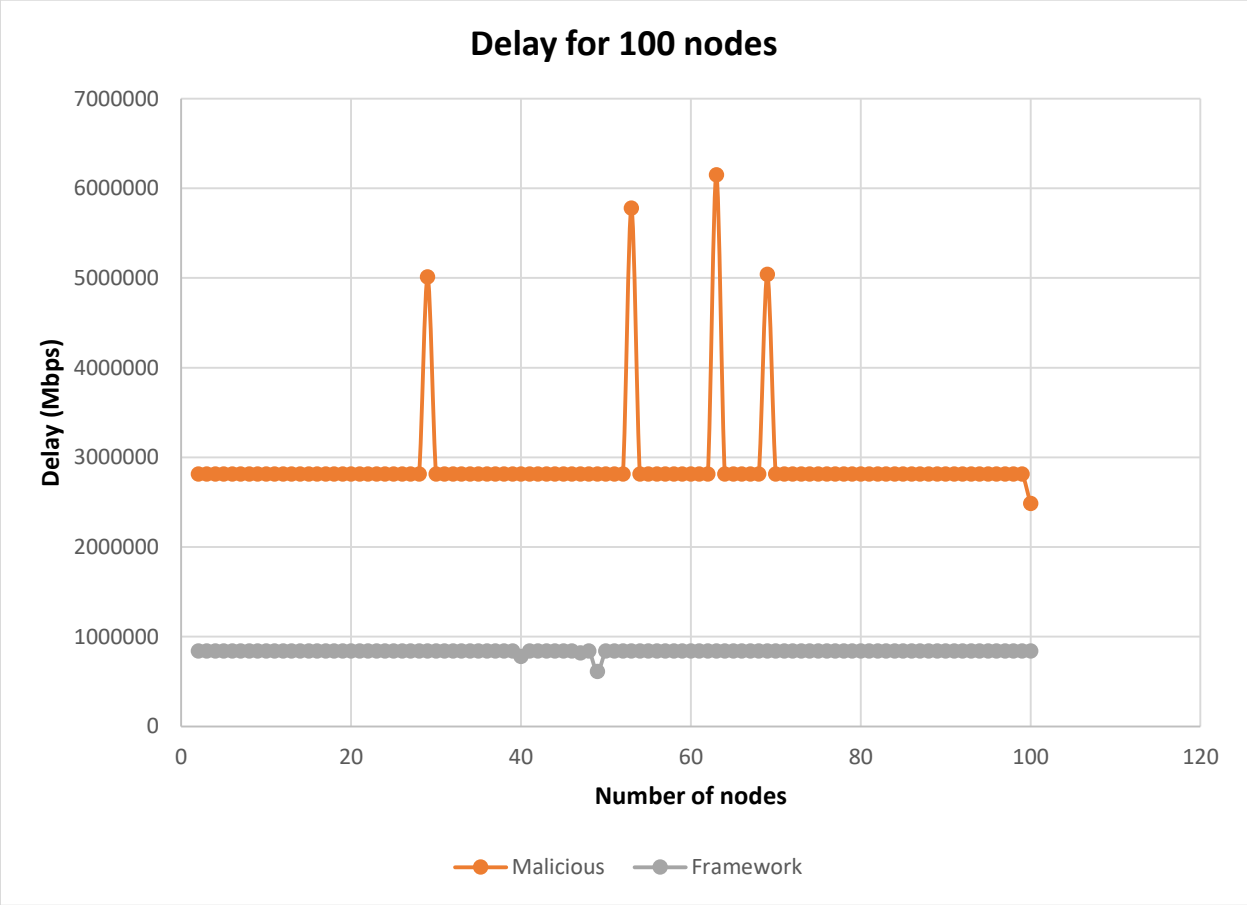

Figure 5.12: Delay for 100 nodes

5.2.3.3 Discussion Jitter

The results in Figure 5.13 show a high jitter margin for the malicious scenario as compared to the other scenario. It was eight times higher than that of the framework. The jitter levels rise from 309406.55 microseconds to a constant 796236.23 microseconds and this result is less desirable from the malicious scenario because it has an even lower throughput to that effect. In the sinkhole attack scenario, the first 28 nodes show a constant level of 796236.267 microseconds. Node 29 is on 309406.55 microseconds. There is a continuous drop of jitter levels from 796236.267 microseconds for 3 other nodes in the network. The drop fluctuates between 297602.65 and

305562.99 microseconds. This is a relatively high level of jitter but it is not surprising as the network is under an active sinkhole attack. The projected result from the malicious scenario was caused by the late access to enough bandwidth as it is shared among the different nodes. The density of the nodes in the network further affected this result but the framework had an even lower margin because there was no malicious behaviour. The sinkhole attack in the other scenario maximizes bandwidth usage unnecessarily in certain nodes and thus starving others.

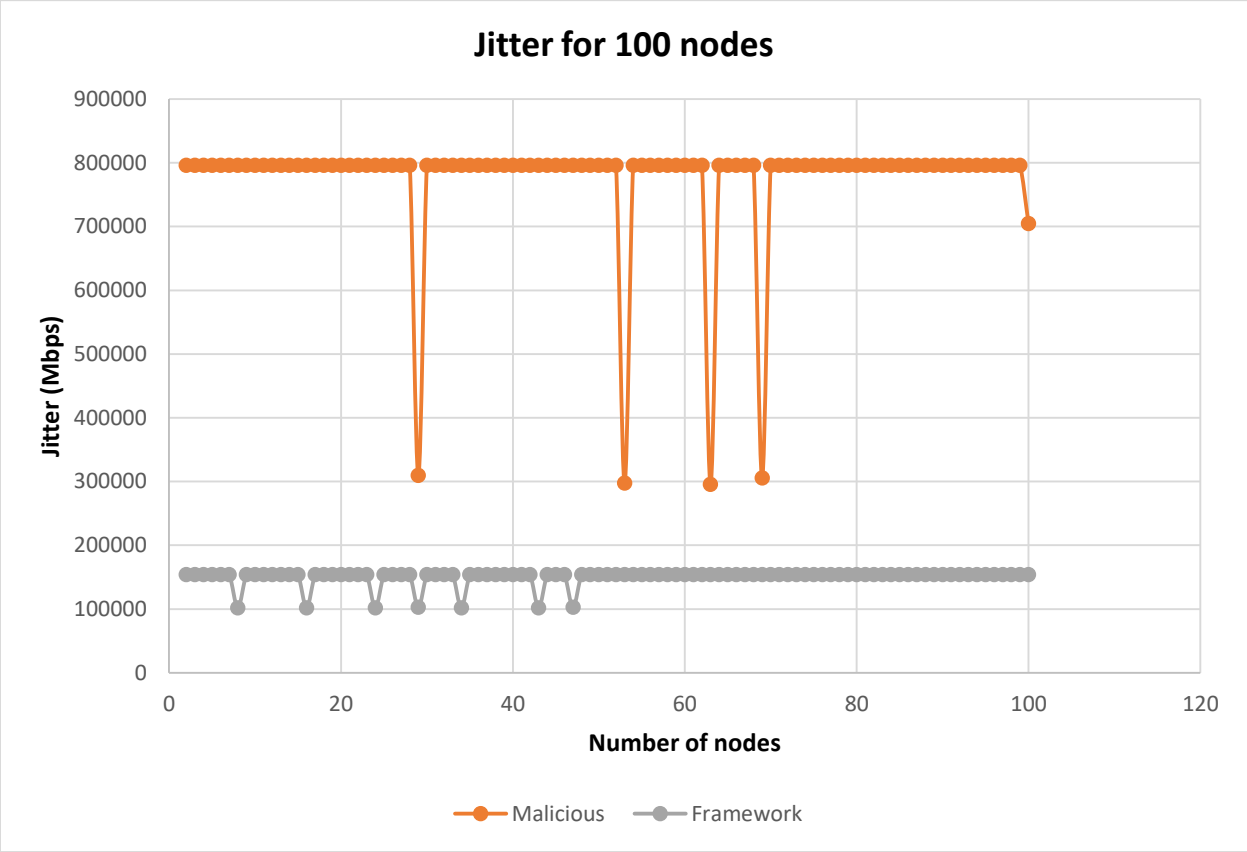

Figure 5.13: Jitter for 100 nodes

5.7 Chapter Summary

The goal of this chapter was to critically analyze and discuss the obtained findings with the help of graphs. The results obtained were evident in our contributions to the development of protocol attainment of quality of service and security. In normal cases, frameworks improve performance in areas where attacks were imposed but this framework goes beyond that corrective measure by providing excellent routing practices and ultimately resulting in an efficient network as seen in the results presented. The results of mapping quality of service whilst being security-aware or rather

visa-versa. Routing and security are MANETs biggest challenges and this work addresses them at the same time without overlooking the significance of the other. There is still a lot to be improved on in terms of routing and security particularly towards ensuring minimal network detriments but this work has provided an effective solution to addressing sinkhole attacks whilst using OLSR protocol. The three QoS metrics used provided enough results to validate some parts of the literature and to bring new findings in terms of video streaming applications and quality of service. A sinkhole attack was introduced to a network running OLSR routing protocol which on its own was an important contribution to protocol development. The implementation of the proposed framework was carried out through a cross-layered approach whereby all layers of the TCP/IP model were put into consideration and it outperformed all the other scenarios. The next chapter presents conclusions based on the results and observations in this chapter.

CHAPTER 6

SUMMARY, CONCLUSION, RECOMMENDATIONS AND FUTURE WORK

6.1 Chapter Overview

This chapter takes a brief outlook towards the entire project in terms of the attainment of goals, aims and objectives poised. Conclusions were drawn based on the results obtained, the accuracy and effectiveness of our proposed solution to network security and QoS. Furthermore, suggestions were made on what could be done in the future to achieve better-scaled results. The suggestions for future work are for researchers to study and make possible improvements in the work.

6.2 Research Summary

Network performance is key to optimizing different applications in an organization. Using QoS means that now organizations achieve an expected service quality for their applications and thus delivering to the expectations of their users. QoS is particularly important to guarantee high-performance critical applications (video streaming) which may require more bandwidth and also prevent the delay of these high-priority applications so that they perform at an acceptable level. It also happens that the performance of the network is hindered by the presence of an attack or unauthorized access of an alien device. This creates a great vulnerability to the integrity of the network and its applications. Literature provides mechanisms that look into performance and security as separate goals in a network. We presented a node-based QoS-aware security framework to bridge the gap and the evaluation was conducted using NetSim v12.1.

The main goal of this research work was to formulate, design and implement an efficient node-based QoS-aware security framework for sinkhole attacks that would address security and QoS in MANETs. This goal was achieved by running a video conferencing application (high-resolution video) in our designed network scenarios to generate traffic in our network. The concentration of the framework was centred around OLSR, a MANET routing protocol. In our simulations, we showed malicious nodes present in the network and how these malicious nodes can harm the network and also demonstrated how the proposed counter-mechanisms from our framework can

increase the network's performance. We also showcased how different node density levels affect the performance of the network in the presence of a sinkhole attack and when the sinkhole attack isn't. The results obtained proved that QoS and security can be equally achieved in a system without compromising performance and the scenarios produced a set of unique results that were brought about that confirm this.

6.3 Conclusion

It can be concluded that our framework scenarios outperformed network scenarios with a sinkhole attack in terms of throughput, jitter and delay. Notably, the delay and jitter levels created by the sinkhole attack are evident in the results provided. This goes out to show just how critical it was for the formulation of a performance-based security solution.

For 16 node scenarios, the throughput levels of the sinkhole scenario were slightly higher than that of the framework. The delay and jitter, however, showed different results altogether. This result was expected for the sinkhole scenario because of the attack present in the network. The framework proved to be effective in low-density node scenarios and the comparison of the two scenarios makes it easier to come up with this conclusion.

For 49 node scenarios, the performance in terms of throughput was in contrast to that of the previous scenarios. The sinkhole scenario significantly dropped performance after about 13000 milliseconds of simulation time while the framework scenario's performance surged for a maximum throughput of more than 2.3 Mbps. The contrast in the outlook of the delay among the sinkhole and framework scenarios again is too significant. The framework was enhanced by the optimization of OLSR and the cross-layer interaction between the network, physical and MAC layers thereby making it delay-sensitive and efficient in its performance. A similar observation was observed even for jitter. The framework's optimization process involved the eradication of sinkhole nodes in the network and also enhancing the OLSR routing protocol's key functions for better performance.

For 100 node scenarios, the throughput for the sinkhole scenario was the worst in this case as it significantly dropped just after 10000 milliseconds. The framework scenario, however, surged higher throughout the simulation. The most critical aspect of the framework is the detection and removal of the sinkhole attack. The QoS mechanisms implemented to optimize the use of the

available bandwidth and traffic prioritization was implemented. The routing practices offered by the framework ensured that it performed four times better than the sinkhole scenario in terms of delay. A similar observation was made for jitter. The sinkhole nodes in the other scenario selfishly maximize bandwidth usage unnecessarily in certain nodes and thus starving others.

The proposed framework was effective in all scenarios. Notably, the performance in mid and high node densities further confirms parts of the literature in terms of OLSR's strengthened performance in terms of throughput maximization. The results also displayed the severity of how sinkhole attacks affect network performance. The performance of our proposed framework was also enhanced by its design. A cross-layer design helps in the sharing of vital information relating to the use of network resources and also the attainment of high adaptivity. This results in the global optimization of the network by leveraging the interactions and dependencies between different layers to achieve better performance. The decision to modify OLSR was influenced by that routing is key to any network's performance. An IDS was key to removing the sinkhole attack in the network so that it does not affect the network's performance. Based on the results obtained, MANETs remain a field of great interest and there is still so much to be done in terms of QoS and security. A limitation to the proposed framework operates on the aspect of an existing sinkhole attack in the network but not a preventative aspect. The reality is that even the most secure systems in the world do succumb to threats at some point in time.No system is fully immune to security attacks and threats.

In conclusion, the study met all its intended objectives and the research questions were answered. The literature reviewed showed the extent to which network developers and researchers have addressed security issues in MANETs with consideration for QoS in networks. This also further revealed the gap between QoS and security as most studies concentrated on these two aspects individually hence the motivation to continue with the formulation of our proposed framework. The simulation of a sinkhole scenario helped us in the investigation of the extent to which malicious nodes interfere with the stability and performance of a network. The results, beyond any reasonable doubt, displayed the effect of the sinkhole attack on the network's performance. It was poor and unacceptable by any standard. The proposed framework was a measure taken to prevent security breaches in the network and guarantee optimal performance. The taken measures were directly linked with the routing mechanisms to ensure that they are QoS-aware. Furthermore, AES

encryption was used to protect the integrity of the packets transmitted. The last objective related to the formulation, simulation, and evaluation of a node-based security framework for sinkhole attacks such that the security and QoS of MANET can be measured. The formulation and implementation of the framework were as shown in Chapter 4 and its evaluation in Chapter 5. The results showed enhanced performance. This achievement made it possible for us to draw the following contributions at large:

a) **Monitoring network QoS and security:** the framework could collect data about routes and application traffic and be able to measure where it was deemed necessary. This made it possible for the allocation of resources at specific times.
b) **Network Traffic Optimization:** the framework addressed many existing irregularities in routing through the optimization of the OLSR protocol and thus bringing about traffic prioritization, management, differentiation and resource optimization that is rooted within the prescripts of QoS. The cross-layer design of the layers made this an achievable goal.
c) **Sinkhole Attack Prevention:** the presence of a sinkhole attack had devastating effects but the introduction of the framework dealt with this irregularity and removed the threat from the network to focus on the optimization of all resources to routing.
d) **Network performance:** based on the generated results from various simulations, the desired network performance was achieved and the system performed as envisaged. There was a reduction in delay levels and thus making the framework delay-sensitive. There was a growth in throughput levels meaning that packets were delivered more efficiently. The removal of a sinkhole attack in the network also a reduction in jitter levels. The integration of security and QoS policies gave a far improved performance.
e) **Custom IDS:** the introduction of a problem-specific IDS opened room on how other known attacks can be addressed instead of using traditional approaches that have often worked as standalone architectures.

QoS and security will continue to bing a subject of discussion for many generations to come. The two technology camps must be integrated, not just passively, to stave off network threats that degrade the performance. Security attacks make application requests until the network resources are overloaded as seen with our results. The employed IDS monitors the network for malicious traffic patterns in the packet streams. QoS manages the performance of applications competing for

network resources like bandwidth on the converged link. Linking the two technologies in a cross-layer framework was done so that they can communicate with each other in the coordination of traffic.

6.4 Recommendations and Future work

This work mainly concentrated on the formulation, design and implementation of an efficient node-based QoS-aware security framework for sinkhole attacks that will address security and QoS in MANETs. Apart from integrating two technologies and incorporating the limitations of this study identified in Chapter 1 under section 1.8, more studies can be carried out to further optimize performance by using enhanced versions of QoS classes like ertPS and UGS. As a part of our contribution to knowledge, we discovered that the two QoS classes can be optimized for modern-day networks. Along that line of thought, other applications like voice and non-streaming applications can be evaluated with the framework and perhaps having to readjust certain functionalities like transport protocol to TCP and admission control mechanisms.

In terms of using other means to remove attacks in the network, it would be interesting to use a different approach to it but also one that is lightweight and does not put a strain on the resources of the network. The nature of MANETs would be such an important consideration in the design of that architecture. What would also be interesting is the direct evaluation of the impact that the other components of the framework bring to the fold. This would mean directly looking at factors like admission control, transport classifier, MAC differentiator and resource reservation.

Another possible opportunity that could be pursued in terms of research and development is the adoption of the practices in the formulation of this framework to create a hybrid model that would put an end to other known security attacks. This could bring about contrast ideas on exactly how the attack would be staved off from participating in the routing process.

Energy-saving and power are common concerns for MANETs based on the fact that the general operation of devices in the field is through the strict reliance on battery power. Future research would also need to look into the balance in topological structure, speed of nodes and battery power supply in each node whilst using the framework. Modelling such relationships would make the next steps to this work more interesting. The eco-friendliness of a device is also important and such investigations are possible future paths for research.

Lastly, we have a vision of modelling a resource-based QoS/Security such that performance and safety are guaranteed. The eco-friendliness of devices is such an important aspect in the design board of this vision as the world is moving forward to devices that emit less and do not cause harm to the balance of nature.

6.5 Closing Remarks

QoS manages and controls access to network resources by setting priorities for specific types of data. This is done to manage data traffic and thus reduce packet loss, jitter, latency, delay and increase throughput. Poor QoS contributes to compromised data integrity and security in an organization's network. When the QoS is affected, then this means that the quality of work and the Quality of Experience (QoE) suffers too. Its link to security is of paramount importance for an efficient running network. Therefore, the framework proposed here in this work can be used by a service provider to manage and monitor the network and thus be able to address security threats during video streaming in a MANET.

REFERENCES

[1] S. Semplay, R. Sobti, and V. Mangat, "Review: Trust management in MANETs," *International Journal of Applied Engineering Research,* vol. 7, p. 2012.

[2] D. S. Aarti, "Tyagi, “Study Of Manet: Characteristics, challenges, application and security attacks”," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, pp. 252-257, 2013.

[3] A. Rajaram and D. S. Palaniswami, "A trust-based cross-layer security protocol for mobile Ad-hoc networks," *arXiv preprint arXiv:0911.0503,* 2009.

[4] P. Goyal, V. Parmar, and R. Rishi, "Manet: vulnerabilities, challenges, attacks, application," *IJCEM International Journal of Computational Engineering & Management,* vol. 11, pp. 32-37, 2011.

[5] M. S. I. M. A. Riaz and M. Tariqu, "Performance analysis of the Routing protocols for video streaming over mobile Ad-hoc Networks," *International Journal of Computer Networks & Communications,* vol. 4, pp. 133-150, 2012.

[6] G. D. Delgado, V. C. Frías, and M. A. Igartua, "Video-streaming transmission with qos over cross-layered Ad-hoc networks," in *2006 International Conference on Software in Telecommunications and Computer Networks*, 2006, pp. 102-106.

[7] M. Lindeberg, S. Kristiansen, T. Plagemann, and V. Goebel, "Challenges and techniques for video streaming over mobile Ad-hoc networks," *Multimedia Systems,* vol. 17, pp. 51-82, 2011.

[8] Y. Singh and M. V. Siwach, "Quality of Service in MANET," *Int. J. Innov. Eng. Technol,* 2012.

[9] N. Sharma, S. Rana, and R. Sharma, "Provisioning of Quality of Service in MANETs performance analysis & comparison (AODV and DSR)," in *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, 2010, pp. V7-243-V7-248.

[10] N. Kaur and M. Joshi, "Implementing MANET Security using CBDS for Combating Sleep Deprivation & DOS Attack," *Open Science Framework. June,* vol. 26, 2017.

[11] M. S. Khan, M. I. Khan, O. Khalid, M. Azim, and N. Javaid, "MATF: a multi-attribute trust framework for MANETs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, p. 197, 2016.

[12] J. Sunderland, "Research questions in linguistics," *Research methods in linguistics*, vol. 1, pp. 9-28, 2010.

[13] S. Ince, C. Hoadley, and P. A. Kirschner, "Research workflow skills for education doctoral students and postdocs: A qualitative study," *The Journal of Academic Librarianship*, vol. 46, p. 102172, 2020.

[14] K. N. Qureshi and A. H. Abdullah, "Study of Efficient Topology Based Routing Protocols for Vehicular Ad-Hoc Network Technology," *World Applied Sciences Journal*, vol. 23, pp. 656-663, 2013.

[15] S. Hamma, E. Cizeron, H. Issaka, and J.-P. Guédon, "Performance evaluation of reactive and proactive routing protocol in IEEE 802.11 Ad-hoc network," in *ITCom 06-next generation and sensor networks*, 2006, p. 638709.

[16] S. Xu, P. Guo, B. Xu, and H. Zhou, "QoS evaluation of VANET routing protocols," *Journal of Networks*, vol. 8, pp. 132-139, 2013.

[17] S. Singh, P. Kumari, and S. Agrawal, "Comparative Analysis of Various Routing Protocols in VANET," in *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*, 2015, pp. 315-319.

[18] N. S. M. Usop, A. Abdullah, and A. F. A. Abidin, "Performance evaluation of AODV, DSDV & DSR routing protocol in grid environment," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, pp. 261-268, 2009.

[19] N. A. M. Saudi, M. A. Arshad, A. G. Buja, A. F. A. Fadzil, and R. M. Saidi, "Mobile ad-hoc network (MANET) routing protocols: A performance assessment," in *Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017)*, 2019, pp. 53-59.

[20] B. D. Shivahare, C. Wahi, and S. Shivhare, "Comparison of Proactive And Reactive Routing Protocols In Mobile Ad-hoc Network Using Routing Protocol Property," *International*

Journal of Emerging Technology and Advanced Engineering. Website: www. ijetae. com (ISSN 2250-2459, Volume 2, Issue 3, 2012.

[21] V. Saravanan and D. Vijayakumar, "Performance Of Reactive And Proactive MANET Routing Protocols With Trajectories," in *International Journal of Engineering Research and Technology*, 2012.

[22] T. Phakathi, F. Lugayizi, B. Esiefarhenrhe, and B. Isong, "An Application-centric Trust Management Framework for Mobile Ad-hoc Networks," *International Journal of Digital Information and Wireless Communications*, vol. 9, pp. 143-157, 2019.

[23] H. Singh and M. Singh, "Effect of Black Hole Attack on AODV, OLSR and ZRP Protocol in MANETs," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 2, 2013.

[24] P. Palta and S. Goyal, "Comparison of OLSR and TORA routing protocols using OPNET Modeler," in *International Journal of Engineering Research and technology*, 2012.

[25] G. Rigazzi, V. Palma, G. Tarnea, F. Santucci, and A. Neri, "A cross-layer routing protocol with Quality of service support for flexible deployment in MANETs," in *2012 IEEE Globecom Workshops*, 2012, pp. 508-512.

[26] L. Li and L. Lamont, "A lightweight service discovery mechanism for mobile Ad-hoc pervasive environment using cross-layer design," in *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, 2005, pp. 55-59.

[27] J. Toutouh, J. García-Nieto, and E. Alba, "Intelligent OLSR routing protocol optimization for VANETs," *IEEE transactions on vehicular technology*, vol. 61, pp. 1884-1894, 2012.

[28] R. R. Chandan, B. S. Kushwaha, and P. K. Mishra, "Performance Evaluation of AODV, DSDV, OLSR Routing Protocols using NS-3 Simulator," *International Journal of Computer Network & Information Security*, vol. 10, 2018.

[29] R. Mehra, R. S. Bali, and P. Kaur, "Efficient clustering based OLSR routing protocol for VANET," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, 2016, pp. 1-7.

[30] E. Winjum, A. M. Hegland, P. Spilling, and O. Kure, "A performance evaluation of security schemes proposed for the OLSR protocol," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*, 2005, pp. 2307-2313.

[31] R. Jain and I. Kashyap, "An QoS aware link defined OLSR (LD-OLSR) routing protocol for MANETs," *Wireless Personal Communications*, vol. 108, pp. 1745-1758, 2019.

[32] H. Kanagasundaram and K. Ayyaswamy, "Multi-objective ALO based energy efficient and secure routing OLSR protocol in MANET," *International Journal of Intelligent Engineering and Systems*, vol. 12, pp. 74-83, 2019.

[33] N. Kumari, S. K. Gupta, R. Choudhary, and S. L. Agrwal, "New performance analysis of AODV, DSDV and OLSR routing protocol for MANET," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 33-35.

[34] S. Y. Dong, "Optimization of OLSR routing protocol in UAV Ad-hoc network," in *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2016, pp. 90-94.

[35] R. Gasmi, M. Aliouat, and H. Seba, "A Stable Link Based Zone Routing Protocol (SL-ZRP) for Internet of Vehicles Environment," *Wireless Personal Communications*, pp. 1-16, 2020.

[36] M. S. Pathan, N. Zhu, J. He, Z. A. Zardari, M. Q. Memon, and M. I. Hussain, "An efficient trust-based scheme for secure and quality of service routing in MANETs," *Future Internet*, vol. 10, p. 16, 2018.

[37] R. Patel and P. Kamboj, "A survey on contemporary MANET security: approaches for securing the MANET," *Int J Eng Technol, February* 2017.

[38] M. K. Nazir, R. U. Rehman, and A. Nazir, "A novel review on security and routing protocols in MANET," *Communications and Network*, vol. 8, pp. 205-218, 2016.

[39] M. Vijayalakshmi and D. S. Rao, "Energy-Aware Multicast Clustering (EAMC) with increased Quality of Service (QoS) in MANETs," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2016, pp. 793-798.

[40] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-based quality of service framework for mobile Ad-hoc networks," *Journal of Parallel and Distributed Computing,* vol. 60, pp. 374-406, 2000.

[41] T. Phakathi, F. Lugayizi, B. Esiefarhenrhe, and B. Isong, "On Trust Management Framework in Video Streaming Applications Over Mobile Ad-hoc Networks," in *The Third International Conference on Electronics and Software Science (ICESS2017)*, 2017, p. 97.

[42] M. Sedrati and A. Benyahia, "Multipath routing to improve quality of service for video streaming over mobile Ad-hoc networks," *Wireless Personal Communications,* vol. 99, pp. 999-1013, 2018.

[43] D. Spanos, "Intrusion Detection Systems for Mobile Ad-hoc Networks," 2018.

[44] D. Chahal and L. Kharb, "Security Concepts Underlying MANET," *International Journal of Emerging Technologies in Engineering Research (IJETER),* vol. 5, 2017.

[45] S. Balasubramani, S. Rani, and K. S. Rajeswari, "Review on Security Attacks and Mechanism in VANET and MANET," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, ed: Springer, 2016, pp. 655-666.

[46] B. K. Tripathy, S. K. Jena, V. Reddy, S. Das, and S. K. Panda, "A novel communication framework between MANET and WSN in IoT based smart environment," *International Journal of Information Technology,* pp. 1-11, 2020.

[47] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," *IEEE Sensors Journal,* vol. 13, pp. 3558-3567, 2013.

[48] P. Goyal, S. Batra, and A. Singh, "A literature review of security attack in mobile ad-hoc networks," *International Journal of Computer Applications,* vol. 9, pp. 11-15, 2010.

[49] P. Rajakumar, V. T. Prasanna, and A. Pitchaikkannu, "Security attacks and detection schemes in MANET," in *Electronics and Communication Systems (ICECS), 2014 International Conference on*, 2014, pp. 1-6.

[50] S. Jain and N. Hemrajani, "Detection and mitigation techniques of black hole attack in MANET: An Overview," *International Journal of Science and Research (IJSR), India Online ISSN*, pp. 2319-7064, 2013.

[51] T. Jamal and S. A. Butt, "Malicious node analysis in MANETS," *International Journal of Information Technology*, vol. 11, pp. 859-867, 2019.

[52] E. V. Shcherba, G. A. Litvinov, and M. V. Shcherba, "A Novel Reputation Model for Trusted Path Selection in the OLSR Routing Protocol," in *2019 International Conference on Information Science and Communications Technologies (ICISCT)*, 2019, pp. 1-5.

[53] P. K. Sahu and B. K. Pattanayak, "Quality of Service based Multicasting Routing Protocols for MANETs: A Survey," *Aptikom Journal on Computer Science and Information Technologies*, vol. 2, pp. 31-46, 2017.

[54] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, p. 20, 2019.

[55] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: A cross-domain overview," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 3639-3681, 2019.

[56] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *Journal of Network and Computer Applications*, vol. 62, pp. 9-17, 2016.

[57] T. Sanguankotchakorn, S. K. Wijayasekara, and N. Sugino, "A Cross-layer design approach in OLSR MANET using BER and weighted Connectivity Index," in *2013 19th IEEE International Conference on Networks (ICON)*, 2013, pp. 1-6.

[58] B. Wang, X. Chen, and W. Chang, "A light-weight trust-based QoS routing algorithm for Ad-hoc networks," *Pervasive and Mobile Computing*, vol. 13, pp. 164-180, 2014.

[59] R. Singh and N. Singh, "Performance Assessment of DSDV and AODV Routing Protocols in Mobile Adhoc Networks with Focus on Node Density and Routing Overhead," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2020, pp. 298-303.

[60] S. Kaple and S. Somkuwar, "Detailed study of Routing Protocols in MANETs."

[61] J. Manoranjini, A. Chandrasekar, and S. Jothi, "Improved QoS and avoidance of black hole attacks in MANET using trust detection framework," *Automatika,* vol. 60, pp. 274-284, 2019.

[62] H. Bali and N. Hemrajani, "Attack analysis and designing of quality of service framework for optimized link state routing protocol in MANET," *International Journal of Intelligent Engineering and Systems.(Oct. 2018),* 2018.

[63] S. Sahu and S. Sharma, "Secure and Proficient Cross Layer (SPCL) QoS Framework for Mobile Ad-hoc," *International Journal of Electrical and Computer Engineering,* vol. 9, p. 2603, 2019.

[64] P. Madhavan and P. Malathi, "Intelligent Framework For Qos Optimization In Manet Using Soft Computing Models," *Int J Adv Engg Tech/Vol. VII/Issue II/April-June,* vol. 890, p. 893, 2016.

[65] S. Kalaivanan, "Quality of service (QoS) and priority aware models for energy efficient and demand routing procedure in mobile Ad-hoc networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-8, 2020.

[66] E. E. Zakaria, H. S. Hamza, and I. A. Saroit, "An integrated security framework for access control and address auto-configuration for MANETs," in *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2015, pp. 253-260.

[67] M. H. Eiza, T. Owens, Q. Ni, and Q. Shi, "Situation-aware QoS routing algorithm for vehicular Ad-hoc networks," *IEEE Transactions on vehicular technology,* vol. 64, pp. 5520-5535, 2015.

[68] K. Pushpalatha and M. Karthikeyan, "A generalized framework for disruption tolerant secure opportunistic routing during emergency situations using MANETs," *Cluster Computing,* vol. 22, pp. 9905-9913, 2019.

[69] S. Sharma and R. Agarwal, "Optimizing QoS parameters using computational intelligence in MANETS," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 708-715.

[70] U. Khan, S. Agrawal, and S. Silakari, "Detection of malicious nodes (DMN) in vehicular ad-hoc networks," *Procedia computer science*, vol. 46, pp. 965-972, 2015.

[71] R. Baiad, O. Alhussein, H. Otrok, and S. Muhaidat, "Novel cross-layer detection schemes to detect black hole attack against QoS-OLSR protocol in VANET," *Vehicular Communications*, vol. 5, pp. 9-17, 2016.

[72] V. Sathish and D. THIYAGARASU, "ENHANCED MAC WITH DELAY AWARE MULTIPATH ROUTING (EMAC–DAMR) MECHANISM FOR QUALITY OF SERVICE IN MOBILE AD-HOC NETWORKS," *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 12-21, 2016.

[73] M. Ahmad, Q. Chen, M. Najam-Ul-Islam, M. A. Iqbal, and S. Hussain, "On the secure optimized link state routing (SOLSR) protocol for MANETs," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2017, pp. 1-8.

[74] H. Duran-Limon, M. Siller, M. Hernandez-Ochoa, C. Quevedo, and V. Robles, "A Network QoS Framework for Real-time Event Systems in highly Mobile Ad-hoc Environments," *Journal of applied research and technology*, vol. 12, pp. 343-358, 2014.

[75] R. H. Jhaveri, N. M. Patel, Y. Zhong, and A. K. Sangaiah, "Sensitivity analysis of an attack-pattern discovery-based trusted routing scheme for mobile ad-hoc networks in industrial IoT," *IEEE Access*, vol. 6, pp. 20085-20103, 2018.

[76] O. Abumansoor, "A Design and Evaluation of a Secure Neighbourhood Awareness Framework for Vehicular Ad-Hoc Networks," Université d'Ottawa/University of Ottawa, 2014.

[77] E. Gurumoorthi, A. Ayyasamy, M. Archana, and J. V. Barathy, "Performance enhancement for QoS in VoIP applications over MANET," *J. Adv. Comput. Electron. Eng.*, vol. 2, pp. 47-54, 2017.

[78] S. Maamar and B. Abderezzak, "Predict link failure in AODV protocol to provide quality of service in MANET," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 8, pp. 1-9, 2016.

[79] M. Thaalbi and N. Tabbane, "Improving Vertical Handover over Heterogeneous Technologies Using a Cross Layer Framework," in *2014 IEEE Intl Conf on High Performance*

Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICESS), 2014, pp. 1170-1176.

[80] H. Xia, Z. Li, Y. Zheng, A. Liu, Y.-J. Choi, and H. Sekiya, "A novel light-weight subjective trust inference framework in MANETs," *IEEE Transactions on Sustainable Computing*, 2018.

[81] Y. Sahu, M. Rizvi, and R. Kapoor, "Intruder detection mechanism against DoS attack on OLSR," in *2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS)*, 2016, pp. 99-103.

[82] G. M. Borkar and A. R. Mahajan, "A secure and trust based on-demand multipath routing scheme for self-organized mobile ad-hoc networks," *Wireless Networks*, vol. 23, pp. 2455-2472, November 01 2017.

[83] F. A. Khan, M. Imran, H. Abbas, and M. H. Durad, "A detection and prevention system against collaborative attacks in Mobile Ad-hoc Networks," *Future Generation Computer Systems*, vol. 68, pp. 416-427, 2017.

[84] D. Hurley-Smith, J. Wetherall, and A. Adekunle, "SUPERMAN: Security Using Pre-Existing Routing for Mobile Ad-hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 2927-2940, 2017.

[85] S. Sharma and M. Mahajan, "Security Mechanisms for Mitigating Multiple Black hole Attack In Manets," *IJISE International Journal of Innovative Science, Engineering & Technology*, vol. 2, pp. 582-588, 2015.

[86] L. L. Monica, "Evaluation of Attacks using different Parameters based on their performance," *Evaluation*, pp. 106-110, 2018.

[87] Q. M. Salih, M. A. Rahman, F. Al-Turjman, and Z. R. M. Azmi, "Smart Routing Management Framework Exploiting Dynamic Data Resources of Cross-Layer Design and Machine Learning Approaches for Mobile Cognitive Radio Networks: A Survey," *IEEE Access*, vol. 8, pp. 67835-67867, 2020.

[88] R. Ranjan and S. Varma, "Challenges and implementation on cross layer design for wireless sensor networks," *Wireless personal communications*, vol. 86, pp. 1037-1060, 2016.

[89] R. K. Saini and S. V. Ritika, "A Survey on Cross Layer Design implementation in Wireless Sensor Network," *Int J Adv Inf Sci Technol IJAIST,* vol. 5, pp. 101-107, 2016.

[90] K. Dooley, "Simulation research methods," *Companion to organizations,* pp. 829-848, 2002.

[91] G. S. Fishman, *Discrete-event simulation: modeling, programming, and analysis*: Springer Science & Business Media, 2013.

[92] J. Karnon, J. Stahl, A. Brennan, J. J. Caro, J. Mar, and J. Möller, "Modeling using discrete event simulation: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force–4," *Medical decision making,* vol. 32, pp. 701-711, 2012.

[93] S. Rajasekar, P. Philomonathan, and V. Chinnathambi, "Research methodology. arXiv," *arXiv preprint physics/0601009,* 2006.

[94] S. Almalki, "Integrating Quantitative and Qualitative Data in Mixed Methods Research--Challenges and Benefits," *Journal of education and learning,* vol. 5, pp. 288-296, 2016.

[95] M. Lakshman, L. Sinha, M. Biswas, M. Charles, and N. Arora, "Quantitative vs qualitative research methods," *The indian journal of pediatrics,* vol. 67, pp. 369-377, 2000.

[96] K.-H. Chen and G. F. Murray, "Truths and untruths in village Haiti: an experiment in third world survey research," *Culture, natality and family planning,* pp. 241-262, 1976.

[97] E. Linares-Espinós, V. Hernández, J. Domínguez-Escrig, S. Fernández-Pello, V. Hevia, J. Mayor, *et al.*, "Metodología de una revisión sistemática," *Actas Urológicas Españolas,* vol. 42, pp. 499-506, 2018.

[98] C. R. Kothari, *Research methodology: Methods and techniques*: New Age International, 2004.

[99] A. Law and W. Kelton, "Simulation modeling ands analysis," ed: McGraw-Hill Company, New York, 1982.

[100] R. Mehta and D. K. Lobiyal, "Energy efficient cross-layer design in MANETs," in *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2017, pp. 448-453.

[101] A. Domanski, J. Domanska, M. Pagano, and T. Czachorski, "The Fluid Flow Approximation of the TCP Vegas and Reno Congestion Control Mechanism," in *ISCIS*, 2016, pp. 193-200.

[102] D. Madhuri and P. C. Reddy, "Performance comparison of TCP, UDP and SCTP in a wired network," in *2016 International Conference on Communication and Electronics Systems (ICCES)*, 2016, pp. 1-6.

[103] A. Abdullah, "Advanced encryption standard (aes) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16, 2017.

[104] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security," *Optik*, vol. 127, pp. 2341-2345, 2016.

[105] M. Umaparvathi and D. K. Varughese, "Evaluation of symmetric encryption algorithms for MANETs," in *2010 IEEE International Conference on Computational Intelligence and Computing Research*, 2010, pp. 1-3.

[106] A. Sharma and G. Kushwaha, "Comparative Analysis of Different Encryption Techniques in Mobile Ad-Hoc Networks (MANETs)," *IITM Journal of Management and IT*, vol. 10, pp. 55-64, 2019.

[107] A. Ahmad and S. Ismail, "User Selective Encryption Method for Securing MANETs," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, pp. 3103-3111, 2018.

[108] S. Thota, "cross layer design at network layer: issues and solutions," *cross-layer design at network layer: issues and solutions*, 12/28 2016.

[109] G.-l. Guo, Q. Qian, and R. Zhang, "Different implementations of AES cryptographic algorithm," in *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, 2015, pp. 1848-1853.

[110] P. Mahajan and A. Sachdeva, "A study of encryption algorithms AES, DES and RSA for security," *Global Journal of Computer Science and Technology*, 2013.

[111] P. Appandairaj and K. Kannan, "Software-Defined Multilayered Admission Control for Quality of Service Assurance in Mobile Ad-hoc Networks," *Wireless Communications and Mobile Computing*, vol. 2020, 2020.

[112] D. Moad, S. Djahel, and F. Naït-Abdesselam, "Improving the quality of service routing in OLSR protocol," in *2012 International Conference on Communications and Information Technology (ICCIT)*, 2012, pp. 314-319.

[113] X. Yu, P. Navaratnam, and K. Moessner, "Resource Reservation Schemes for IEEE 802.11-Based Wireless Networks: A Survey," *IEEE Communications Surveys &amp Tutorials*, vol. 15, 11/29 2012.

[114] C. Hou, Z. Xu, W.-K. Jia, J. Cai, and H. Li, "Improving aerial image transmission quality using trajectory-aided OLSR in flying Ad-hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, pp. 1-21, 2020.

[115] S. K. Suman. (2020, 11/12/2020). *File Exchange*. Available: https://www.tetcos.com/file-exchange.html

[116] D. A. Kumar and S. Venugopalan, "Intrusion detection systems: A review," *International Journal of Advanced Research in Computer Science*, vol. 8, 2017.

[117] K. M. Saifuddin, A. J. B. Ali, A. S. Ahmed, S. S. Alam, and A. S. Ahmad, "Watchdog and Pathrater based Intrusion Detection System for MANET," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (IEEE ICT)*, 2018, pp. 168-173.