Detection and Prevention of ARP Spoofing Attacks on Mobile Ad Hoc Networks

A Thesis Presented

By

Hilina Tadele Denbel

to

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Computer Science

July,2019

Acceptance

Detection and Prevention of ARP Spoofing Attacks on Mobile Ad Hoc Networks

By

Hilina Tadele Denbel

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science

Thesis Examination Committee:

_____

Internal Examiner

_____

External Examiner

_____

Dean, Faculty of Informatics

July,2019

Declarations

I, the undersigned, declare that this thesis is my original work has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Hilina Tadele Denbel

Student

_____

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor:

Dr. Asrat Mulatu

Advisor

_____

Addis Ababa

Ethiopia

July, 2019

Acknowledgment

My biggest thanks and exhortation goes to the almighty God for helping me complete this thesis work. My second deepest gratitude goes to my family who have been with me all the way out supporting and encouraging me. Last but not least I would like to thank Dr. Asrat Mulatu for his valuable guidance in the completion of this work.

Hilina Tadele

Table of Contents

Acceptance ..... **i**
Declarations ..... **ii**
Acknowledgment ..... **iii**
List of Acronyms ..... **vi**
List of Figures ..... **viii**
List of Tables ..... **ix**
Abstract ..... **x**
Chapter One **1**
Introduction **1**
1.1 Background **1**
1.2. Statement of the Problem **4**
1.3. Objectives **6**
1.4. Significance of the Study **6**
1.5. Scope **6**
1.6. Organization of the Rest of the Report **7**
Chapter Two **8**
Literature Review and Related Works **8**
2.1. Literature Review **8**
2.1.1. Mobile Ad Hoc Network **8**
2.1.2. Routing in MANETs **10**
2.1.4. ARP Request and Reply Packets **14**
2.1.5. Gratuitous ARP Packet **15**
2.1.6. The DSR protocol **15**
2.1.7. ARP attacks **18**
2.2. Related Works **20**
2.3. Summary **25**
Chapter Three **27**
Methodology **27**
3.1. Steps Taken by Each Node to Process A Gratuitous Packet **29**
3.2. Pseudocode of The Algorithm **31**
Chapter Four **33**
Simulation, Results and Discussions **33**
4.1. Simulation Setup **33**

4.2. Test Scenarios **33**
4.3. Assumptions Taken **44**
4.4. Test Results **45**
4.5. Comparative Analysis of Results and Discussions **50**
Chapter Five **54**
Conclusions and Future Works **54**
5.1. Conclusions **54**
5.2. Future Works **55**
Appendices **56**
A. Appendix A. Codes **56**
B. Appendix B. Test Results **62**
References **85**

List of Acronyms

AD: Attack Diagnosis

ALP: Alignment Prediction

AP: Access Point

A**RP:** Address Resolution Protocol

BGP: Border Gateway Protocol

CA: Certification Authority

CDF: Cumulative Distribution Function

CGSR: Cluster Head Gateway Switch Routing Protocol

CUSUM: Cumulative Sum

DDoS: Distributed Denial of Service

DHCP: Dynamic Host Configuration Protocol

DoS: Denial of Service

DRSS: Differential Received Signal Strength

DSDV: Destination Sequenced Distance Vector

DSR: Dynamic Source Routing

FIFO: First in First Out

FSR: Fish-Eye State Routing

GDoP: Geometric Dilution of Precision

GPS: Global Positioning System

HMAC: Hashed Message Authentication Code

ICMP: Internet Control Message Protocol

IDPF: Inter Domain Packet Filter

IDS: Intrusion Detection System

IP: Internet Protocol

LAN: Local Area Network

LANMAR: Landmark Routing Protocol

MAC: Media Access Control

MANET: Mobile Adhoc Network

MIMO: Multiple Input Multiple Output

MITM: Man in the Middle

MS: Mobile Station

NIIC: Network Interface Card

OLSR: Optimized Link State Routing

OPNET: Network Simulator

PAD: Parallel Attack Diagnosis

RARP: Reverse Address Resolution Protocol

RERR: Route Error

RFC: Remote Function Call

RFF: Radio Frequency Fingerprinting

RREP: Route Reply

RREQ: Route Request

SNAP: Standard Network Access Protocol

TBRPF: Topology Broadcast Reverse Path Forwarding

TTL: Time To Live

List of Figures

Figure 1: Blacklist and Removal of Entry from Tables 31
Figure 2: Testing Scenario 1 Network topology 34
Figure 3: OPNET Screenshot Scenario 1 35
Figure 4: Testing Scenario 2 Network topology 36
Figure 5: OPNET Screenshot Scenario 2 37
Figure 6: Testing Scenario 3 Network topology 38
Figure 7: OPNET Screenshot Scenario 3 39
Figure 8: Testing scenario 4 Network Topology 41
Figure 9: OPNET Screenshot Scenario 4 41
Figure 11: OPNET Screenshot Scenario 5 44
Figure 12: Result of Test Scenario 1 46
Figure 13: Result of Test Scenario 2 47
Figure 14: Result of Test Scenario 3 48
Figure 15: Result of Test Scenario 4 49
Figure 16: Result of Test Scenario 5 50
Figure 17: Node A Packet Exchange 56
Figure 18: Node B Packet Exchange 57
Figure 19: Node C Packet Exchange 57
Figure 20: Node D Packet Exchange 57
Figure 21: Node E Packet Exchange 58
Figure 22: Node F Packet Exchange 58
Figure 23: Node G Packet Exchange 59
Figure 24: Node H Packet Exchange 59
Figure 25: Node A Packet Exchange 60
Figure 26: Node B Packet Exchange 60
Figure 27: Node C Packet Exchange 61
Figure 28: Node D Packet Exchange 61
Figure 29: Node E Packet Exchange 62
Figure 30: Node F Packet Exchange 62
Figure 31: Node G Packet Exchange 63
Figure 32: Node H Packet Exchange 63
Figure 33: Node M Packet Exchange 64
Figure 34: Node A Packet Exchange 64
Figure 35: Node B Packet Exchange 65
Figure 36: Node C Packet Exchange 65
Figure 37: Node D Packet Exchange 66

Figure 38: Node E Packet Exchange 66
Figure 39: Node F Packet Exchange 67
Figure 40: Node G Packet Exchange 67
Figure 41: Node H Packet Exchange 68
Figure 42: Node M Packet Exchange 68
Figure 43: Node A Packet Exchange 69
Figure 44: Node B Packet Exchange 69
Figure 45: Node C Packet Exchange 70
Figure 46: Node D Packet Exchange 70
Figure 47: Node E Packet Exchange 71
Figure 48: Node F Packet Exchange 71
Figure 49: Node G Packet Exchange 72
Figure 50: Node H Packet Exchange 72
Figure 51: Node M Packet Exchange 73
Figure 52: Node A Packet Exchange 73
Figure 53: Node B Packet Exchange 74
Figure 54: Node C Packet Exchange 74
Figure 55: Node D Packet Exchange 75
Figure 56: Node E Packet Exchange 75
Figure 57: Node F Packet Exchange 76
Figure 58: Node G Packet Exchange 76
Figure 59: Node H Packet Exchange 77
Figure 60: Node M Packet Exchange 77

List of Tables

Table 1: Summary of Review of Related Works 27
Table 2: Summary of Comparative Analysis of Results 52

Abstract

A Mobile Ad-Hoc Network (MANET) is a convenient wireless infrastructure which presents many advantages in network settings. However, such networks are vulnerable to face lots of challenges due to the relatively more flexible setup. As opposed to the wired infrastructure, such networks are more susceptible to attacks of various types. This is due to the non-centralized nature of their overall architecture. Spoofing, session hijacking and man-in-the-middle attacks can be taken as an example for such attacks. Each of the-so-far identified attack types have been thoroughly tackled by different researchers in the past decades. The focus of this particular work was on the ARP spoofing attacks and by proposing an algorithm which implements the Dynamic Source Routing (DSR) protocol it was possible to detect and prevent attacks on MANETs. It was intended to help mitigate the damage that malicious nodes cause on the network once having access to it with a falsified identity. The algorithm proposed in this work introduced a barrier in the form of lease files to prevent unnecessary updates to the ARP table of each node. Whenever a malicious node joins the network, the nearby node detected that it was an attacker and got it blacklisted from the network. This implementation played the role of securing every ARP cache entry. The simulation was made using the OPNET network simulator. It is highly expected that the findings of this paper would present a viable way of securing MANETs from spoofing attacks and opens the door for researchers who intend to expand the algorithm as to reach out multiple MANETs in a single scenario.

Keywords: MANETs, Network Security, ARP Spoofing Attacks, OPNET, Network Simulation

Chapter One
Introduction

1.1 Background

Network security is one of the areas in computer technology which is attracting a lot of interest from a mass of researchers on the area. It is the process of taking physical and software preventative measures to protect the underlying networking infrastructure from unauthorized access and misuse, thereby creating a secure platform for network devices and their programs to perform their duties. Ensuring network security may appear to be very simple and the goals to be achieved may seem to be straightforward. But in reality, the mechanisms used to achieve these goals are highly complex and understanding them involves sound reasoning. Both the wired and wireless types of network are prune to a wide range of attacks. An attack is an information security threat that involves an attempt to intercept and alter information without authorized access or permission [12]. There are many different kinds of attacks, which fall into two broad categories; active and passive attacks. Network security in its truest sense thus implies building a shielding mechanism for network devices and programs which can protect them from attacks.

The general network category on which this thesis work was done is the wireless network. Wireless networks can be classified in two types. These are the infrastructure networks and the infrastructure less (ad hoc) networks. This thesis work has aimed at securing one of the infrastructures less networks, the Mobile Ad Hoc network (MANET) from network security attacks. Mobile Ad Hoc Network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless links where no centralized authority is available, and all nodes independently follow the routine of mobility. Due to the absence of the centralized authority an attacker can easily join the network and perform any malicious activity in mind. Beyond this gap to overcome this fault or problem various trust-based security architecture are proposed for MANETs [39].

MANETs have typical features namely unreliability of wireless links between nodes and constantly changing topology. Each device in a MANET is free to move independently in any direction and will therefore change its links to other devices frequently. Each node must forward traffic unrelated to its own use, and therefore be a router. [40] MANET are interconnected system of wireless nodes which communicate over bandwidth constrained wireless links. Each wireless node can function as a sender, a

receiver or a router. When the node is a sender, it can send messages to any specified destination node through some route. As a receiver, it can receive messages from other nodes. When the node functions as a router, it can relay the packet to the destination or next router in the route.

Different layers of the OSI model require different attack mitigation procedures. A security mechanism designed to operate at a higher layer cannot provide protection for data at lower layers, because the lower layers perform functions of which the higher layers are not aware. Hence, it may be necessary to deploy multiple security mechanisms for enhancing the network security throughout the OSI model.

This paper has given emphasis on improving the security on the Data Link Layer of the OSI model. The datalink layer, as it is the second layer of the OSI model, is concerned with the local delivery of frames between nodes. Examples of the Data Link Layer protocols include ARP, PPP, HDLC and more. Each protocol has its own security hole that give an attacker a chance to intrude into the network and compromise its reliability.

Address Resolution Protocol (ARP) is one of the major protocols in the OSI model and the purpose of this protocol is to map an IPv4 address to a physical address. Network Applications at the application layer use IPv4 address to communicate with another device. But at the Data link layer, the addressing is MAC address, and this address is burned into the network card permanently. The purpose of Address Resolution Protocol (ARP) is to find out the MAC address of a device in the Local Area Network (LAN), for the corresponding IPv4 address, which network application is trying to communicate.

The ARP protocol is implemented using MAC-IP mapping tables known as ARP cache's, which is stored at every node on a network.[40] These tables are indexed using the IP address and each entry has a Time to Live (TTL) before it is wiped to save space and processing time. In MANETs there is no such thing like TTL, and nodes can only communicate with the nodes which are within the wireless network signal range. This creates a strong likelihood of the ARP cache be accessed by any node within a predefined wireless range.

There have been some research papers devoted to mitigating ARP attacks in ad hoc networks. In this paper, an approach to solve ARP spoofing within MANETs has been proposed. This approach does not require any modifications to the protocol headers and goes in line with the no central authority feature of MANETs. The algorithm introduced in this paper brings about the concept of using lease files similar to those implemented by routers and access points in networks. These files will be stored at each node

within the MANET and serve as a protection barrier against the ARP cache which contains the necessary MAC address to IP address mappings needed for network communication. The algorithm has been implemented in two phases. Phase one is the initialization phase, which makes use of the ARP protocol's gratuitous ARP request/reply packet which a node can use to inform its neighbors to update their ARP caches with its MAC-IP mapping. Each new node entering the MANET is forced to send out such a packet. The second phase involves each node within the network acting as a guard by dropping any ARP packets that have a MAC-IP mapping that does not match within its lease file.

1.2. Statement of the Problem

It has been decades since the security of wireless networks has drawn the attention of researchers in the communication world. The ad hoc category of the wireless network has been proved to be more susceptible to network attacks than the wired category [40]. Ad hoc networks do not have any central authority which can manage the system. This means nodes by themselves build the network, links are made and broken by the will of the nodes and the wireless signal range capacity. There is no way of checking the integrity of a message, to check whether or not a node is really who it claims to be. This creates a major security hole in the ad hoc network world. Mobile ad hoc networks are the victims of such security hole, the ARP protocol in the 2nd layer of the OSI model drawing an eye for attackers. All operating systems in a MANET keep an ARP cache. If an attacker gets the chance to access the ARP cache of a node, given the nodes can do nothing about it, can easily monitor the communication that node will have in the future. Communication generally is initiated when hosts try to communicate at the application layer and want an IP address to be resolved into a MAC address. Every time a host requests a MAC address in order to send a packet to another host within its wireless range, it checks its ARP cache to see if the IP to MAC address translation already exists. If it does, then a new ARP request is unnecessary. If the translation does not already exist, then the request for network addresses is sent and ARP is performed by resolving the IP address into a MAC address.

ARP broadcasts a request packet to all the machines on the MANET and asks if any of the machines know they are using that particular IP address. When a machine recognizes the IP address as its own, it sends a reply so ARP can update the cache for future references and proceed with the communication. This process of IP-MAC mapping should proceed in a truthful way where nodes make sure to be the one, they claim to be. They should also be able to check whether or not another node is not sending a falsified IP-MAC mapping. Before granting a node to access the ARP cache of another node, it is important to first to check if it is really who it claims to be. This means when a node tries to communicate it broadcasts an ARP request along with its own IP and MAC addresses. Thus, before commencing the communication and address resolution process, these IP and MAC addresses should be checked if they truly belong to the node who claimed to be the owner. Otherwise anyone who can send an ARP request is going be granted the access to the ARP cache of any node. This provides a heartbeat for attackers. Having access to the ARP table of a node means a lot since every communication passes through the ARP resolution process.

During the process of the ARP request and reply, as discussed earlier, every node receives an ARP request. If

a node that has received such an ARP request packet claims to be the one that was on demand for the communication and sends a reply, there will be no way by which the sender can check its reliability. The Man in The Middle attacks can be taken as a good example here. This attack is a type of attack where a malicious node poisons the ARP caches of two nodes. This happens when the attacker had accessed the ARP caches of the two nodes and have changed some IP-MAC mappings. If the MAC address reported in the packet for the given IP has changed, the new value will overwrite the old one in the cache. This is what we call ARP spoofing attack and would mislead the nodes to send all their messages to the attacker thinking they are really sending it to one another. Thus, the attacker will receive all the packets, and be a man in the middle of the communication. Hence it will have a full right to make, monitor, modify and break the communication between the genuine nodes. Many attack types can use this method of spoofing in order to have a grasp of the ARP packets. Session hijacking, ARP flood and Black hole attacks can be mentioned here. All of them start their attacking by first spoofing the IP-MAC mapping of a genuine node/s.

Such attacks can cause a major in MANETs since they play a great role in compromising the confidentiality and integrity of the packets being switched throughout the network. It is an undeniable truth that MANETs are having a huge range of applications in the pervasive computing world. Some of the typical applications include:

Collaborative work: For some business environments, the need for collaborative computing might be more important outside office environments than inside and where people do need to have outside meetings to cooperate and exchange information on a given project [23].

Military battlefield: Ad-Hoc networking allows the military to take advantage of common place network technology to maintain an information network between the soldiers, vehicles, and military information head quarter.

Local level: Ad-Hoc networks can autonomously link an instant and temporary multimedia network using network computers to spread and share information among participants in a conference or so [23].

Business sectors: Ad hoc networks can be used for rescuing and emergency processes for adversity assistance struggles, such as flood, fire or earthquake emergencies. Such emergency saving procedures should take place where damaged and non-existing transmissions structure and quick preparation of a transmission network is required.

These and many more applications of MANETs are all victims of the so far discussed ARP spoofing attacks. And in order to fill up this security hole, a mitigation approach is needed. This approach will be implemented using Digital Source Routing (DSR) protocol and is intended to secure ARP cache entries.

1.3. Objectives

1.3.1 General Objective

The main objective of this research is to detect and prevent ARP spoofing attacks on mobile ad hoc networks.

1.3.2. Specific Objectives

- ✓ Ensure that any attacker cannot intrude into the ARP caches of the nodes.
- ✓ Secure a network when nodes are of moving type.
- ✓ Detect any attackers particularly those using spoofing method.
- ✓ Learn from the noticed scenario and make the nodes to counteract future attack attempts.

1.4. Significance of the Study

The very first benefiters of this ARP spoofing detection and prevention mechanism are the emerging technology sectors that deploy ad hoc networks. Since the flexible nature of ad hoc networks made it a fitting technology in the modern world, clearly any implementation which is deployed based on an infrastructure less architecture benefits from the securing mechanisms. The so far existing scheme over the area of securing mobile ad hoc networks can be refined in such a way that attackers are cut out of a network once detected instead of waiting until a next attack attempt is made.

1.5. Scope

The research involved the introduction an algorithm to come up with a MAC-IP mapping technique to detect fraudulent nodes. It focuses on ARP spoofing attacks with the aim of coming up with a viable algorithm for detecting, recovering from and preventing future attacks. The nodes considered in this research are of moving type which means they have the tendency to join another nearby MANET.

1.6. Organization of the Rest of the Report

The organizational structure of the rest of this report is as follows:

- **Chapter Two** is the Literature Review where it describes the literatures that have been written so far on the area of the detection, prevention and mitigation of ARP spoofing in mobile ad hoc network.
- **Chapter Three** is the Methodology chapter, it will explain the methodology of the algorithm, including what the proposed algorithm does and how it does its job.
- **Chapter Four** is the Result & Discussion chapter, it will present the simulation procedures taken, the different scenarios considered to test the algorithm, the results of the tested scenarios, comparative analysis of results and Discussion.
- **Chapter Five** is the Conclusion & Recommendation chapter, it contains the summary of what has been done and found, conclusion made from the findings and lastly some recommendations for future work.
- At last the References and Appendices are included.

Chapter Two
Literature Review and Related Works

This chapter presents and discusses the basic concepts that serve as a milestone for the better understanding of this thesis work. The first section presents a conceptual framework on the area of Mobile Ad Hoc networks, the ARP protocol, the DSR protocol, the different types of the ARP Spoofing Attacks and more. The second section depicts some of the major contribution on the area of securing mobile ad hoc networks from ARP spoofing attacks.

2.1. Literature Review

2.1.1. Mobile Ad Hoc Network

Wireless networks can be classified in two types. These are, the infrastructure networks and the infrastructure less (ad hoc) networks. Infrastructure networks consist of network devices which are connected with wire and have wired gateways. The second type is the infrastructure less network which is made up of multiple nodes connected by links. such networks are called Ad-hoc which is a Latin word and it means "for this or for this only" [39].

An ad hoc network typically refers to any set of networks where all devices have equal status on a network and are free to associate with any other ad hoc network device in a prespecified link range. Ad hoc networks have been proposed as an appealing communication technology to deal with the unexpected conditions emerged. Therefore, this kind of network does not rely on any fixed infrastructure and has the properties of self-organizing and self-managing. The three common sub types of these networks are Mobile Ad hoc Networks (MANETs), Wireless Sensor Networks and Wireless Mesh Networks. The main focus of this paper will be centered on the mobile ad hoc network. The following paragraphs will briefly discuss the MANETs.

A MANET by its nature is a network that is void of any infrastructure requiring that the hosts participating in the network be responsible for such tasks as route discovery, packet routing and security. Nodes within wireless range of each other can communicate directly through OSI layer 2 protocols such as 802.11 MAC. To communicate with nodes, outside of their wireless range,

they employ one of many ad hoc routing protocols such as Dynamic Source Routing (DSR), to build source routes to destinations built on hops from node to node. This process is known as the route discovery stage of a protocol. Mobile Ad Hoc Networks (MANETs) refer to a class of wireless networks that can be formed dynamically and randomly without the need for infrastructural setups [39]. Such networks can adapt and reconfigure themselves on the fly according to node mobility and changing network topologies. These characters of MANETs makes them desirable for environments which by nature are continuously changing and require a high level of security. The military sector can be a good example.

A MANET has several distinguishing characteristics. First, it operates on bandwidth constrained variable-capacity links. Wireless links have significantly lower capacity than hard-wired links. In contrast to wired networks which are characterized by high bandwidth links, low bit error rates and stable and symmetric links, a MANET has relatively low bandwidth links, high bit error rates, and unstable and asymmetric links. One effect of having a low link capacity is that congestion is typically the norm rather than the exception [40]. The second characteristic of a MANET is that it has a dynamic topology. Nodes are free to move arbitrarily, causing the network topology to change rapidly and unpredictably over time. Alternative paths are automatically found, after which data packets are forwarded across the multi-hop paths of the network. MANETs use various routing mechanisms to accomplish this.

Thirdly, it does not have a centralized infrastructure. It is unlike the traditional mobile wireless networks in which base stations, access points and servers must be deployed before the networks can be used. Intuitively, this means that the MANET is also a self-configuring network in which network activities, including the discovery of the topology and delivery of messages, are executed by the nodes themselves.

Fourthly, a MANET is often bound by energy constrained operations [40]. This is because its nodes are often hand-held battery-powered devices. Since the mobile nodes rely on these exhaustible means for energy, power conservation is important in a MANET system design.

Finally, in MANETs there is limited physical security. Mobile wireless networks are more prone to the physical security threats of eavesdropping, interception, denial-of service and routing attacks as compared to fixed-cable networks [40]. Hence, security techniques must be applied to reduce these threats. Nodes prefer to radiate as little power as necessary and transmit as

infrequently as possible. This will decrease the probability of detection and interception. In addition, the decentralized nature of network control will add robustness against failure as opposed to the centralized networks.

A MANET, with its inherent dynamic and flexible architecture, demonstrates attractive potential for military applications. It can overcome traditional communications limitations through its automatic relaying and self-healing/forming features. In the commercial sector, developments in MANETs are still ongoing. Emerging technologies such as Multiple-Input Multiple Output, or MIMO, and smart antennas can be integrated within a MANET framework for an even more powerful networking experience.

2.1.2. Routing in MANETs

In MANETs the routing process can be employed using three types of routing protocols; Proactive Routing Protocols, Reactive Routing Protocols and Hybrid Routing Protocols. The following paragraphs discuss these protocols briefly.

A. Proactive Routing Protocols

Proactive MANET protocols are table-driven and will actively determine the layout of the network. Through a regular exchange of network topology packets between the nodes of the network, a complete picture of the network is maintained at every single node. There is hence minimal delay in determining the route to be taken. This is especially important for time-critical traffic. However, a drawback to a proactive nature of protocol is that the life span of a link is significantly short. This phenomenon is brought about by the increased mobility of the nodes, which will render the routing information in the table invalid quickly.

When the routing information becomes invalid quickly, there are many short-lived routes that are being determined and not used before they turn void. Hence, another drawback resulting from the increased mobility is the amount of traffic overhead generated when evaluating these unnecessary routes. This is especially aggravated when the network size increases. The fraction of the total control traffic that consists of actual practical data is further decreased. Lastly, if the nodes transmit infrequently, most of the routing information is deemed redundant. The nodes, however, continue to expend energy by continually updating these unused entries in their routing

table [40]. As mentioned, energy conservation is very important in a MANET system design. Hence, this excessive expenditure of energy is not desired. Thus, proactive MANET protocols work best in networks that have low node mobility or where the nodes transmit data frequently. Examples of proactive MANET protocols include: Optimized Link State Routing, or OLSR Topology Broadcast based on Reverse Path Forwarding, or TBRPF Fish-eye State Routing, or FSR Destination-Sequenced Distance Vector, or DSDV Landmark Routing Protocol, or LANMAR Cluster head Gateway Switch Routing Protocol, or CGSR.

B. Reactive Routing Protocols

Reactive MANET protocols only find a route to the destination node when there is a need to send data. The source node will start by transmitting route requests throughout the network. The sender will then wait for the destination node or an intermediate node (that has a route to the destination) to respond with a list of intermediate nodes between the source and destination. This is known as the global flood search [40], which in turn brings about a significant delay before the packet can be transmitted. It also requires the transmission of a significant amount of control traffic [40]. Thus, reactive MANET protocols are most suited for networks with high node mobility or where the nodes transmit data infrequently.

C. Hybrid Routing Protocols

Since proactive and reactive routing protocols each work best in oppositely different scenarios, there is good reason to develop hybrid routing protocols, which use a mix of both proactive and reactive routing protocols. These hybrid protocols can be used to find a balance between the proactive and reactive protocols. The basic idea behind hybrid routing protocols is to use proactive routing mechanisms in some areas of the network at certain times and reactive routing for the rest of the network. The proactive operations are restricted to a small domain in order to reduce the control overheads and delays. The reactive routing protocols are used for locating nodes outside this domain, as this is more bandwidth-efficient in a constantly changing network.

In this protocol, the radius of each node's local routing zone plays an important part in determining the proactive zone. The proactive routing protocol is used to determine the topology within the radius of the node. The reactive routing protocol is then used to locate nodes outside the radius of the node on demand. The adjustment of the zone radius will allow the protocol to

adapt to different MANET environments. A larger radius will favor the proactive routing protocol, optimal for slow-moving nodes or large amounts of traffic [40].

Consequently, a smaller zone radius will favor the reactive protocol, which is optimal for fast-moving nodes or small amounts of traffic. The WARP, on the other hand, constantly updates all the active routes between the nodes in the network. This is done using routing tables and link-update propagations. When there are link breakages, the destination may become unreachable. In this scenario, WARP will use reactive protocols to find alternative routes to break the deadlock.

2.1.3. ARP Protocol

Address Resolution Protocol (ARP) Address Resolution Protocol (ARP) is one of the major protocols in the TCP/IP suit and the purpose of Address Resolution Protocol (ARP) is to map an IPv4 address to the physical address. Network Applications at the application layer use IPv4 address to communicate with another device. But at the Data link layer, the addressing is MAC address, and this address is burned into the network card permanently. The purpose of Address Resolution Protocol (ARP) is to find out the MAC address of a device in your Local Area Network (LAN), for the corresponding IPv4 address, which network application is trying to communicate.

Static mapping on the other hand, means creating a table that associates a logical address with a physical address. This table is stored in each machine on the network. Each machine that knows, for example, the IP address of another machine but not its physical address can look it up in the table. At times machines change their MAC addresses, a static mapping table must be updated periodically. This overhead could affect network performance. In dynamic mapping a different scenario is implemented. Each time a machine knows the logical address of another machine, it can use a protocol to find the physical address. Two protocols have been designed to perform dynamic mapping: Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP). ARP maps a logical address to a physical address; RARP maps a physical address to a logical address. Since the focus of this thesis work is evolved around the ARP protocol, we will limit our discussion about the RARP protocol to this.

In the following sections the steps involved in an ARP process are discussed briefly:

1. The sender knows the IP address of the target.

2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with 0s.

3. The message is passed to the data link layer where it is encapsulated in a frame using the physical address of the sender as the source address and the physical broadcast address as the destination address.

4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes the IP address.

5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast.

6. The sender receives the reply message. It now knows the physical address of the target machine.

7. The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

The tables are generated during route discovery and during an update process which can be initiated by any node using a gratuitous ARP request/reply message. Once the tables have been generated, they are used to facilitate packet routing to the destination specified within the packet header. When a node receives a packet that is not addressed to it, it checks its ARP table for a matching entry based on the IP address. If such an entry exists, the node forwards the packet accordingly. If no such entry exists, the node must solicit the network for the correct MAC-IP mapping. To generate the tables, and to use them to determine a target address, nodes implement a set of ARP messages known as ARP request, ARP reply and gratuitous ARP request/reply.

The Address Resolution Protocol (ARP) was developed to enable communications on an internetwork and is defined by RFC 826. Layer 3 devices need ARP to map IP network addresses to MAC hardware addresses so that IP packets can be sent across networks. Before a device sends a datagram to another device, it looks in its ARP cache to see if there is a MAC address

and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network. Each device compares the IP address to its own. Only the device with the matching IP address replies to the sending device with a packet containing the MAC address for the device [40]. The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data. The figure below illustrates the ARP broadcast and response process.

When the destination device lies on a remote network, one beyond another Layer 3 device, the process is the same except that the sending device sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The Layer 3 device on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet use Subnetwork Access Protocol (SNAP).

2.1.4. ARP Request and Reply Packets

The ARP request packet is sent out by a node that needs to resolve the MAC address of a host it is attempting to communicate with. When a node needs to send data, to a specific target host, it will first check to see if the target IP address is on the same subnet, if so it proceeds to check its ARP cache table for an entry that contains a mapping from the target IP to a MAC address. If a node determines that the IP is not from the same subnet, it will check its ARP cache for the MAC-IP mapping to the subnets gateway (which then performs its own ARP request to forward the packet outside of the subnet). Regardless of which of the previous two steps it performs, the node proceeds depending on whether it found a mapping in its ARP cache table. When a node determines that it does not currently have a MAC-IP mapping within its ARP cache table, ARP requests are generated and broadcast onto a subnet. Within a LAN any host (including routers and access points) that is on the same subnet would receive this broadcast packet. In MANETs, the broadcast is received only by those nodes that are within wireless signal range of the transmitter since there is no TTL field within an ARP header and thus the packet is a single hop packet.

A node that receives an ARP request on route to the target may need to perform its own ARP

request broadcast to determine the next hop node's MAC. This will depend on whether it has had to resolve this address before for previous routing and has an entry within its ARP cache. When a target node receives an ARP request it crafts an ARP reply which is unicast back to the source of the message. The other address fields are taken from the ARP request packet header, source address fields, and set as the destination addresses of the reply. The data is then sent back onto the network either on the same route that the ARP request came on or on another route. At this point the node may again need to resolve the next hop address as the packet is routed hop by hop along the route.

2.1.5. Gratuitous ARP Packet

A gratuitous ARP packet is another form of ARP packet that does not require a reply and is used as part of route maintenance portion of a network. This packet is referred to as a gratuitous ARP request/reply and is broadcast by a node who wishes to inform the other network devices to update their ARP cache tables for its entry. This could be because of a hardware change resulting in a different MAC address or a renewal of IP address. This is also a point of interest when dealing with malicious nodes that wish to perpetrate an attack on the network. If a network doesn't have measures in place on how to handle gratuitous ARP packets an attacker can easily disrupt network flow or poison ARP caches.

2.1.6. The DSR protocol

This section gives an overview of the DSR protocol. Just to start with, the DSR protocol runs in layer 3 of the OSI model. Data at this layer is transmitted in segments. Each segment contains a header identifying the protocol and what options are employed for this data like those headers above however this protocol and others within this layer do not concern themselves with how data gets onto a network [40]. The sole propose of the DSR protocol is to request and provide routing information. To facilitate the route discovery and maintenance processes, the DSR protocol employs two mechanisms. These are Route discovery and Route Maintenance, each of these are briefly discussed in the following paragraphs.

Route Discovery is the mechanism by which a node S wishing to send a packet to a destination node D obtains a source route to D. Route Discovery is used only when S attempts to send a

packet to D and does not already know a route to D [34]. Whenever a packet is being sent out to a certain receiver, the sequence of hops that the packet should follow on its way to the receiver will be put on the header of the packet. In a normal scenario, the sender node will search its route cache if it has previously learnt the routing path to follow to reach to the receiver node. If it gets it, it will go on with the sending process. But if no route is found in its cache, it will automatically initiate the Route Discovery protocol to dynamically find a new route to its destination node. In this case, we call the sending node, the initiator and the receiver node, the target of the Route Discovery. When a node receives a Route Request, if it is the target of the Route Discovery, it returns a Route Reply message to the initiator of the Route Discovery, giving a copy of the accumulated route record from the Route Request; when the initiator receives this Route Reply, it caches this route in its Route Cache for use in sending subsequent packets to this destination. Otherwise, if this node receiving the Route Request has recently seen another Route Request message from this initiator bearing this same request id, or if it finds that its own address is already listed in the route record in the Route Request message, it discards the Request [34]. Otherwise, this node appends its own address to the route record in the Route Request message and propagates it by transmitting it as a local broadcast packet.

When initiating a Route Discovery, the sending node saves a copy of the original packet in a local buffer called the Send Buffer. The Send Buffer contains a copy of each packet that cannot be transmitted by this node because it does not yet have a source route to the packet's destination [34] Each packet in the Send Buffer is stamped with the time that it was placed into the Buffer and is discarded after residing in the Send Buffer for some timeout period; if necessary for preventing the Send Buffer from overflowing, a FIFO or other replacement strategy can also be used to evict packets before they expire. While a packet remains in the Send Buffer, the node should occasionally initiate a new Route Discovery for the packet's destination address.

However, the node must limit the rate at which such new Route Discoveries for the same address are initiated, since it is possible that the destination node is not currently reachable. In particular, due to the limited wireless transmission range and the movement of the nodes in the network, the network may at times become partitioned, meaning that there is currently no sequence of nodes through which a packet could be forwarded to reach the destination. Depending on the movement pattern and the density of nodes in the network, such network partitions may be rare or may be common.

If a new Route Discovery was initiated for each packet sent by a node in such a situation, many unproductive Route Request packets would be propagated throughout the subset of the ad hoc network reachable from this node. In order to reduce the overhead from such Route Discoveries, we use exponential back-off to limit the rate at which new Route Discoveries may be initiated by any node for the same target. If the node attempts to send additional data packets to this same node more frequently than this limit, the subsequent packets should be buffered in the Send Buffer until a Route Reply is received. But the node must not initiate a new Route Discovery until the minimum allowable interval between new Route Discoveries for this target has been reached [34]. This limitation on the maximum rate of Route Discoveries for the same target is similar to the mechanism required by Internet nodes to limit the rate at which ARP Requests are sent for any single target IP address.

Route Maintenance is the mechanism by which node S can detect, while using a source route to D, if the network topology has changed such that it can no longer use its route to D because a link along the route no longer works. When Route Maintenance indicates a source route is broken, node S can attempt to use any other route it happens to know to D or can invoke Route Discovery again to find a new route. Route Maintenance is used only when S is sending packets to D.

One critical feature of these processes; i.e. Route Discovery and Route Maintenance, is that they both only operate on demand. In DSR, there is no need of periodic packets of any kind at any level within the network. For example, DSR does not use any periodic routing advertisement, link status sensing, or neighbor detection packets, and does not rely on these functions from any underlying protocols in the network [42]. The main benefit of this entirely on-demand feature of these processes comes to life when all nodes in the network are stationary and all current routes have already been discovered. At such times the number of overhead packets caused by the DSR scales down to zero. The DSR protocol is configured such that all the routing packet overhead of the system automatically scales down to a certain figure, as nodes begin to move more or as communication patterns change. A node with multiple routes to a destination can attempt another cached route if the one it has been using fails. And this is the main attractive feature of the DSR protocol since the reaction to routing changes is much more rapid. This caching of multiple routes also avoids the overhead of needing to perform a new Route Discovery each time a route

in use breaks. The operation of Route Discovery and Route Maintenance in DSR are designed to allow unidirectional links and asymmetric routes to be easily supported [42].

In wireless networks, it is possible that a link between two nodes may not work equally well in both directions. This might be caused due to differing antenna or propagation patterns or sources of interference. Such unidirectional links if used properly will definitely improve overall performance and network connectivity traits of the system. DSR enhances this feature. DSR also supports internetworking between different types of wireless networks, allowing a source route to be composed of hops over a combination of any types of networks available.

Instances can be mentioned to justify the feature of the DSR protocol supporting internetworking between different types or wireless networks. For example, some nodes in the ad hoc network may have only short-range radios, while other nodes have both short-range and long-range radios; the combination of these nodes together can be considered by DSR as a single ad hoc network [42]. In addition, the routing of DSR has been integrated into standard internet routing, where a gateway node connected to the Internet also participates in the ad hoc network routing protocols; and has been integrated into Mobile IP routing, where such a gateway node also serves the role of a Mobile IP foreign agent.

A characteristic of MANETs is that nodes are capable of moving, leaving or entering the network at any time and this can result in previous routes being broken. DSR implements a mechanism to aid in route maintenance referred to as Route Error (RERR) messages.

A node that receives data from a source as a result of being on the path to a destination will attempt to transmit data to the next hop on a route a number of times. If it is unsuccessful in reaching its target it will send a route error message back to the source IP. Any node receiving this message on route to the source IP remove the route entry for this path from their cache. Once the source node receives this message it can send data to the target along one of the other paths or, if it should not have another route, perform route discovery for this target again.

2.1.7. ARP attacks

This section describes some of the attacks that are perpetrated using the ARP protocols vulnerabilities. All the attacks below can be classified as active or passive and are either

impersonation attacks or route disruption attacks.

A. Man-In-The-Middle (MITM) attack

A man in the middle attack is achieved when an attacker poisons the ARP cache of two devices with the MAC address of their Ethernet NIC (Network Interface Card). Once the ARP cache has been successfully poisoned, each of the victim devices send all their packets to the attacker when communicating to the other device. This puts the attacker in the middle of the communications path between the two victim devices; hence the name Man-In-The-Middle (MITM) attack [34]. It allows an attacker to easily monitor all communication between victim devices. The objective of this MITM attack is to take over a session. The intent is to intercept and view the information being passed between the two victim devices.

B. The ARP Protocol and Cache Poisoning

The Address Resolution Protocol serves the function of determining the mapping between IP addresses and MAC hardware addresses on local networks. A requesting host sends the message and stores the IP-to-MAC mapping for future packets. In order to minimize network traffic, ARP implementations update their cache of ARP to-IP mappings whenever an ARP request or reply is received. If the MAC address reported in the packet for the given IP has changed, the new value will overwrite the old one in the cache. ARP replies are unicast packets directed at one machine and cause only that machine to update its cache.

C. Session Hijacking

Session hijacking occurs when a malicious node takes over a communication session with the intention to steal authentication information that can later be used for false purposes. A node that performs a successful ARP spoof can monitor packets being sent to a victim. By convincing a victim they are talking to a web server when in fact they are talking to the attacker, the attacker can easily steal the session cookies required for authentication to the web site. This can be serious if such a website is a banking site and sensitive financial information becomes unsecure.

D. ARP Flood

ARP flood is another form of DoS attack. In this case the attacker repeatedly sends a victim ARP request packets, each with a different forged IP address causing the victim to constantly must update their ARP cache entry and reply to the request. While the victim is responding to this flood of requests, they are unable to process other messages from legitimate nodes and

eventually drop the packets.

E. Black Hole

Black hole attacks are a type of Denial of Service (DoS) attack. Whereas most DoS attacks focus on tying up a node, so that it cannot communicate with the rest of the network, by sending a flood of messages to it, black hole attacks cause denial of service by dropping packets routed through them. If enough malicious nodes participate in the attack, they can possibly cut off communication to a substantial portion of the network causing serious outages and preventing necessary data from being transmitted. This can be particularly harmful in military and emergency systems where a delay in communication could be fatal.

2.2. Related Works

Amongst all various other types of attacks on mobile ad hoc networks, identity spoofing attacks are easy to launch and can cause a significant damage on the network performance. It is an undeniable fact that it is easy for an attacker to gather useful MAC address information during passive monitoring and then modify its MAC address by simply issuing some low-level commands. Majority of the existing ad hoc systems are prone to ARP spoofing attacks in such a way that the systems are not able to defend themselves in times of attacks. They can only identify the attacker.

The traditional approach is to use authentication application to address spoofing attacks. However, authentication requires additional infrastructural overhead and computational power associated with distributing and maintaining cryptographic keys [1]. The MANETs work without a centralized administration where the nodes communicate with each other based on mutual trust. This characteristic makes MANETs more vulnerable to be exploited by an attacker inside the network. In this section we are going to discuss some related works thoroughly. Securing each node's identity in a wireless network is one of the most challenging problems. Attacking MAC addresses is not a hard job since it can be done using simple tools. Attack detection, count the number of attackers, identify the location of multiple adversaries in the network are challenging task in mobile ad hoc network. Various authors have tried to address such problems by introducing different approaches. To address the problems associated with spoofing attacks, several traditional approaches are used in authentication application. A review in [2] states that

authentication requires additional infrastructure and computational power associated with distributing and maintaining cryptographic keys.

In the early 2000s, Bellardo et al., [8] conducted an experimental analysis for identification of the attacks by using efficacy and potential low-overhead implementation to mitigate the underlying vulnerabilities. The authors suggested some steps to identify the attack and remove those attackers. The communication between client and AP (Access Point) is established by using authentication request and association request. During this communication, some attacker may access IP address. At the time of data sending the reauthentication message may send to the AP, and then AP again sends the same to client. The attacker can be identified and analyzed using potential stopgap countermeasures. The result showed that the mechanism effectively identifies the DoS attacks.

A review by M.Bohge et al., [9] proposed a framework, called TESLA certificate, for the scalability problems in hierarchical ad hoc networks. The proposed framework authenticates incoming nodes, maintains trust relationship during topology changes and provides data origin authentication for Ad Hoc sensor network. The weak nodes are not involved in the creation or validation of public key signatures. The result showed that the security certificate provides more secure, adaptable and scalable for Ad Hoc sensor networks.

P. Kyasanur et al., [10] proposed modified IEEE 802.11 MAC protocol and correction scheme in 2003. The method used distributed contention resolution mechanisms for sharing the wireless channel. Handling of MAC layer's misbehavior is important for ensuring a reasonable throughput for share well behaved nodes. The proposed protocol easily detects the misbehavior nodes. The correction scheme is very effective in restricting the throughput of selfish nodes. The result showed that the scheme accurately predicts misbehavior nodes.

Ping Tao et al., [11] too in 2003 proposed a technique Traditional localization for increasing robustness. Malicious nodes can easily violate the assumptions by modulating their transmission power of each packet. The mechanism helps for locating mobile devices in an indoor environment; even if the nodes are malicious. The techniques sacrifice some amount of accuracy in the ideal case of localizing cooperative nodes whereas it maintains robustness while facing a variety of model errors, including malicious nodes, nodes with different hardware. The result showed that the mechanism provides better localization accuracy of the system.

In 2004, J. Hall et al., [12] demonstrated an approach, which is incorporation of Radio Frequency Fingerprinting (RFF) and a wireless Intrusion Detection System (IDS). RFF uniquely identify a transceiver based on the transient portion of the generated signal. Moreover, the success rate of a wireless IDS is also improved by correlating several observations in time, using Bayesian filters. Simulation results showed that the RFF technique successfully address the intrusion detection problem in wireless network.

In the year of 2005, A. Wool et al., [13] proposed Wired Equivalent Privacy (WEP) encryption technique which provides key management to address host-revocation problem. The IEEE 802.11 Wireless LAN standard has been designed with very limited key management capabilities and it makes quite difficult to fully revoke the access from previously authorized hosts. Using the technique, Access Point periodically generates new keys and transferred the same to the hosts at given authentication time. The fact is that the keys are only valid for one rekey period of making possible host revocation and scalable. A revoked host will not simply receive the newly introduced 879 keys. WEP is quite simple, very efficient, as well as has no additional requirement beyond the AP and hosts.

Wu et al., [14] proposed a Secure and Efficient Key Management (SEKM) framework in 2005 to build a Public Key Infrastructure (PKI) by using a secret sharing approach and an underlying multicast server group. In SEKM, the cluster of server creates a Certification Authority (CA) view and provides certificate update service to all nodes and servers themselves. A ticket-based scheme for efficient certificate service and efficient server cluster updating scheme are used in SEKM. It is noted that the coordination of server within the group is more efficient rather than the entire network during the secret share updating phase. The result showed that the server group provides certificate update service for all nodes including the servers themselves which are more efficient than the other.

Another paper by Yang Xiao et al., [15] proposed two approaches. One aimed at enhancing security and applied Keyed Message Authentication Code and Enhanced Authentication. The other aimed at enhancing the WEP with Private IV and Session Keys (WEP-PIV-SDK) to overcome some known vulnerabilities and thus to provide better data confidentiality and authentication. Key Management is partially solved since the system is not easily compromised despite the secret key remaining unchanged for a long time. Message Tampering is completely

avoided from the use of Keyed Message Authentication mechanism. The result showed that the proposed enhancements provide better data confidentiality with some degree of computing cost.

F. Guo et al., [16] proposed an algorithm called Sequence Number-Based Spoof Detection algorithm in 2006 to detect the spoofed MAC address effectively without any changes in access points. The algorithm was able to detect all existing spoofing attacks without any false positive or negative. The test results showed that the algorithm can tolerate static access points with abnormal sequence number evolution patterns without generating any false positives. If casual attackers don't exploit the false negative, each spoofed frame will be detected.

In 2007, Xiao L et al., [17] proposed a physical layer algorithm for enhancing authentication in a wireless in-built environment. The technique uses channel frequency response measurements and hypothesis testing to determine whether current and prior communication attempts are made by the same user. In this way, legitimate users can be reliably authenticated, and false users can also be reliably detected. The result showed that the efficiency of the approach is in terms of static channel conditions.

In 2007, Qing Li et al., [18] presented a non-cryptographic mechanism for detecting device spoofing on a wireless network. The authors used a complementary method to authentication and can detect device spoofing with little or no dependency on cryptographic keys. The forge-resistant relationships associated with transmitted packets, and forge-resistant consistency checks allow other network entities to detect anomalous activity. The method uses sequence number field and temporary identifier fields for the evaluation according to a one-way function chain.

In 2008, Jie Yang et al., [20] proposed a scheme that helped derive an analytical expression for the localization of errors. The authors came up with RF- based fingerprint matching schemes [20] to quantify the localization accuracy. The derived analytical expression was termed as the Cumulative Distribution Function (CDF). Further the effects of the sampling point's size and the distance between two adjacent sampling points are studied and the mathematical relationship of the localization error to sampling points is derived. Upon having the results at hand, the authors were able to prove that the performance of RF-based fingerprint matching algorithms to quantify the localization has brought a magnificent change on the system.

In 2009, Jeong H. Lee et al., [23] proposed a localization technique using Differential Received Signal Strength (DRSS). The method did not use signal source cooperation for location estimation. In geometric interpretation, both local and global positioning are used to facilitate the understanding of the approach DRSS. Then, a Least-Squares (LS) optimization framework is formulated for DRSS-based location estimation (DRLE). The result showed that the DRLE has practically several advantages over other positioning techniques and the location accuracy of DRLE improves, where most existing positioning techniques may struggle.

In 2009, G. Auenther Lackner et al., [25] proposed fingerprint technique for detecting the MAC address spoofing in wireless network. The authors designed the method as to rely on supervised machine learning algorithms to learn the typical histograms for each chipset. The client was designed as to use the chipset which is the target for the analysis and creates traffic by communicating with another machine. The probe was configured as to capture the traffic in the monitor mode of its network card. The traffic from the client to the server has been generated by using ICMP pings for covering a large range of possible packet sizes. The captured data is used to create histograms for the construction of the training set and the test set. The result showed that the size of the packet does not increase the time delay.

Gayathri Chandrasekaran et al., [26] proposed an architectural detection algorithm to detect and identity spoofing attacks robustly in 2009. Network services that are restricted to legitimate users only are the main interest of attackers. The authors claimed that identifying those legitimate users allows the attacker to avail those restricted network services. The attacker most probably finds it hard to intrude into the transmitting devices and may become ineffective. The algorithm RSSI-based per-packet localizer employs powerful detectors to identify and eliminate the spoofing attack. The authors have come up with an experimental result that showed the proposed algorithm effectively detects identity spoofs with a low false positive rate of 0.5%.

A paper by A. Mitrokotsa et al., [37]is another remarkable effort in the detection of intrusion in mobile ad hoc network. The paper looks at five supervised classification algorithms and how to properly use these algorithms in intrusion detection. One of the main focuses of the paper was to look at how performance is affected by the problem cost matrix (that is uniform versus weighted cost matrix). Additionally, the paper accessed tuning possibilities for the classifiers when unanticipated spawns of attacks occur. K-1-part cross validation was used as opposed to the

traditional k part cross validation with sequential cross validation being compared to the normal randomized cross validation method. In general, it investigated how hyper-parameter tuning affects performance when new unforeseen attacks are found in the test dataset. The method of data collection employed helped to ensure that the research more closely modeled real world attacks since there are some attacks that were not expected to occur in simulations. The results showed that the cost matrix method not only minimized expected cost but also classification error for most classifiers.

A Survey by Bing Wu et al.,[38] sought to offer a solution to flooding attacks that take place in Neural networks. The attacks that resulted by taking advantage of the AODV protocols specification for sending out fake control packets (such as RREQ or RREP packets). This is achieved in malicious nodes by disregarding the RREQ_RATELIMIT parameter defined in the protocol.

A paper written by P.Goyal et al., [39] basically had a look at the problems that current MANET research seeks to address, the aim and goals of such researches; security-wise as well as the possible attacks that can take place on such networks. There is also a look at the routingprotocols and the different scenarios under which reactive (source initiated, or demand driven) or proactive (table-driven) protocols would be needed. Hybrid protocols; featuring a combination of intra-zonal proactive approach and an inter-zonal reactive approach are also mentioned in the paper. The authors have tried to recognize the absence of a centralized management in MANETS, power supply constraints, scalability issues. Problems related to dynamic topology issues were also mentioned. Below is included a summarized table of the most relatively relevant research work amongst the so far discussed papers.

2.3. Summary

This section summarizes the reviewed papers and those which relatively have a higher connection to this work.

Table 1: Summary of Review of Related Works

No	Author	Objective of the Paper	Methodology Employed	Detection	Prevention

1	Qing Li et al. [18]	To detect device spoofing with little or no dependency on cryptographic keys.	NetSim	✓	×
2	F. Guo et al. [16]	To detect al existing spoofing attacks without any false positive or negative.	Sequence Number Based Spoof Detection Algorithm	✓	×
3	A. Mitrokotsa et al. [37]	To investigate how hyper parameter tuning affected performance in a mobile ad hoc network.	Comparing the test system to a pre-defined cost matrix system.	✓	✓
4	B. Wernik et al. [40]	To detect and prevent ARP spoofing attacks in mobile ad hoc network	ARP lease files which acted as a barrier for attackers.	✓	✓
5	P. Goyal et al. [39]	To mitigate ARP flooding attacks in neural networks	Taking advantage of the AODV protocols specification or sending out fake control packets.	✓	×
6	B. Wu et al. [38]	To compensate the absence of a centralized management system in MANETs	Studied th e vulnerabilities, challenges, and attacks	×	✓

Chapter Three
Methodology

This chapter presents the structure of the methodology employed in achieving this research work. The general research approach used for the implementation of this work was the Design Science approach. Steps including Awareness of the problem, Suggestion, Development, Evaluation and Conclusion have been followed. The main emphasis when developing the proposed method was the structure of the method and the evaluation criteria. The evaluation approach used was a simulation procedure which can prove the appropriateness and consistency of the proposed algorithm.

The proposed algorithm was developed by dividing it into parts and designing those parts independently. Each part had a specific task to accomplish and then those parts were integrated to come up with the comprehensive version of the algorithm. The programming language used for writing the proposed algorithm was the C. The network simulator used for modeling and simulation purposes was the OPNET, Optimized Network Engineering Tools. The major three tools of OPNET are a Modeling tool, Simulation and Execution tool and Result Analysis tool. The Modeling tool took the total number of nodes, IP and MAC addresses of each node, the proposed algorithm as inputs. Then it designed a model which operated as per the inputs it has taken. The Simulation and Execution tool, which served as an evaluation tool for this thesis work, was used to study the network performance and behavior with five test scenarios. The Result Analysis tool was used to observe the packet transaction throughout the network after the simulation execution has been completed.

The proposed algorithm was designed as to detect malicious nodes by creating a conjugate ARP table, called ARP lease file, which acts as a barrier to the ARP cache table. These tables are built using gratuitous ARP packets. A gratuitous ARP packet is an ARP packet that does not require a reply and is used as part of route maintenance portion of a network.

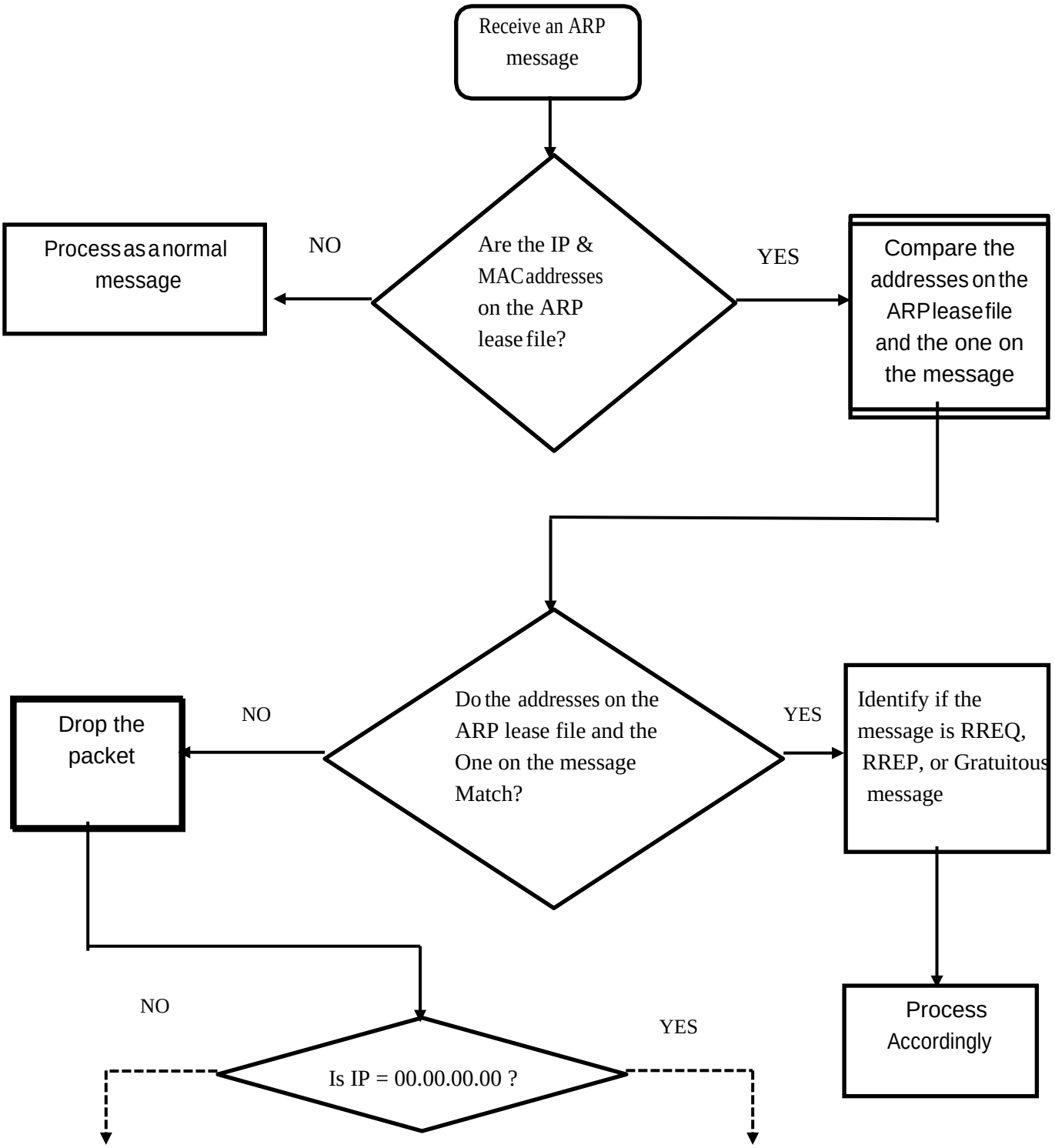
Whenever a new node joins a network it is required to broadcast gratuitous ARP message. Whoever receives such gratuitous ARP message is required to rebroadcast it and then broadcast its own gratuitous ARP message. Nodes which already have the information about the node that just sent a gratuitous ARP message will have to drop the packet. This helps mitigate packet storm problem. Both ARP cache table and ARP lease file do not have TTL, which means no entry will be removed from the table once registered.

The ARP tables have the added benefit of securing the route cache tables. Anytime a node flags a

message as being malicious, it uses the IP address it actually has mapped in its ARP lease file to the MAC in the message to remove any routes in its route cache that include the pairing in the ARP lease table. As a result, nodes can collude to cut off an attacker from the network by assuring that data is never routed to them on any alternate path. We also eliminate the need for resolving the MAC address of one hop neighboring nodes.

At normal operating system, when a node joins a network, it is required to broadcast a gratuitous ARP message to all other nodes in the transmission network. The following flowchart depicts how a gratuitous ARP message is processed whenever a node broadcast it. The steps taken by each node to process a gratuitous ARP packet in our algorithm are outlined in Figure 1, which later have been converted into a C code and fed to the modeling kernel of the OPNET. The code can be found in Appendix 1. A node can be any wireless communication device including but not limited to mobile phone, laptop, tablets, personal digital assistance, MP3 player or personal computer. For simulation purposes laptops has been chosen to be the nodes for this particular work.

3.1. Steps Taken by Each Node to Process A Gratuitous Packet



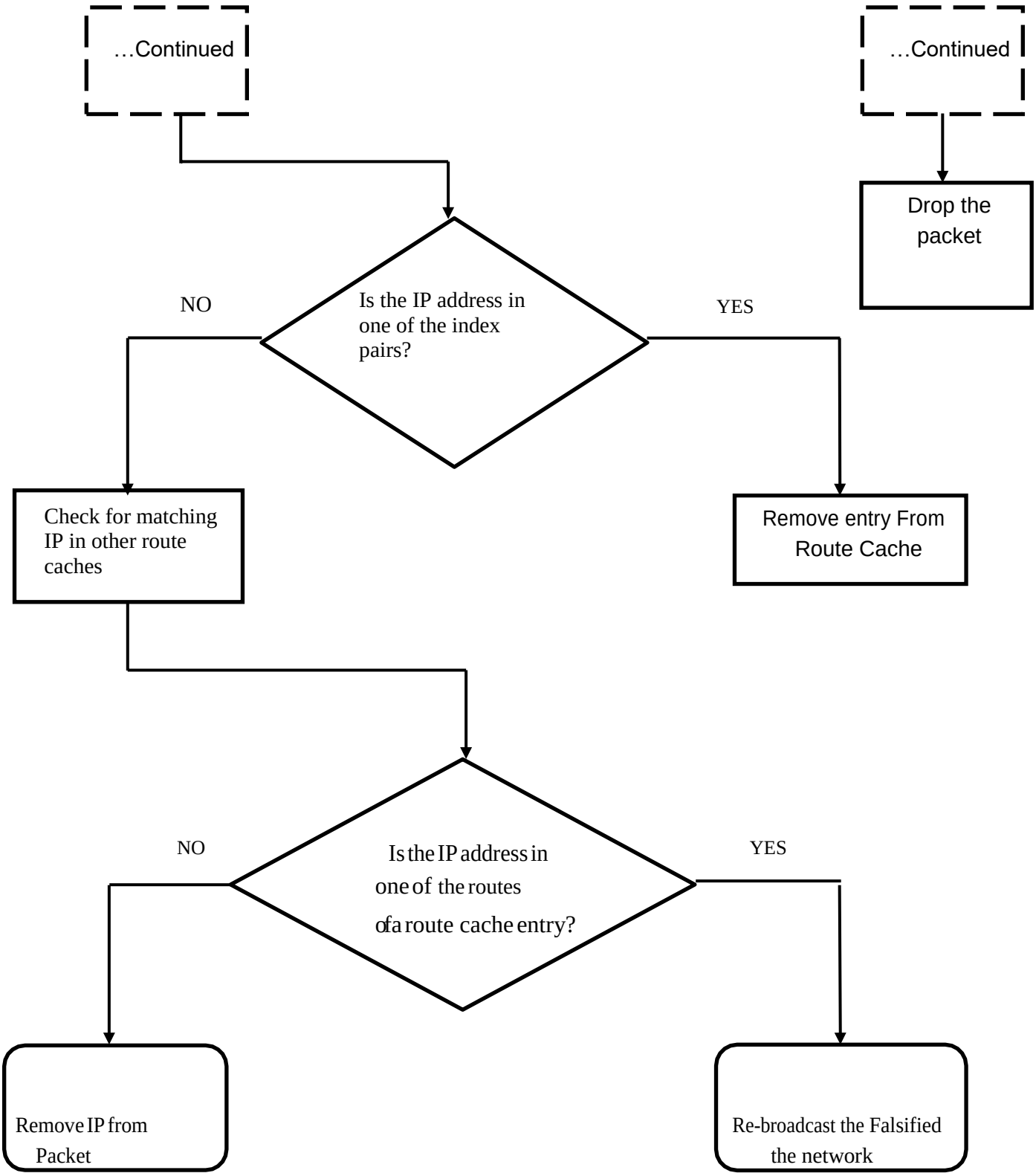


Figure 1: Blacklist and Removal of Entry from Tables

By not re-broadcasting the message and by removing the entries in the tables the algorithm helps prevent the spread of the attack to neighboring nodes and can effectively cut off the attacker from the network by not providing it with any routes to communicate with. Such nodes would only be able to receive and send broadcast messages, not any data flow.

3.2. Pseudocode of The Algorithm

3.2.a ARP Lease File Construction

Whenever a node receives an ARP packet with a header, the tasks to be implemented are depicted in the below pseudocode.

```
Begin
for IP1 up to IP8 and MAC1 up to MAC8
    if source IP = IP1 and source MAC = MAC1
        Process ARP packet as a request
        Reply accordingly
   else
       Drop the packet
   endif
 Do this for all the nodes
 end for
Stop
```

3.2.b Java code for Getting Malicious Node Blacklisted

```
Begin
    for IP1 up to IP8 and MAC1 up to MAC8
        if IP1 = 00.00.00.00
            Drop packet
        else if MAC1 exist in the ARP file
             Remove entry from cache
        else if IP1 is in one of the routes of a route cache entry
              Remove IP from route and any IP's following it.
              Re-broadcast the falsified packet.
        else
            Re-broadcast the falsified packet.
        endif
      Do this for all the nodes
      end for
    Stop
```

Chapter Four
Simulation, Results and Discussions

This section describes how the simulations of the proposed algorithm was setup on the Simulation tool of the OPNET. After having the Modeling tool of OPNET all set, the next step was to simulate and the test the effectiveness of the proposed algorithm using the Simulation kernel of OPNET. Different test scenarios have been constructed on the Simulator tool to test the performance of the model built in the Modeling tool.

4.1. Simulation Setup

The simulation tool simulates the DSR, ARP and IP protocol functions needed for our purpose and it is run on a Windows operating system environment. All traffic is passed over the same network device on the same system. The network to be simulated was assumed to have multiple nodes, each of which was run in separate terminal windows. When an attacker is introduced to the system it will be given a falsified IP-MAC mapping. Each node has its own static IP and MAC address assigned. The traffic that was subjected to the simulating software was only ARP and DSR packets, which means the traffic of any actual data transmissions has not been considered. Only the RREPs have been simulated and the routes have statically been maintained in each instance of a node.

Nodes that are out of the wireless range were configured in such a way that they can send ARP and DSR packets, but their packets will not be processed. To measure the success of the algorithm it is important to note whether the attacker is successful in poisoning any of the simulated ARP cache entries on each node. The number of packets that are exchanged during the initial start-up portion of the algorithm where every node is passing around gratuitous ARP packets have also been recorded. This will be used to mathematically measure against regular initialization time in a MANET that employs the proposed algorithm, without a malicious node to determine how much of an impact the algorithm has.

4.2. Test Scenarios

This section describes the test scenarios used to evaluate the success of the proposed algorithm. The first scenario simulated an uncorrupted network which has only honest nodes and not

malicious ones. This was important so that we grasp the estimated of the number of packets exchanged in a normal operating environment. The rest of the scenarios include some chosen instances of a malicious node being placed at different positions and perpetrating ARP spoofing on the network. Each link between nodes has been given a conceptual distance of 100 meters.

4.2.1. Scenario 1: No Malicious Node

In this scenario there is no malicious node. The nodes that have a double-sided arrow in between them were considered to be within the wireless range of the network. The ones that are not connected by such arrows are deemed out of wireless range of each other.

Topology:

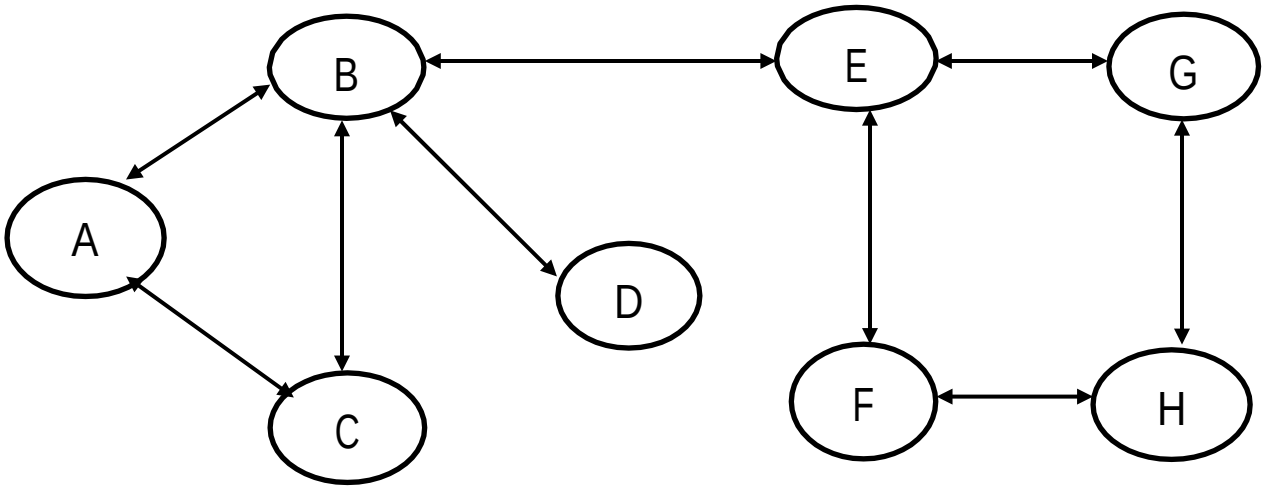


Figure 2: Testing Scenario 1 Network topology

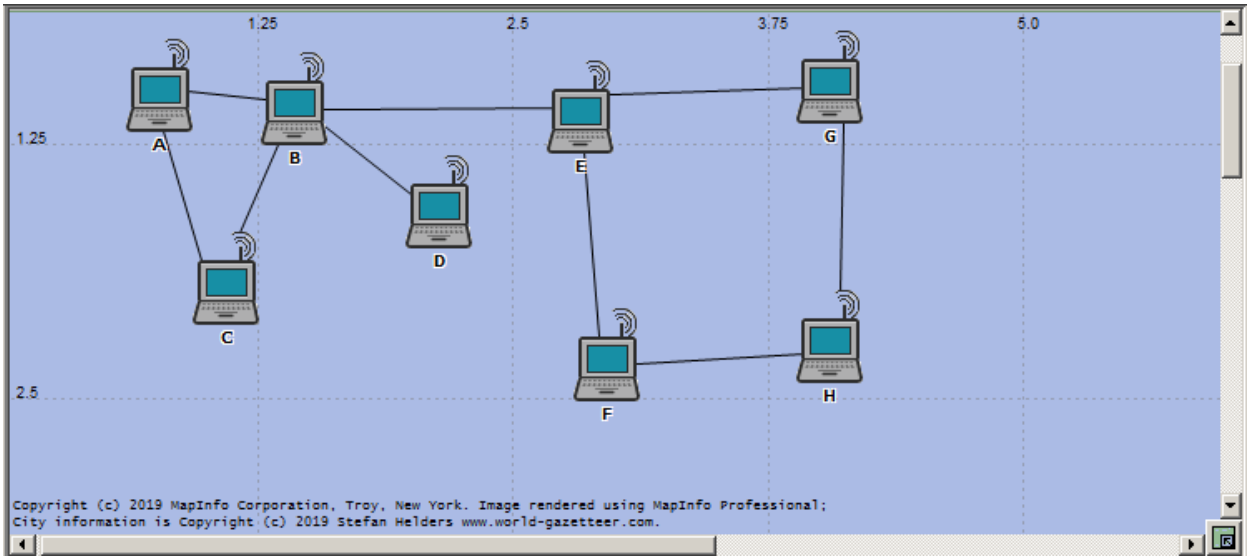


Figure 3: OPNET Screenshot Scenario 1

Process:

The simulation is designed to proceed as follows:

1. The nodes enter the network in their alphabetical order, A being the first one to join the network.

2. All nodes build their respective ARP lease file based on the topology. This step enables us to register the number of packets exchanged and dropped.

3. After the ARP lease files were built, the route discovery portion of the DSR protocol was simulated which made it possible to know the number of ARP protocol packers were exchanged to create the route paths.

Expected Outcome:

No attack is expected to take place. No ARP packets are expected to be passed during the RREP stage.

4.2.2. Scenario 2: Malicious Node is Last to Enter Network

In this scenario the malicious node M enters the network after it has already been established as described in scenario 1. For the nodes that are within the wireless range with each other a bidirectional arrow is put to represent the communication between them. Those which are out of the wireless range with each other cannot exchange packets.

Topology:

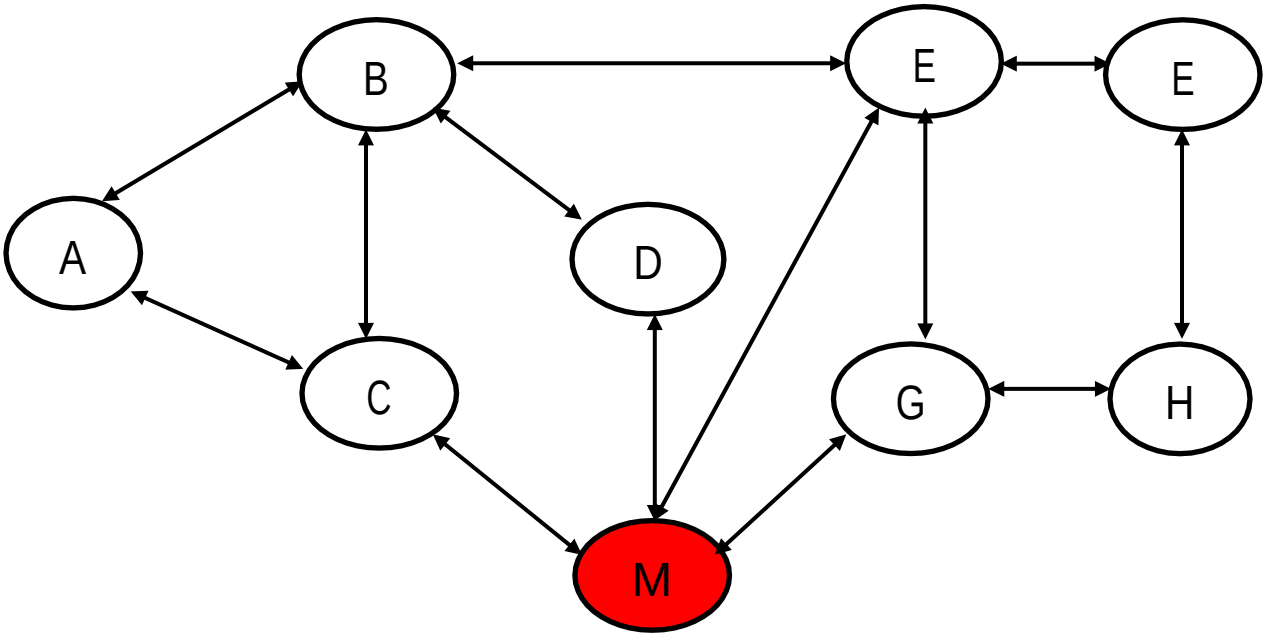


Figure 4: Testing Scenario 2 Network topology

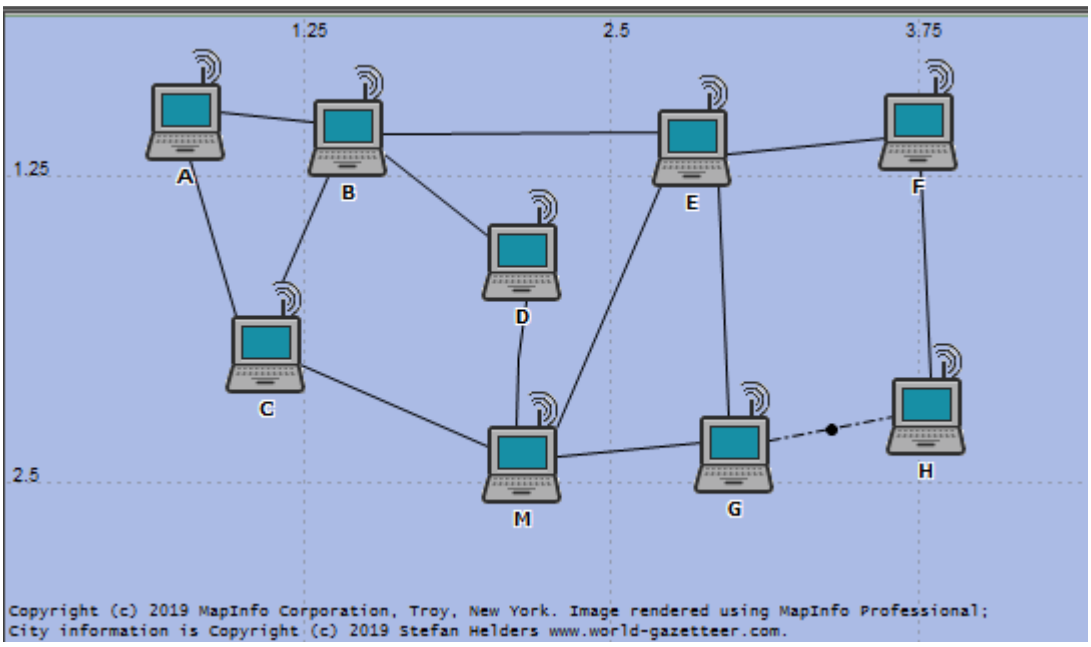


Figure 5: OPNET Screenshot Scenario 2

Process:

The implementation will be as follows:

1. The nodes enter the network in their alphabetical order, A being the first one to join the network.

2. All nodes build their respective ARP lease file based on the topology. This step enables us to register the number of packets exchanged and dropped.

3. Node G & H replies accordingly with a RREP.

Expected Outcome:

At this point, in a normal ad hoc network employing the DSR protocol, node A and all nodes on route back to G & H would need to resolve the next hop MAC address to pass on the RREP message. This would require them to send ARP requests that could result in spoofed ARP replies from malicious node M however since the addresses of next hop neighbors have already been resolved there is no need for this. As a result, the packets are unicast accordingly, and the attack will be unsuccessful.

4.2.3 Scenario 3: Malicious Node Enters Before Node H

In this scenario the malicious node M enters the network before node H.

Topology:

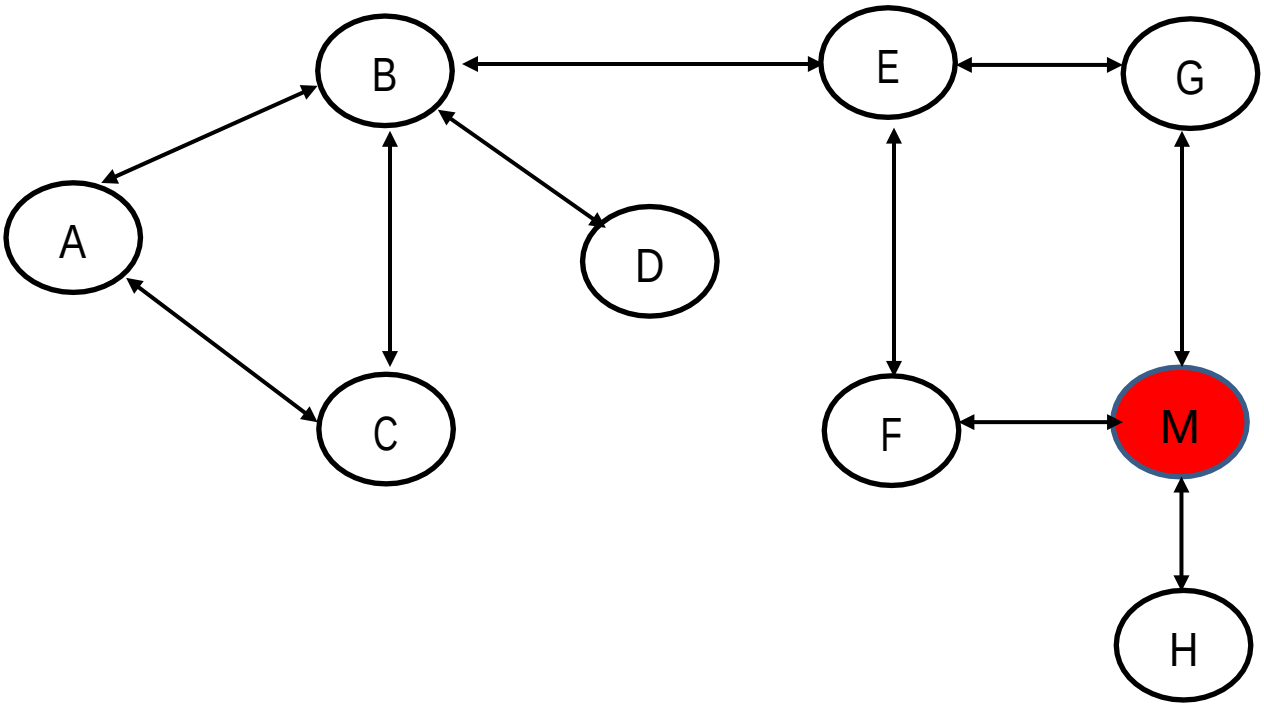


Figure 6: Testing Scenario 3 Network topology

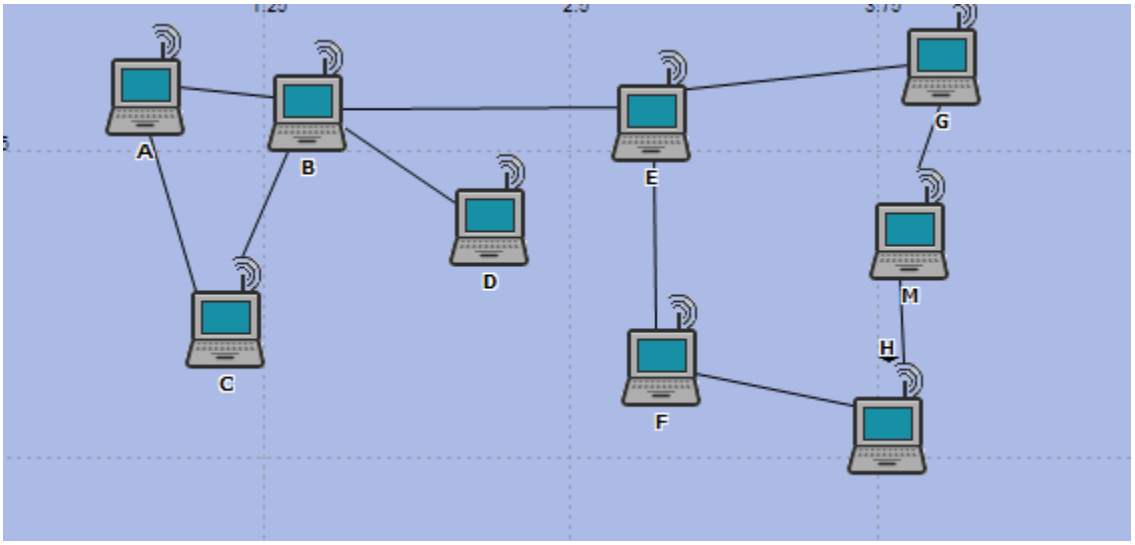


Figure 7: OPNET Screenshot Scenario 3

Process:

The implementation is as follows:

1. All nodes except H enter the network in their alphabetical order, A being the first one to join the network.

2. All nodes build their respective ARP lease file based on the topology. This step enables us to register the number of packets exchanged and dropped.

3. Node M, trying to mimic an honest node will broadcast a gratuitous ARP message.

4. H then enters the network and tries to broadcast its gratuitous ARP message.

5. M drops H's gratuitous ARP message.

Expected Outcome:

The expected result of this scenario is expected two schemes. Considering M as an honest node, it is going to be required to broadcast its gratuitous ARP message to all the nodes within the wireless range including H. Then the spoofing process will start by poisoning H’s cache with fake MAC-IP mappings without being aware that this action gets it backlisted by the other nodes within the wireless range; by E, F and G. This in turn results in M being cut off from the network since its only contacts were E, F and G.

4.2.4. Scenario 4: Malicious Node Enters Before Nodes G, F and H

This scenario describes a situation where the attacker, node M, enters the network prior to two of the honest nodes; G, F and H.

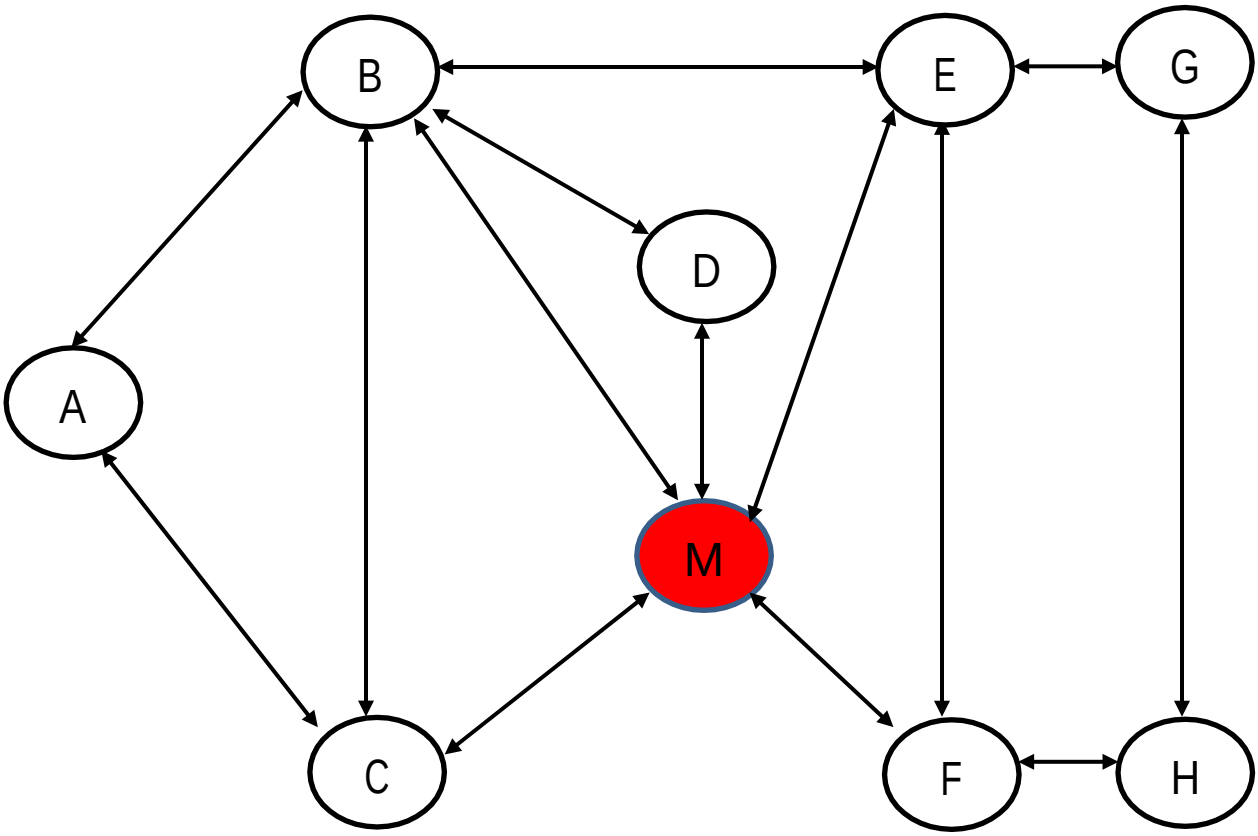


Figure 8: Testing scenario 4 Network Topology

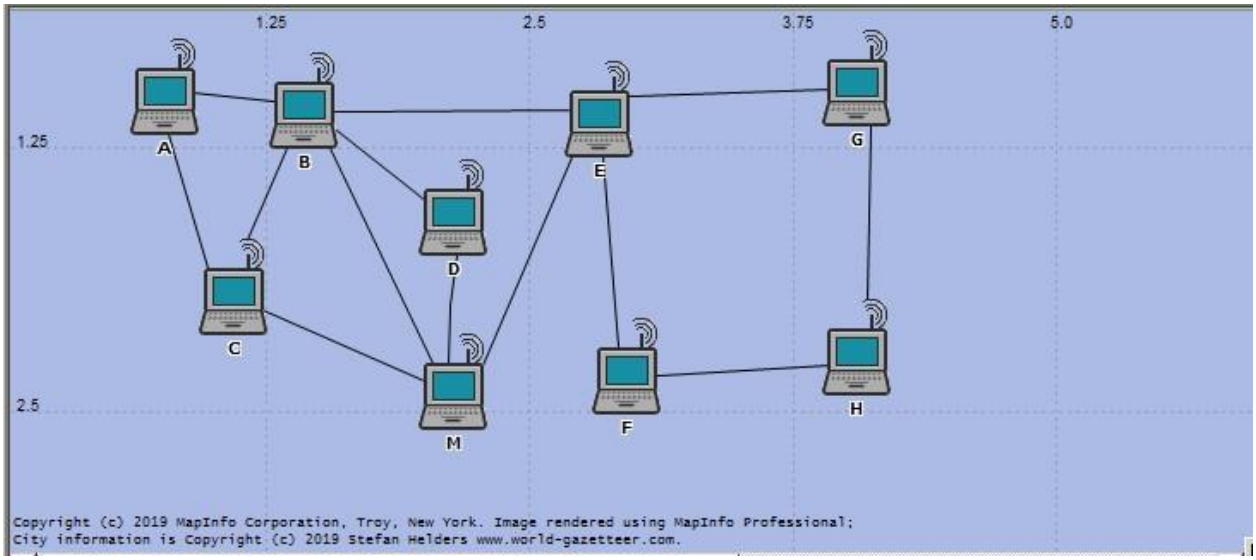


Figure 9: OPNET Screenshot Scenario 4

Process:

The implementation proceeds like:

1. Nodes A, B, C, D, & E enter the network, then the malicious node M enters the network.
2. Node that have already joined the network advertise themselves through a gratuitous ARP message.
3. Nodes F, G, & H enter the network.
4. Node F & G advertise themselves through a gratuitous ARP message.
5. At this point node H tries to initialize itself and broadcast a gratuitous ARP message to the nodes that are in its wireless range.
6. All nodes except H build their respective ARP lease file based on the topology. This step enables us to register the number of packets exchanged and dropped.
7. When malicious node M receives H's broadcast message, it attempts to poison the network by re-broadcasting the original message with a fake MAC-IP mapping.

Expected Outcome:

The expected result is more like that of scenario 2, with a little bit more gratuitous ARP message traffic being passed throughout the network which is caused by the increase in the number of routes. M too is expected to best out of the network as a result of trying to fake F's initializing message.

4.2.5. Scenario 5: Malicious Node Attempts to Poison Caches with Gratuitous ARP

This scenario describes a situation where the attacker, node M, attempts to poison the cache entries of all nodes by broadcasting a gratuitous ARP message with a falsified mapping. The network topology is the same as the one used in scenario 4.

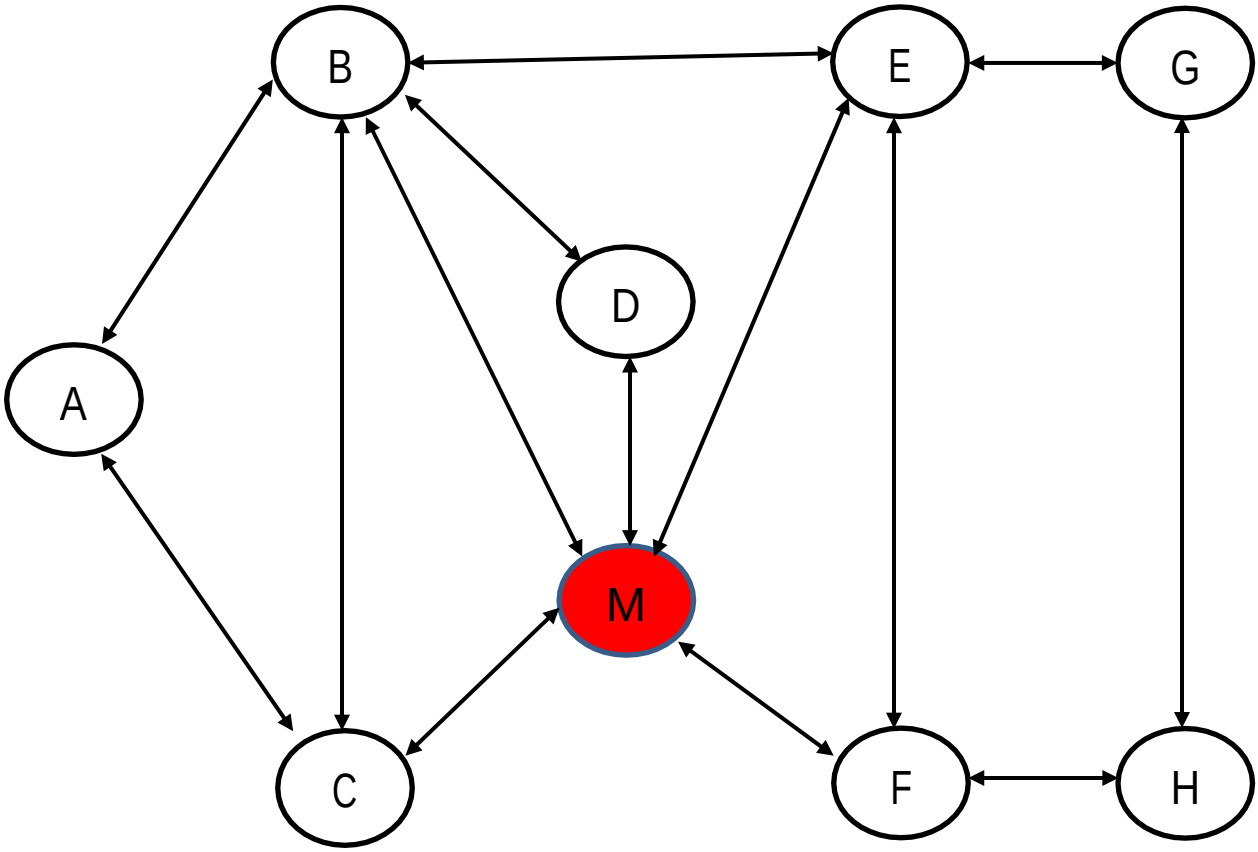


Figure 10: Testing scenario 5 Network Topology

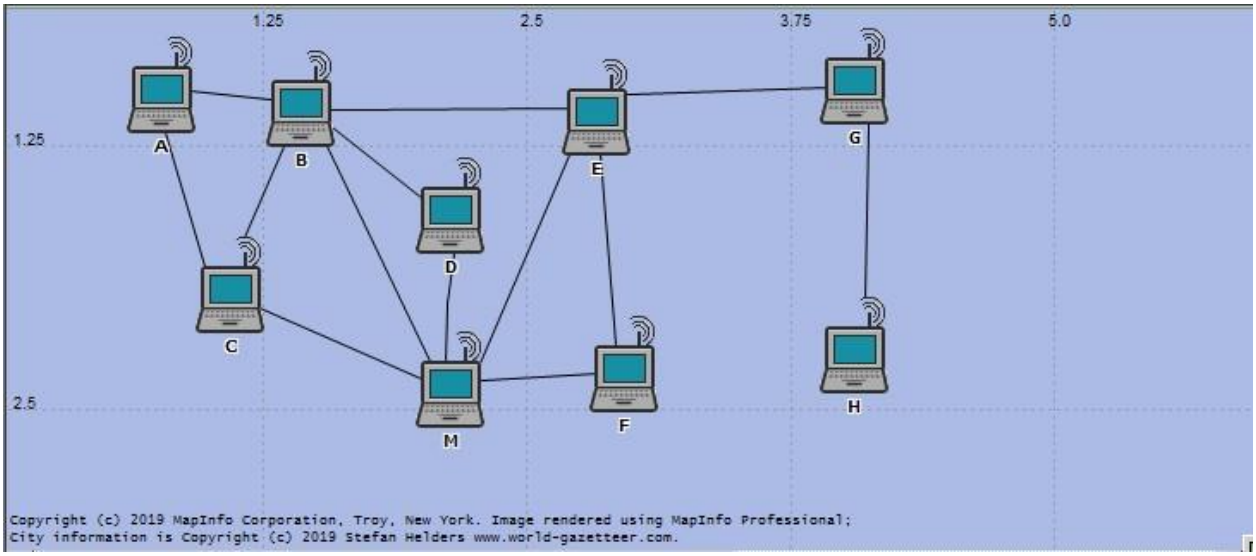


Figure 11: OPNET Screenshot Scenario 5

Process:

The implementation is as follows:

1. Nodes A, B, C, D & E enter the network, then the malicious node M enters the network.
2. Node that have already joined the network advertise themselves through a gratuitous ARP message.
3. Nodes F, G & H enter the network.
4. Node F & G advertise themselves through a gratuitous ARP message.
5. At this point node H tries to initialize itself and broadcast a gratuitous ARP message to the nodes that are in its wireless range.
6. All nodes except H build their respective ARP lease file based on the topology. This step enables us to register the number of packets exchanged and dropped.
7. M then attempts to poison the ARP caches of its one hop neighbors through a falsified gratuitous ARP message.

Expected Outcome:

The expected result is for M to be blacklisted by all the neighbors and to be blocked out of the network.

4.3. Assumptions Taken

This section lists the assumptions that were made before proceeding with the simulation.

- The first to be noted is the only traffic in the network tested in the test scenarios are the ARP and the DSR traffic.
- The next basic assumption made was that the nodes are of moving types, i.e. the routes discussed in the above test scenarios change whenever the nodes change their positions in the topology. Whenever a node changes its position in the network, it will rebroadcast a new gratuitous message to its new neighbors. Then the normal process discussed in the test scenario will proceed.
- Thirdly it has also been assumed that all links between nodes have the same distance of 100meters. Fixing the distance to a certain amount was important due to the direct relation of signal strength and distance have in wireless networks. This data has been fed to the OPNET modeler.

4.4. Test Results

This section describes the results of the test scenarios that were discussed in section 4.2.

4.4.1. Results from Scenario 1: No Malicious Node

The main aim of simulating this scenario was to measure the number of paces exchanged when nodes are being connected with one another. Having this count at hand we were able to make comparisons to that of the scenarios where malicious nodes were introduced. It is also used to confirm that no ARP request and reply messages are exchanged when sending a RREP from node F back to node A. Table 2 depicts the results gained from the simulation.

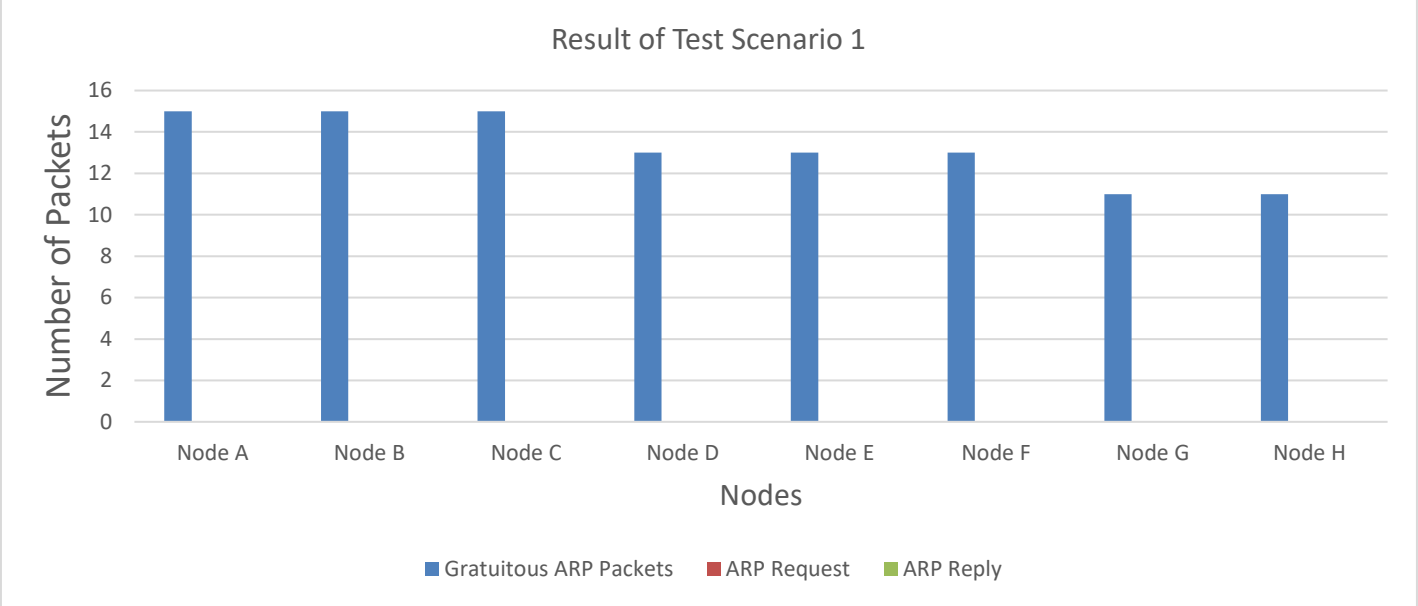


Figure 12: Result of Test Scenario 1

Here it should be noted that since the test scenario does not have any malicious node there are no ARP Request and ARP Reply packets as expected. And the total number of packets exchanged during the initialization process is 21. This number was used to compare the effect of the proposed algorithm when a malicious node is introduced into the network.

4.4.2. Results from Scenario 2: Malicious Node Enters the Network Last

The network topology of this test scenario has been modified from the first one. Here we have a malicious node introduced into the network which increases the number of gratuitous ARP packets being exchanged in the network. The test results of the test algorithm are summarized in the below table.

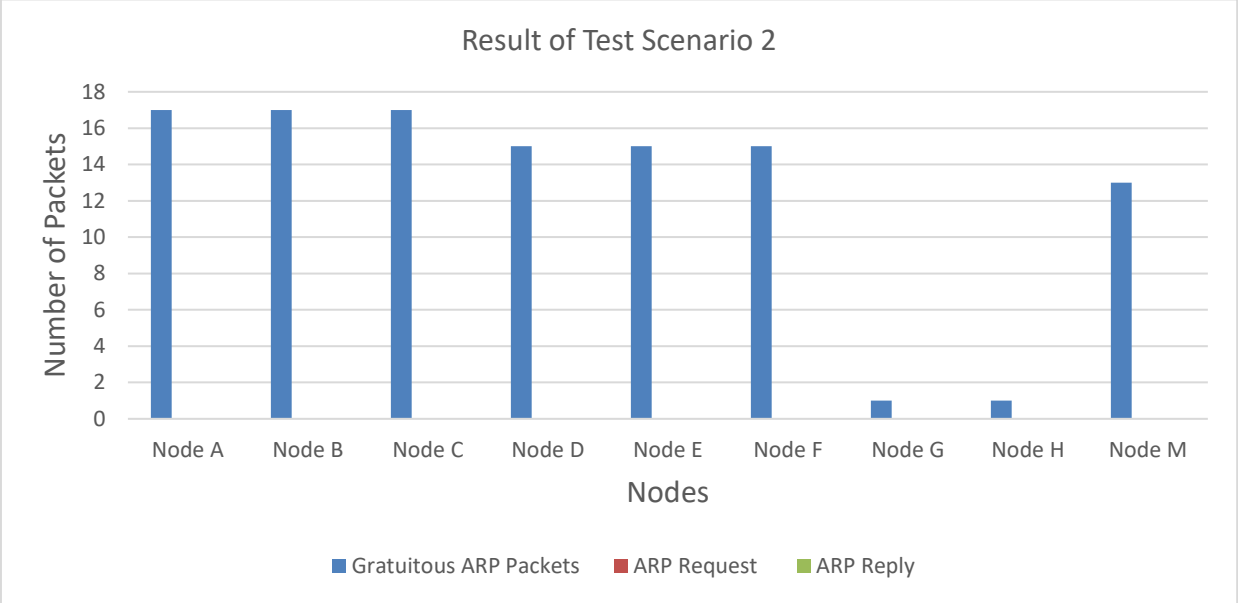


Figure 13: Result of Test Scenario 2

As it can easily be grasped from the table, there were no ARP requests or reply packets during RREP stage. This fact closed all the opportunity for the attacker to do any kind of spoofing attacks, since there is no address resolution taking place. This implies the algorithm was successful in preventing ARP spoofing in this specific scenario.

4.4.3 Results from Scenario 3: Malicious Node Enters Before H

Table 4 below summarizes the test results of scenario 3. The number of packets exchanged in the network has increased as compared to that of scenario 1. But it was less than the number of packets that were exchanged in the case of scenario 2. This was caused since in scenario 2 whenever nodes receive a falsified gratuitous ARP, they do not re-broadcast their own gratuitous ARPs. Instead they broadcast the malicious packet to all other nodes so that all the other nodes blacklist the malicious node as soon as they receive the malicious broadcast packet.

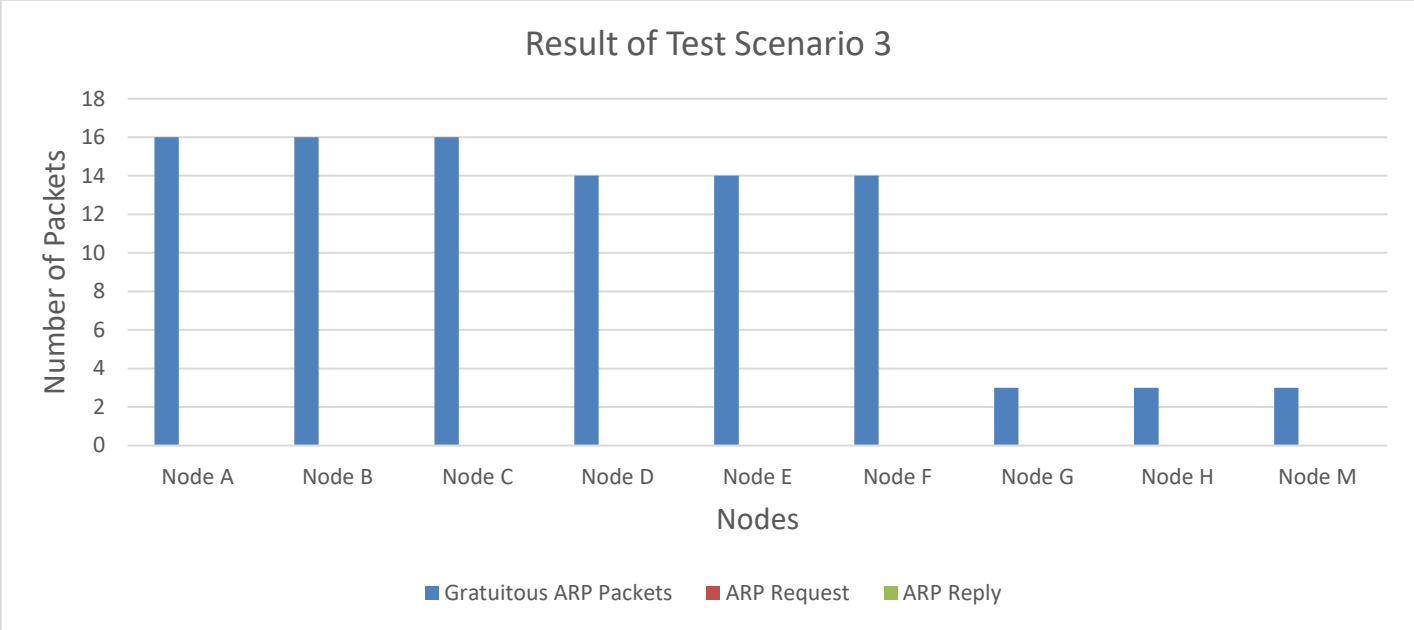


Figure 14: Result of Test Scenario 3

Here again there was no ARP request or reply packets during the RREP. During the initializing stage of node H, the malicious node, M, was able to poison the ARP cache of node H but it ended up getting itself and node H cut out of the network. This was caused by the fact that nodes F and G have received the falsified gratuitous ARP. Consequently, nodes F and G had successfully blacklisted node M from their respective and removed any routing entries intended towards this node. Then these two nodes re-broadcast the falsified message to make the rest of the nodes on the network to also blacklist the malicious node. The test has proved that the spoofing attack on node H was successful but the routes to node M have all been removed from the caches of all the other nodes.

4.4.4. Results from Scenario 4: Malicious Node Enters Before Nodes F, G and H

The test results of scenario 4 has been tabulated in the following table. There has been an increase in the number of packets exchanged in the network. This was due to topology change and the need to re-broadcast the falsified ARP packets to inform other nodes on the network of the malicious address.

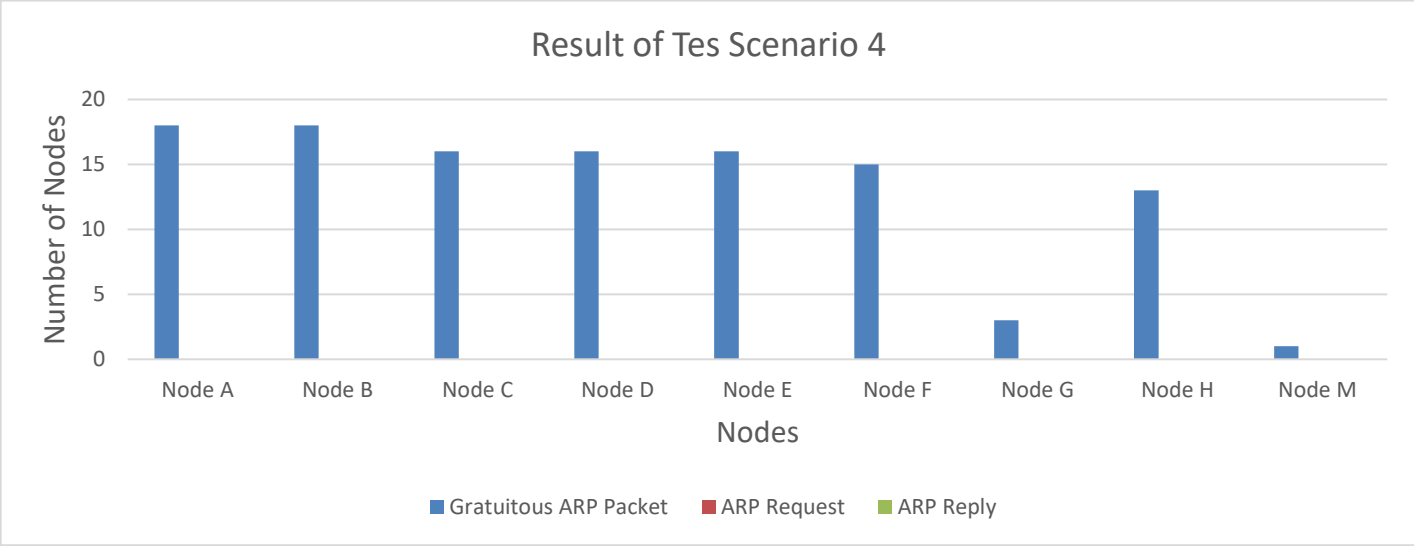


Figure 15: Result of Test Scenario 4

The result from this test scenario went like whenever M attempts to poison the ARP tables of its neighbors with a falsified message on behalf of F, G and H, all other nodes backlisted from their entries.

4.4.5. Results for Scenario 5: Malicious Node Attempts to Poison Caches with Gratuitous ARP

The results of the initialization stage can be summarized in Table 6. As with previous scenarios there was an increase in initialization packets sent compared to the number sent in scenario 1. This is due to change in routing topology.

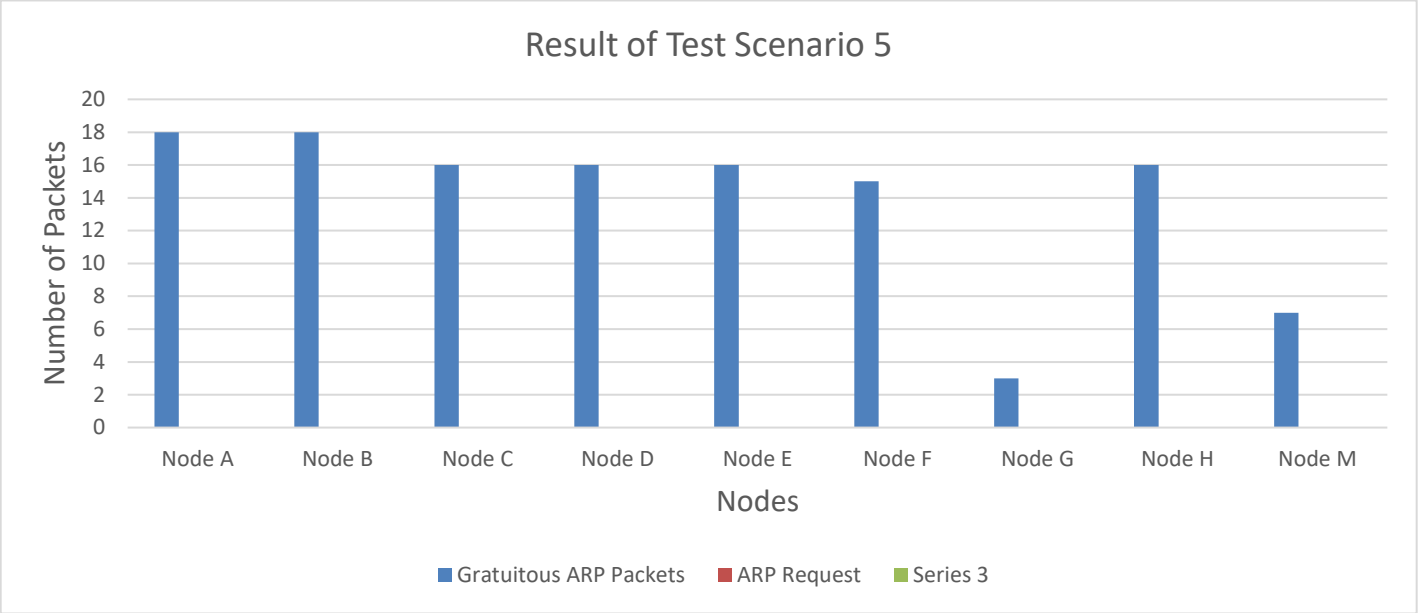


Figure 16: Result of Test Scenario 5

This scenario depicts the situation where malicious node M attempted to poison the ARP tables of its neighbors by broadcasting a falsified gratuitous ARP, after the whole network has been established. At this stage all the nodes about each other's ARP information which makes the chance for M to spoof a node's identity very little. As a result, it is cut off from the network with a minimal impact as all nodes are still reachable through alternate routes.

4.5. Comparative Analysis of Results and Discussions

As has been briefly discussed in the literature review chapter of this paper, many scholars have done their researches on how to mitigate ARP spoofing attacks in mobile ad hoc networks. It has been tried to compare two of such papers with our work to see the drawbacks and good sides of our algorithm.

B. Wernik [40] proposed an algorithm that used ARP protocol messages to build a table that acted as a barrier to the ARP cache table to prevent any false updates. They used gratuitous ARP packets to generate the filtering table. They simulated a network with nonmoving nodes and used NS2 for the simulation. They argued and proved that if there are no ARP packets being exchanged in a network there is a little to zero chance of spoofing attacks taking place. And they developed an algorithm that builds a table which substitutes the exchange of ARP packets. Then they proved that their algorithm does mitigate any suspected ARP spoofing attacks.

J. Binkley et. al. [33] used Internet Control Message Protocol (ICMP) Route Discovery packets to provide a solution to ARP spoofing attacks in mobile ad hoc networks. They augmented the ICMP packets so that the MAC addresses of the sending nodes were included on the ARP cache entries. Whenever a node enters the neighborhood of a group of nodes it broadcasts this ICMP packet. The size of each ICMP packet increases by 48 bits since that is the size of MAC addresses which was cascaded with it. This in turn will increase the transmission time for each packet. They took the example of encapsulation of DSR messages within IP packets and applied it to ICMP packets into IP protocol.

Their algorithm had made the packet being passed around the network to be of higher sizes and this has drawn a major drawback on the effectiveness of their algorithm. Originally the size of an IP header without the variable length options field being included is 20 bytes. Then there was the added size of the ICMP header which was 8 bytes. This equates to a total size of 28 bytes. Then the 6 bytes long MAC address was added and enlarged the packet to a size of 34 bytes. Next header to be cascaded with this 34-byte message was the 802.11 MAC header, which is 30 bytes. The total size of the final message summed up to be 64 bytes. And this was just the header of the message, not the actual message.

In contrast an ARP header has a size of 28 bytes and when encapsulated in a wireless MAC header it has a total size of 58 bytes. To make a comparison to our algorithm we have compared the total number of packets, for the initialization stages of each protocol, using the topology shown in Figure 1, with conceptual data and the previously made assumptions.

The simulation results showed that the time it takes for each individual message to be transmitted throughout the network in our algorithm was less than that of the time it took when simulating using the ICMP Route Discovery procedure. But due to the added time for the initialization stage the overall time taken for the nodes to be ready for further communication was slower than that of the ICMP Route Discovery procedure. The same result has been achieved on the algorithm B.Wernik [40] designed. The main benefit of the time lapsed for initialization stage was that nodes do not need move within range of each other to learn the MAC-IP mapping of the other nodes. Whenever they enter the network, they will broadcast a gratuitous ARP message, then they will be known by the other nodes thereby having all the MAC-IP mapping of the rest of the nodes. The table below depicts the summary of the papers including this one. The methodologies used, test scenarios used, simulation tools used, contribution and limitation are taken as aspects for comparison.

Table 2: Summary of Comparative Analysis of Results

No	Author	Methodology Employed	Test Scenario Used	Simulation Tool Used	Contribution	Limitation
1	B. Wernik et al	Used ARP packets that acted as a barrier for the ARP cache	The test scenarios were done on a network whose nodes are fixed and non-moving	NS2 Simulator	Paved the way on finding ways to mitigate ARP attacks without the need of ARP Request and ARP Reply packets to be exchanged.	The mitigation process does not apply when the nodes are of moving types.
2	J. Binkley et. al.	Used ICMP Route Discovery packets	The ICMP packets were augmented so that the MAC address of	NS2 Simulator	A lower number of Packets is exchanged throughout the network. This feature	There is an extended size of packets since the MAC addresses of

			each node is included in the ARP cache.		makes the nodes less busy.	the nodes are aggregated into the ARP cache.
3	Proposed Work	Used ARP lease files which acted as protective suit for the ARP cache of each node	The nodes used in the test scenario are assumed to have the tendency to leave their network whenever they get far away from the wireless range and join a new one.	OPNET Simulator	The security problem of moving nodes has been addressed in this paper. An attacker can attack only one node at most before it is cut out form the network.	Extra time lapsed for the initialization stage whenever a node joins a network.

Chapter Five
Conclusions and Future Works

5.1. Conclusions

Mobile ad hoc network is a highly effective and widely used means of communication. At its present state, however, it is an undeniable truth that mobile ad hoc networks are equally, if not more, vulnerable to ARP attacks as LANs are. And this requires making the system to be made more robust in order to prevent loss of data and maintain optimal network performance. Intrinsically, the routing information in MANETs is readily passed around by the infrastructure less network. And this feature creates a higher chance of the wireless transmissions to be eavesdropped. Thus, a mechanism is needed to mitigate such spoofing and impersonation attacks.

In this paper we proposed a strategy to counter the spoofing attacks in mobile ad hoc networks. The spoofing attack was mitigated by augmenting the already existing ARP protocol to build a filtering table, which we called ARP lease files. Whenever a packet is to be sent out of a node, these files will be checked to determine the next hop to be taken. These lease files can protect the ARP cache entries and each node uses its own lease file to blacklist a malicious node whenever detected. Then the malicious node will be cut off from the network and will be prevented from causing any abuse to the network. Whenever a node joins a MANET, it will initialize itself by broadcasting a gratuitous APR packet. Then they will automatically build their respective lease files thereby informing peer nodes if there has been a MAC-IP map spoofing attack and which node was the malicious node.

It is to be mentioned that there was an extra time lapsed due to the added initialization stage whenever a node joins a network. The added time is directly proportional to the size of the network, the architectural topology of the network, and the number of packets passed around the network. This is the main trade-off for the added security that the algorithm provides. To investigate the effectiveness of the proposed algorithm we first simulated the network without any malicious node, took a grasp of the added time for the initialization stage and the number of packets exchanged during the process. Then the four different network scenarios were simulated redundantly. The proposed algorithm had successfully detected and prevented the spoofing

attack. The malicious node was only able to spoof the identity of one node which was its closest instance. And consequently, it was cut off from the network and blacklisted by the rest of the nodes.

The chance of an ARP spoofing attack was diminished since there are no ARP request or reply packets passed through the network. This provides the attackers much less opportunity for the attacks. This resulted in the risk minimization of MANETs of being spoofed.

5.2. Future Works

For future works one can enlarge the algorithm proposed in this paper with scenarios with multiple number of malicious nodes than ours. This is suggested because as the number of malicious nodes increases, there might be a principal spoofing mitigation change on the network and the nodes would need to explicitly be configured how to process simultaneous attacks.

Appendices

A. Appendix A -Codes

A.1. C code for ARP Lease File Construction

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <netinet/in.h>
#include <net/if.h>
#include <arpa/inet.h>
#include<stdlib.h>

int main()
{
int *ptr;  /*declaration of integer pointer*/
int limit; /*to store array limit*/
        int i; /*loop counter*/
        int sum; /*to store sum of all elements*/

      scanf("%d",&limit);
 /*declare memory dynamically*/
        ptr=(int*)malloc(limit*sizeof(int));
/*read array elements*/
        for(i=0;i<limit;i++)
        {
                printf("Enter element %02d: ",i+1);
                scanf("%d",(ptr+i));
        }

        /*print array elements*/
        printf("\nEntered array elements are:\n");
        for(i=0;i<limit;i++)
        {
                printf("%d\n",*(ptr+i));
        }
     /*calculate sum of all elements*/
        sum=0; /*assign 0 to replace garbage value*/
        for(i=0;i<limit;i++)
        {
                sum+=*(ptr+i);
        }
        printf("Sum of array elements is: %d\n",sum);

        /*free memory*/
        free(ptr);          .
```

```
        return 0;
}
   unsigned char ip_address[15];
   int fd;
   struct ifreq ifr;

   /*AF_INET - to define network interface IPv4*/
   /*Creating soket for it.*/
   fd = socket(AF_INET, SOCK_DGRAM, 0);

   /*AF_INET - to define IPv4 Address type.*/
   ifr.ifr_addr.sa_family = AF_INET;


   /*eth0 - define the ifr_name - port name
   where network attached.*/
   memcpy(ifr.ifr_name, "eth0", IFNAMSIZ-1);

   /*Accessing network interface information by
   passing address using ioctl.*/
   ioctl(fd, SIOCGIFADDR, &ifr);
   /*closing fd*/
   close(fd);
   /*Extract IP Address*/
   strcpy(ip_address,inet_ntoa(((struct sockaddr_in *)&ifr.ifr_addr)->sin_addr));

   printf("System IP Address is: %s\n",ip_address);

   return 0;
}
 /*Checking whether or not the IP address is on the ARP lease file*/
{
  int array[100], search, IP, n;

 printf("Enter number of elements in array\n");
 scanf("%d", &n);

 printf("Enter %d integer(s)\n", n);

 for (IP = 0; IP < n; IP++)
   scanf("%d", &array[IP]);

 printf("Enter a number to search\n");
 scanf("%d", &search);

 for (IP = 0; IP < n; IP++)
 {
   if (array[IP] == search)    /* If required IP is found */
   {
     printf("%d is present at location %d.\n", search, IP+1);
     break;
   }
 }
```

```
  if (IP == n)
    printf("%d isn't present in the array.\n", search);
return 0;
}
 /*Checking whether or not the MAC address is on the ARP lease file*/
{
   int array[100], search, MAC, n;

  printf("Enter number of elements in array\n");
  scanf("%d", &n);

  printf("Enter %d integer(s)\n", n);

  for (MAC = 0; MAC < n; MAC++)
    scanf("%d", &array[MAC]);

  printf("Enter a number to search\n");
  scanf("%d", &search);

  for (MAC = 0; MAC < n; MAC++)
  {
    if (array[MAC] == search)    /* If required MAC is found */
    {
      printf("%d is present at location %d.\n", search, MAC+1);
      break;
    }
  }
  if (MAC == n)
    printf("%d isn't present in the array.\n", search);
return 0;
}
/*Checking whether or not the IP address has been blacklisted*/
{
   int array[100], search, IP, n;

  printf("Enter number of elements in array\n");
  scanf("%d", &n);

  printf("Enter %d integer(s)\n", n);

  for (IP = 0; IP < n; IP++)
    scanf("%d", &array[IP]);

  printf("Enter a number to search\n");
  scanf("%d", &search);

  for (IP = 0; IP < n; IP++)
  {
    if (array[IP] == search)    /* If required IP is found in the list of the blacklists */
    {
      printf("%d is present at location %d.\n", search, IP+1);
      break;
```

```
    }
  }
  if (IP == n)
    printf("%d isn't present in the array.\n", search);
return

}
```

A.2. Java code for Getting Malicious Node Blacklisted

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.Random;
import java.io.File;
import java.io.IOException;
import weka.classifiers.Evaluation;
import weka.classifiers.functions.MultilayerPerceptron;
import weka.core.Instances;
import java.io.*;
import java.util.*;
import java.util.regex.*;
import weka.classifiers.Evaluation;
import weka.classifiers.functions.MultilayerPerceptron;
import weka.core.Instances;
import weka.core.Utils;
import weka.filters.Filter;
import weka.filters.unsupervised.attribute.Reorder;


public class MitmProtectorWithWeka {
public static void main(String[] args)
 {
 try {
 PossibleFunctions pF = new PossibleFunctions();
 //String inputFile = new File("../mitmaodv/reducedFilteredResults.arff").getCanonicalPath();
String inputFile = new File("../mitmaodv/FINAL_DATA/POSSIBLE_FINAL_DATA/reduced Filtered Results
Preprocessed In Weka From Scale Down Attack With Latency Training Data With Packet Mode
.arff").getCanonicalPath();
 //String inputFile = new File("../mitmaodv/FINAL_DATA/furtherReducedtrainDataForJava. arff").getCanonicalPath();
String outputFile = new File("../mitmaodv/FINAL_DATA/further Reduced test Data For Java. a
rff").getCanonicalPath();
System.out.println("Input: " + inputFile); System.out.println("Output: " + outputFile);
BufferedReader trainer = new BufferedReader(new FileReader(inputFile)); BufferedReader tester = new
BufferedReader(new

package mitmprotectorwithweka;
public class PossibleFunctions {
static String blacklistWithBrackets; //this function predicts the classification by looking at the probabilities assigned to
each class and picking the maximum//
   public void classOfInstanceByProbability(MultilayerPerceptron mP,
```

```
Instances train, int instanceNo) throws Exception
{
 String packet = "";
double[] prediction = mP.distributionForInstance(train.get(instanceNo));
 int maxValueIndex = 0;
for (int i = 0; i < prediction.length; i = i + 1)
{
//System.out.println("Probability of class " + train.classAttribute().value(i)+":"+Double.toString(prediction[i]));
 if ((prediction[i] > prediction[maxValueIndex])) { maxValueIndex = i; packet = train.classAttribute().value(i);
} } System.out.print("\nclass " + train.classAttribute().value(maxValueIndex) + ":" + packet);
 }
public void printSummary(MultilayerPerceptron base, Evaluation eval
FileReader(outputFile));
Instances train = new Instances(trainer);//training data //
train = pF.removeColumnsData(train); //remove unwanted columns from training set and reorder remaining
columns//
Instances test = new Instances(tester); //testing data
train.setClassIndex(train.numAttributes() - 1);
test.setClassIndex(train.numAttributes() - 1);
trainer.close();
tester.close();
MultilayerPerceptron mP = new MultilayerPerceptron();
 mP.buildClassifier(train);
Evaluation eval = new Evaluation(train);
 eval.crossValidateModel(mP, train, 10, new Random(1));
pF.printSummary(mP, null, test); //pF.classOfInstanceByProbability(mP, train,1);
}
catch (Exception e) { e.getMessage();
}
 }
}
Instances data) throws Exception {
HashMap<String, Integer> counters = new HashMap<String, Integer>();
//output evaluation System.out.println(); System.out.println("=== Setup  ==="); System.out.println("Classifier:
" + base.getClass().getName() + " " + Utils.joinOptions(base.getOptions()));
System.out.println("Dataset: " + data.relationName());
int counterInitializer = 1;
 String blacklist = "{'none','";  //initialize this to none.
 for (int i = 0; i < data.numInstances(); i++)
 {
 //NB: OBSERVATIONS MADE FROM WEKA API
//System.out.print(" i0_4 : "+ data.instance(i).stringValue(data.attribute("t")));//did not work for numeric value
//
System.out.print(" idata.instance(i).value(data.attribute("t")));//but this works for numeric value //
System.out.print(" i1 : "+ data.instance(i).classAttribute());//for the classification attributes //
System.out.print(" i2 : "+ data.instance(i).numClasses());//for the number of classes
String pred = String.valueOf(base.classifyInstance(data.instance(i)));
double[] dist = base.distributionForInstance(data.instance(i));
double classifyInstance = base.classifyInstance(data.instance(i));
System.out.print((i + 1));
System.out.print("\tactual: " + data.classAttribute().value((int) data.instance(i).classValue()));
System.out.println(" \tpredicted :" + data.classAttribute().value((int) classifyInstance));
```

```
//for storing up how many times a particular node is sending out a falsified packet
 if (!(data.instance(i).stringValue(data.attribute("It")).equals("?")) &&//first condition for removing
unknown instances with unknown classes
(data.instance(i).stringValue(data.attribute("PM")).equals("s")) && (data.classAttribute().value((int)
classifyInstance).equalsIgnoreCase("aodv_mitm")))
{
 if (counters.containsKey(Double.toString(data.instance(i).value(data.attri bute("Hs")))))
 {
 counters.put(Double.toString(data.instance(i).value(data.attribute("Hs "))),
counters.get(Double.toString(data.instance(i).value(data.attribute("Hs" )))) + 1);
 }
else {
 counters.put(Double.toString(data.instance(i).value(data.attribute("Hs "))), counterInitializer);
}
}
/* System.out.print(" i4 : "+ data.instance(i).classValue());//for the value of the classes
System.out.print(" - "); //System.out.print(Utils.arrayToString(dist))
; System.out.print(" - ");
System.out.print(" i5 :"+Utils.maxIndex(dist)); System.out.print(" - ");
System.out.print(" i6 :"+classifyInstance);
  System.out.println();
 }
//for printing how many times a particular node is sent out packets Iterator<String> keySelector =
counters.keySet().iterator();
 while (keySelector.hasNext()) { String key = keySelector.next(); blacklist += key + ",";
//System.out.println("\n\nkey: "+key+" value: "+counters.get(key));
}
 blacklist += "}";//classing the list
//formatting 'blacklist' by removing the last comma before the closing brace and then writing to the file String
stringContainingPattern = blacklist;
blacklist = replaceWithRegex(",}","}",stringContainingPattern);//the last comma would be removed
String blackListWithSpaceDelimiters = replaceWithRegex(","," ",blacklist); System.out.println("blacklist:" +
blackListWithSpaceDelimiters);
 String inputFile = new File("../mitmaodv/MUTABLE_CODE/mutableScriptTemplate.tcl").
getCanonicalPath();
 String outputFile = new File("../mitmaodv/MUTABLE_CODE/mutableScriptTemplate_tem
p.tcl").getCanonicalPath();
String patternToReplaceInScript = "set.blacklist.\\{.*\\};";
String replacement1 = "set blacklist " + blackListWithSpaceDelimiters + ";";
editFileWithRegex(patternToReplaceInScript,replacement1,inputFile, outputFile);
//printing the blacklist to itsPerl script inputFile = new
File("../mitmaodv/setdestMutator.pl").getCanonicalPath();//original input;
file outputFile = new File("../mitmaodv/setdestMutator_temp.pl").getCanonicalPath();//for creating a temporary
output file that is finally rename to the original input file to replace the original file
public String replaceWithRegex
(
String patternToReplace,
String replacement,
String stringContainingPattern)
{
```

```
 Pattern p = Pattern.compile(patternToReplace);
Matcher m = p.matcher(stringContainingPattern);
 return m.replaceAll(replacement);
}
public void editFileWithRegex(
String patternToReplaceInScript,
String replacement1,
String editableFile,
String tmpFile) throws
 Exception
{
 String _newstring,_matchingString;
String _patternToReplaceInScript = patternToReplaceInScript;
String _replacement1 = replacement1;
String _editableFile = editableFile;
String _tmpFile = tmpFile; PrintWriter writer = new PrintWriter(new BufferedWriter(new
FileWriter(_tmpFile)));
 BufferedReader trainer = new BufferedReader(new FileReader(_editableFile));

try {
Pattern p1 = Pattern.compile(_patternToReplaceInScript); while ((_matchingString = trainer.readLine()) !=
null)
{ Matcher m1 = p1.matcher(_matchingString); _newstring = m1.replaceAll(_replacement1);
 //System.out.println(_newstring);
 writer.println(_newstring); writer.flush();
}
 }
catch (IOException e)
{
 e.getMessage();
 }
File realName = new File(_editableFile);
realName.delete();
new File(_tmpFile).renameTo(realName);
}
//for removing columns that are not wanted public Instances removeColumnsData(Instances data)
{
//remove and reordering these columns Instances newData = new Instances(data); String[] reorderOptions =
new String[2]; reorderOptions[0] = "-R"; reorderOptions[1] = "1,2,3,4,6,7,8,10,14,16,19,20,22,23,26,9";
Reorder reorderer = new Reorder();
 try{
//reordering the remaining columns reorderer.setOptions(reorderOptions); reorderer.setInputFormat(data);
//Instances newData =  Filter.useFilter(data, remove);
newData = Filter.useFilter(data,reorderer); //  for (int i = 0; i < newData.numInstances(); i++)
{
 //  System.out.println(" i2 : "+ newData.instance(i)); //for the number of classes      //  }

}
catch(Exception e){System.out.println(e.getMessage());} return newData; }}
```

B. Appendix B. Test Results

B.1. Scenario 1 Results

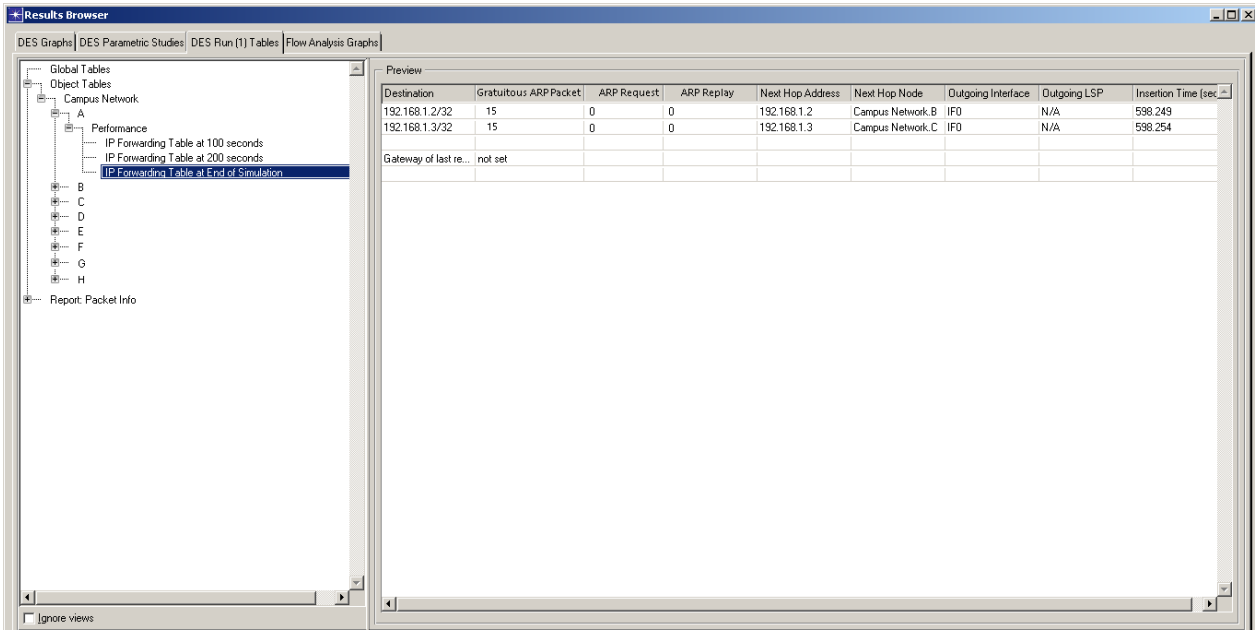


Figure 17: Node A Packet Exchange

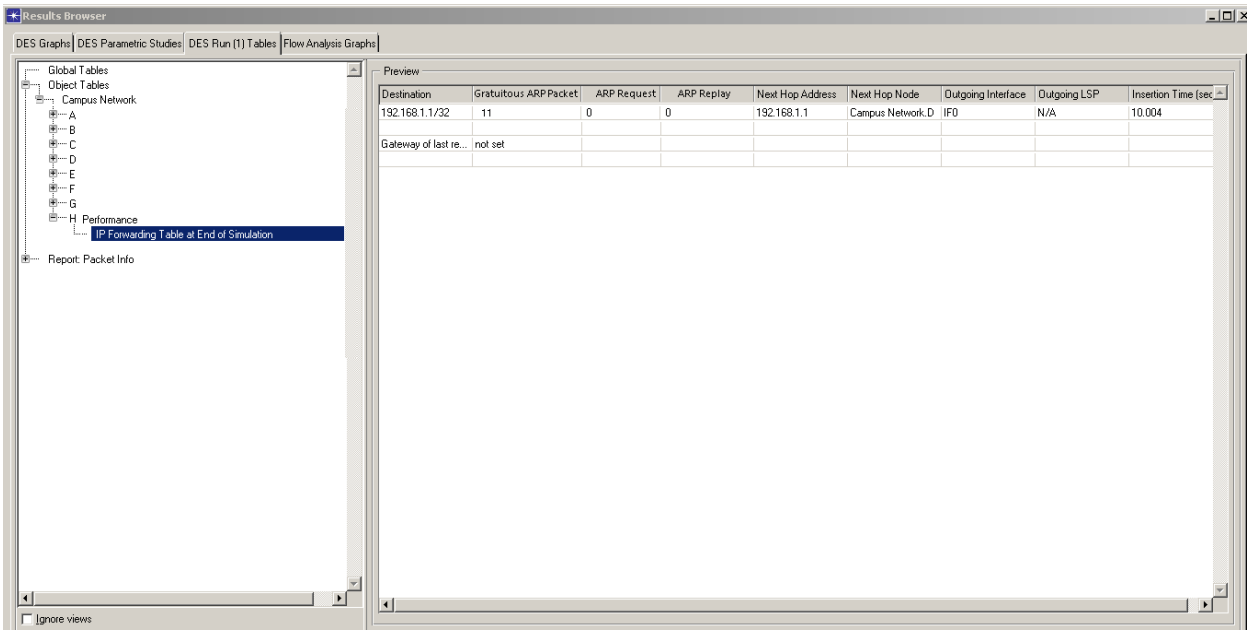


Figure 18: Node B Packet Exchange

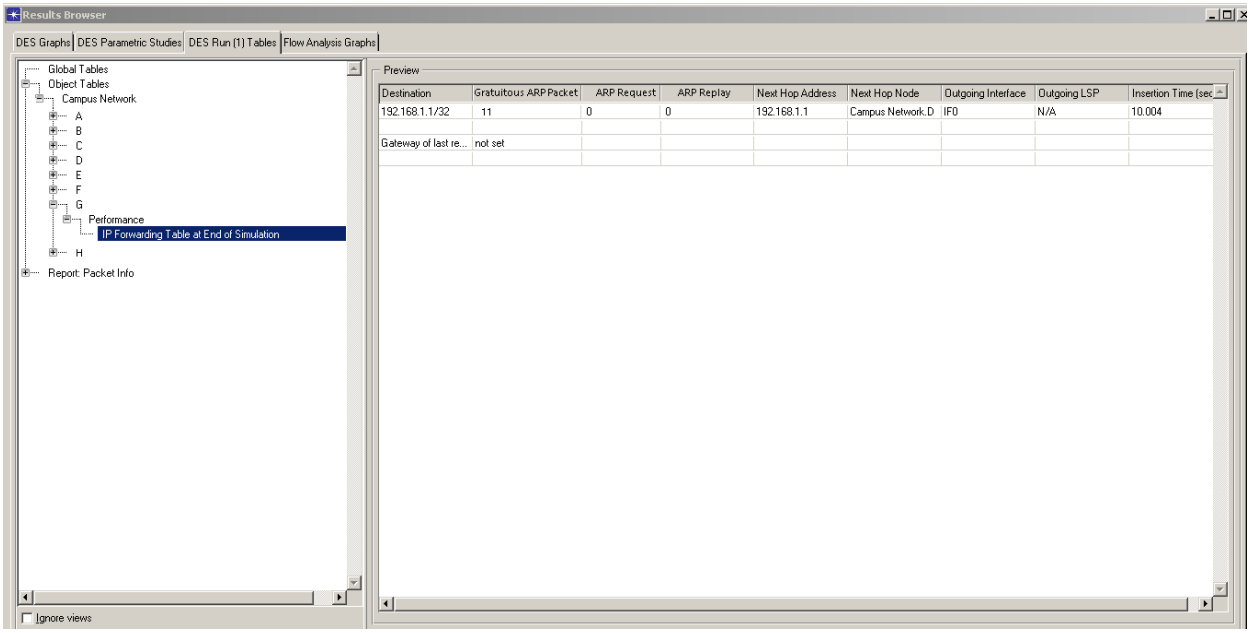


Figure 19: Node C Packet Exchange

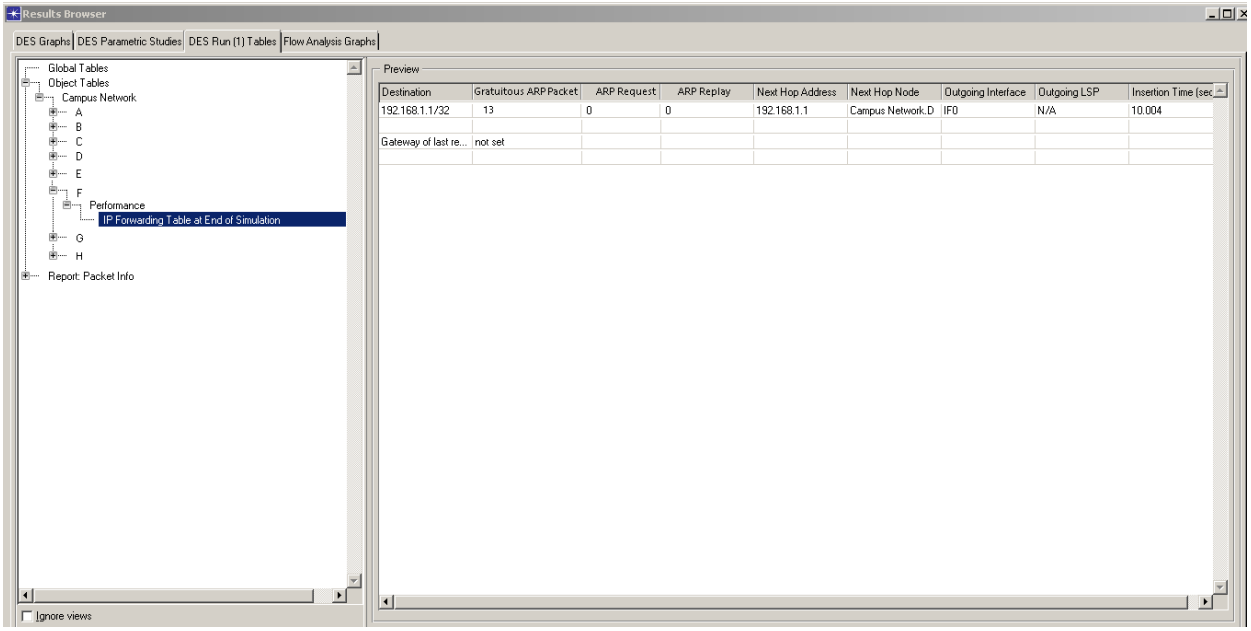


Figure 20: Node D Packet Exchange

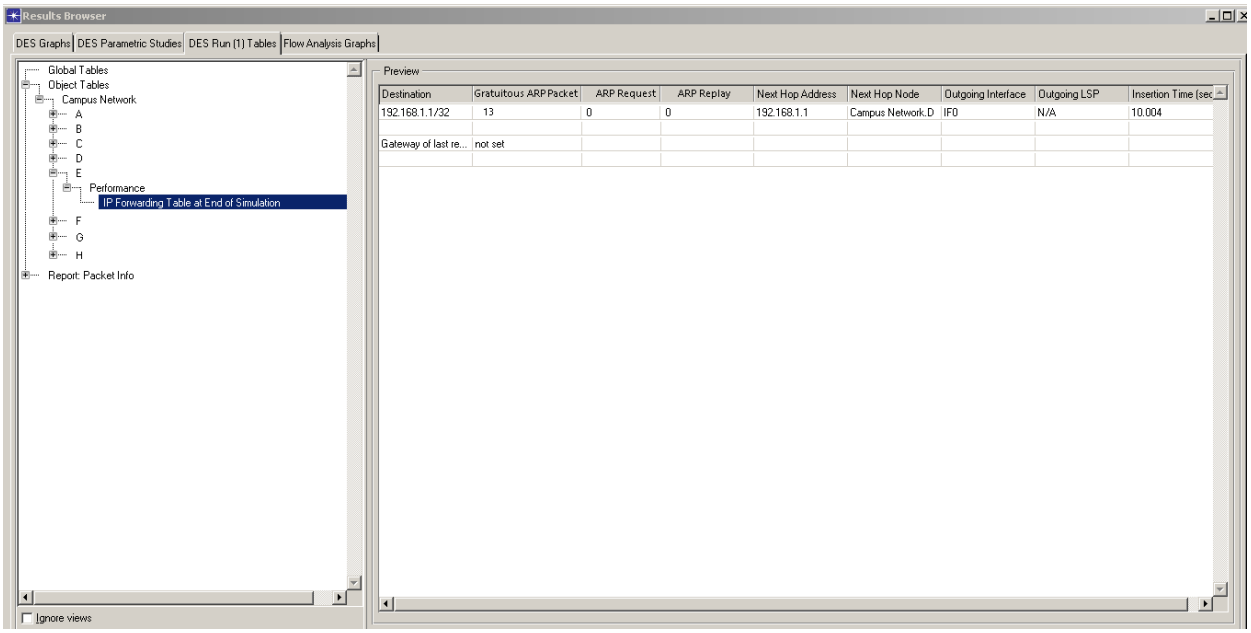


Figure 21: Node E Packet Exchange

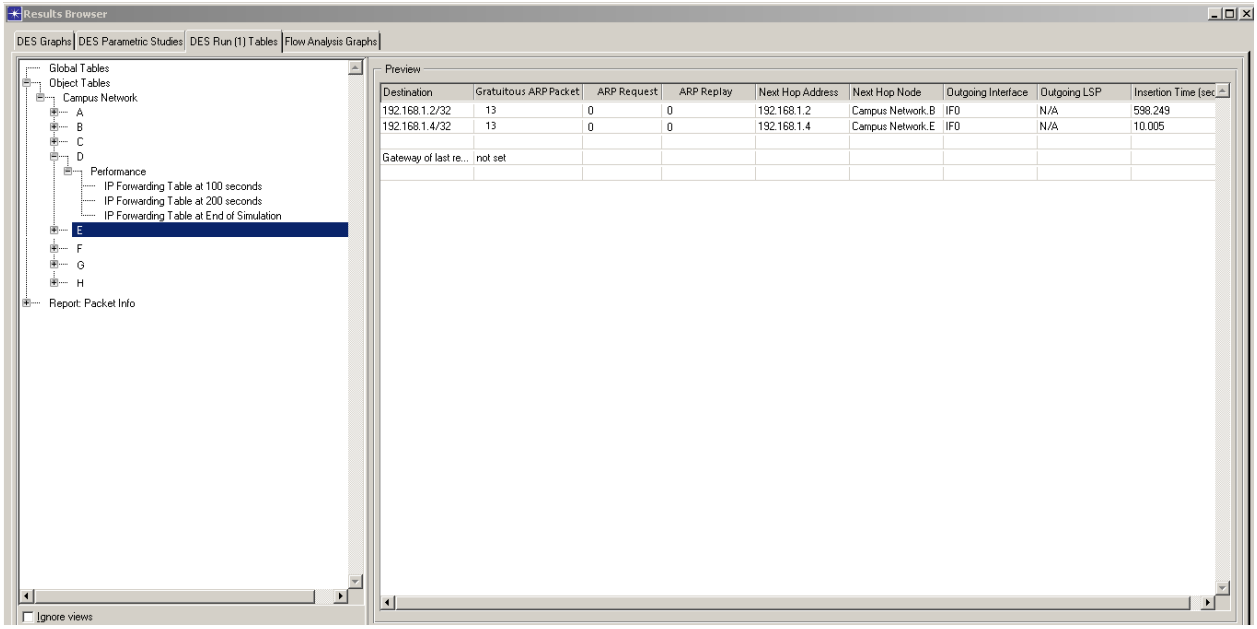


Figure 22: Node F Packet Exchange

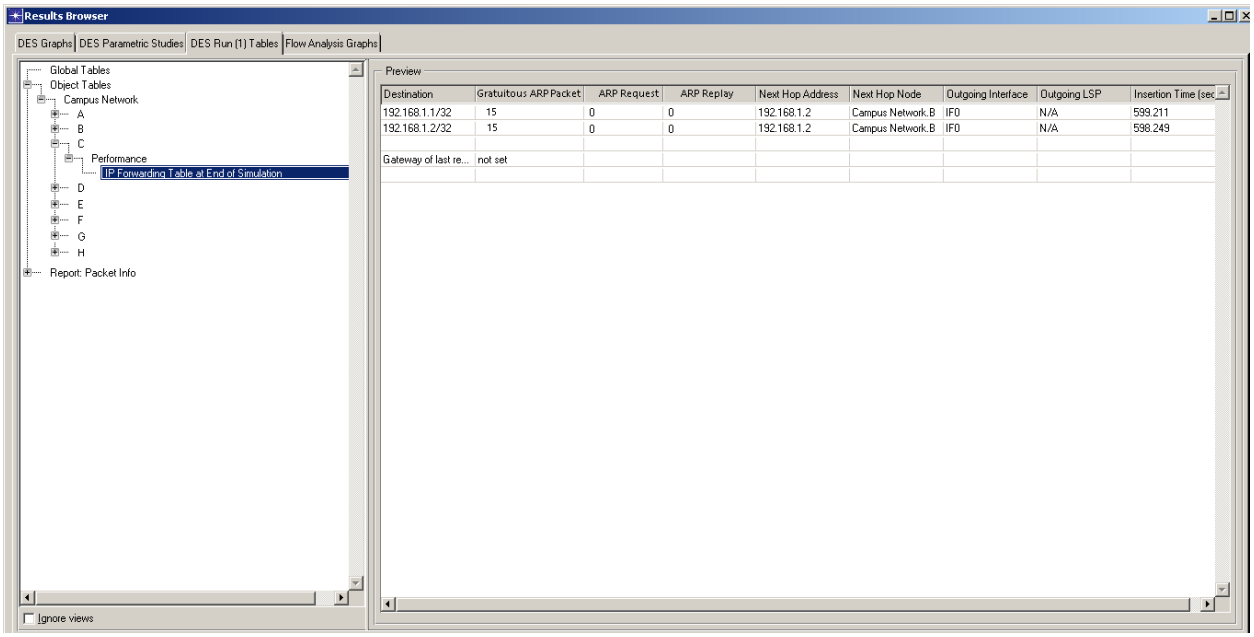


Figure 23: Node G Packet Exchange

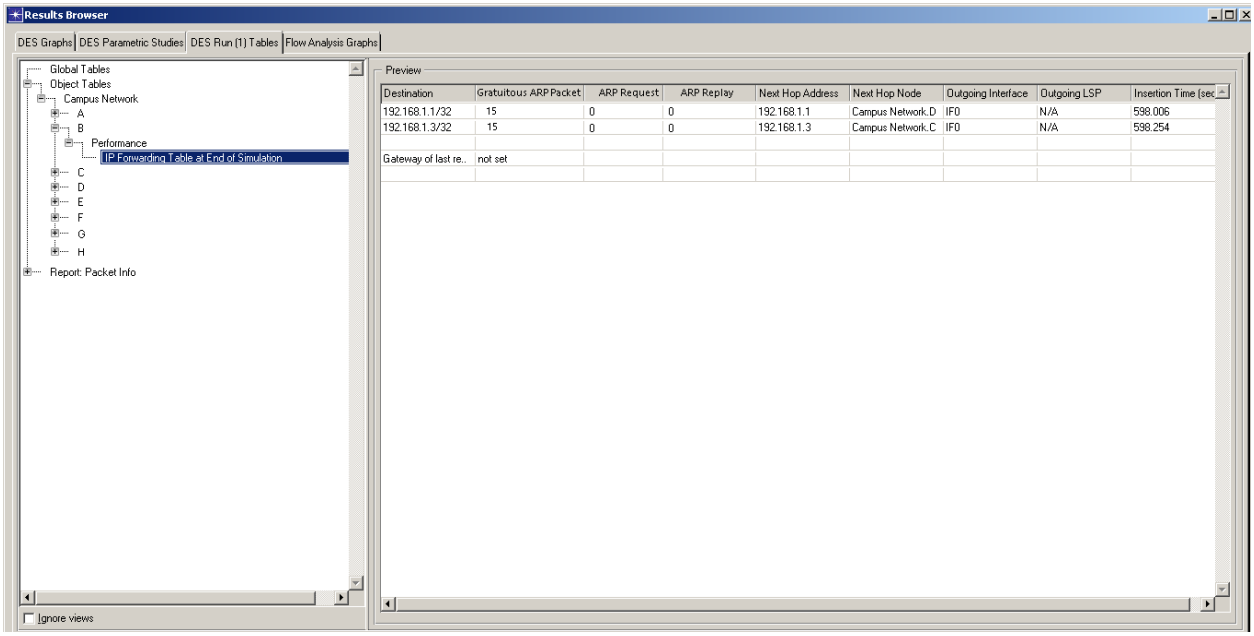


Figure 24: Node H Packet Exchange

B.2. Scenario 2 Results

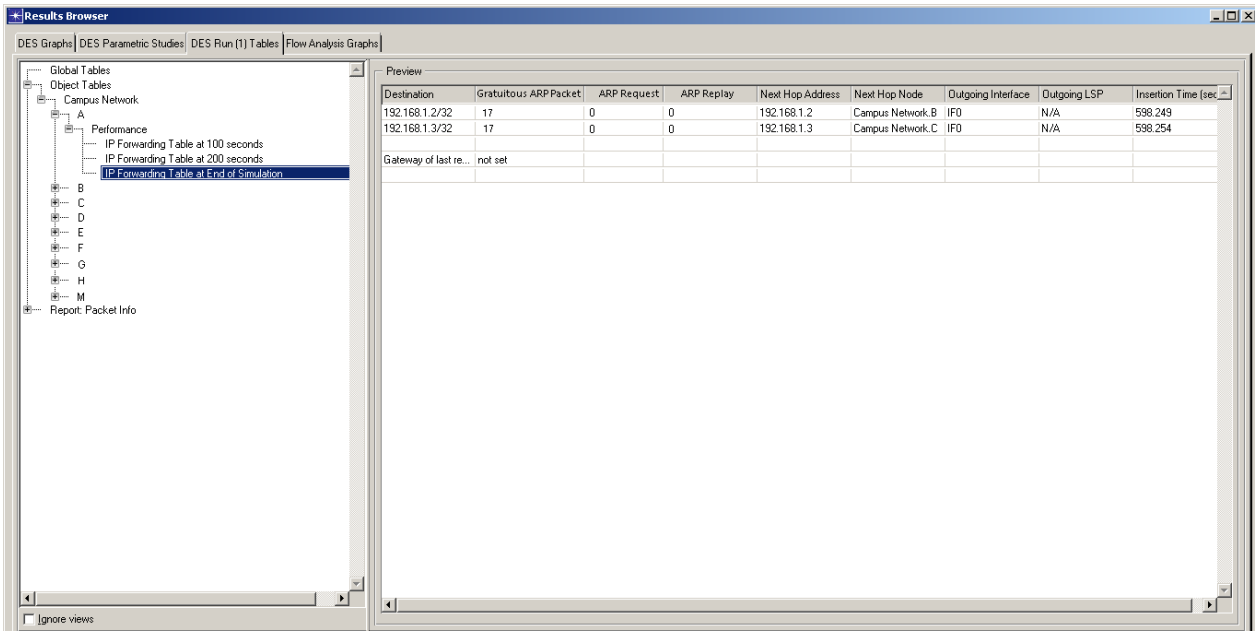


Figure 25: Node A Packet Exchange

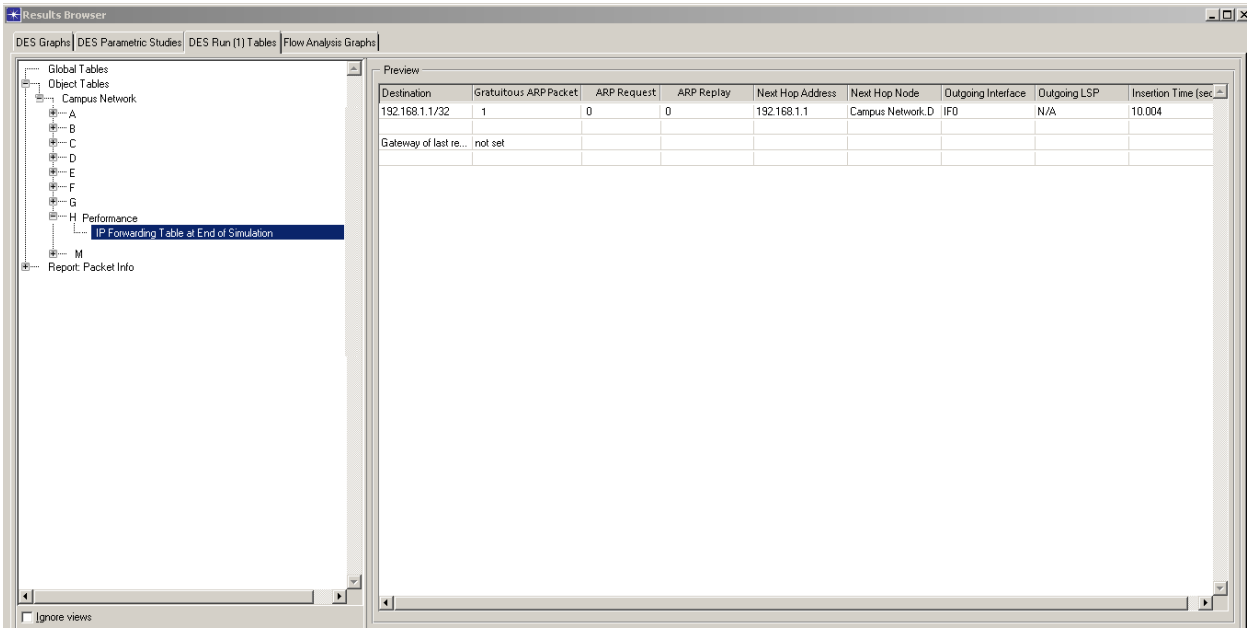


Figure 26: Node B Packet Exchange

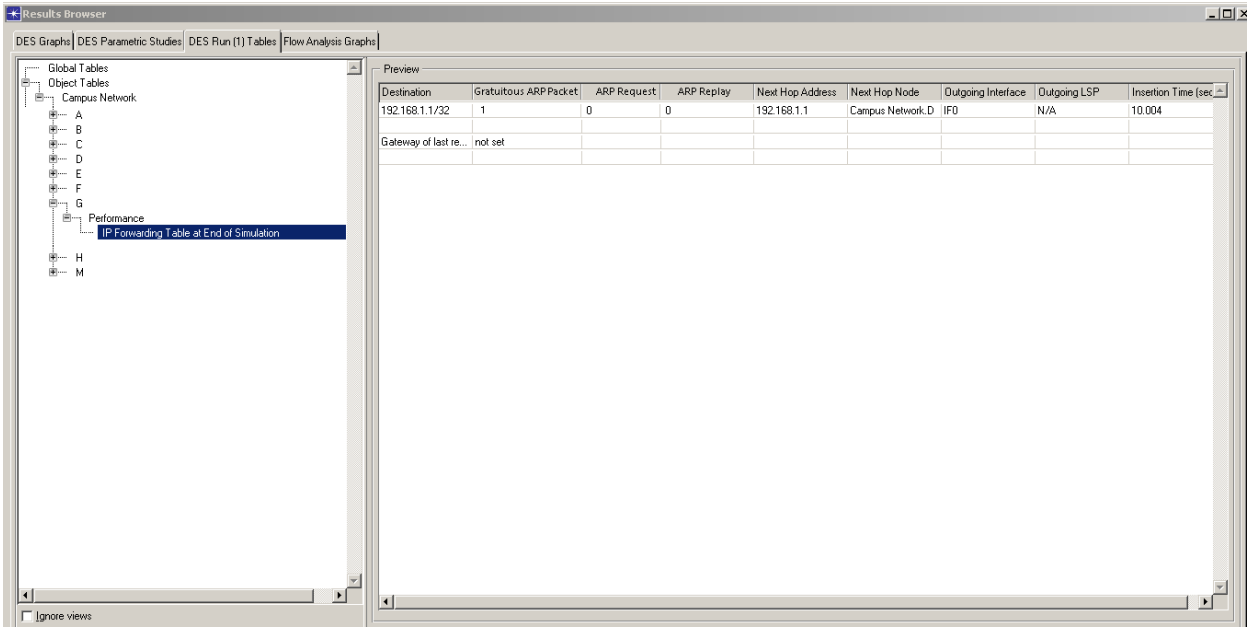


Figure 27: Node C Packet Exchange

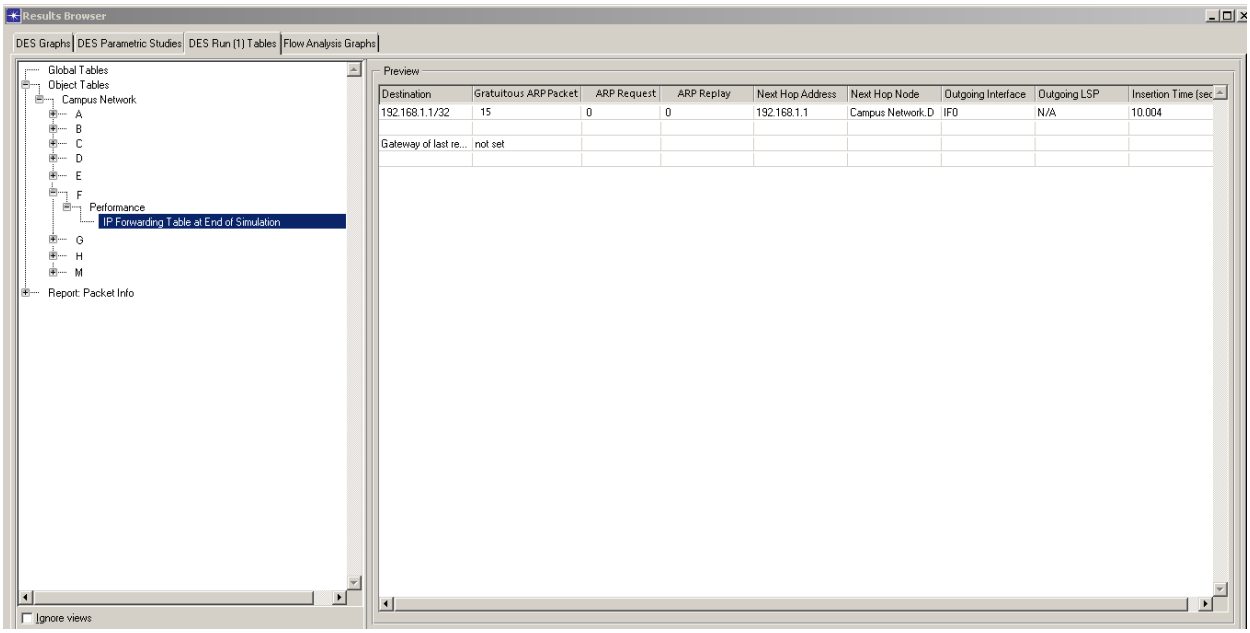


Figure 28: Node D Packet Exchange

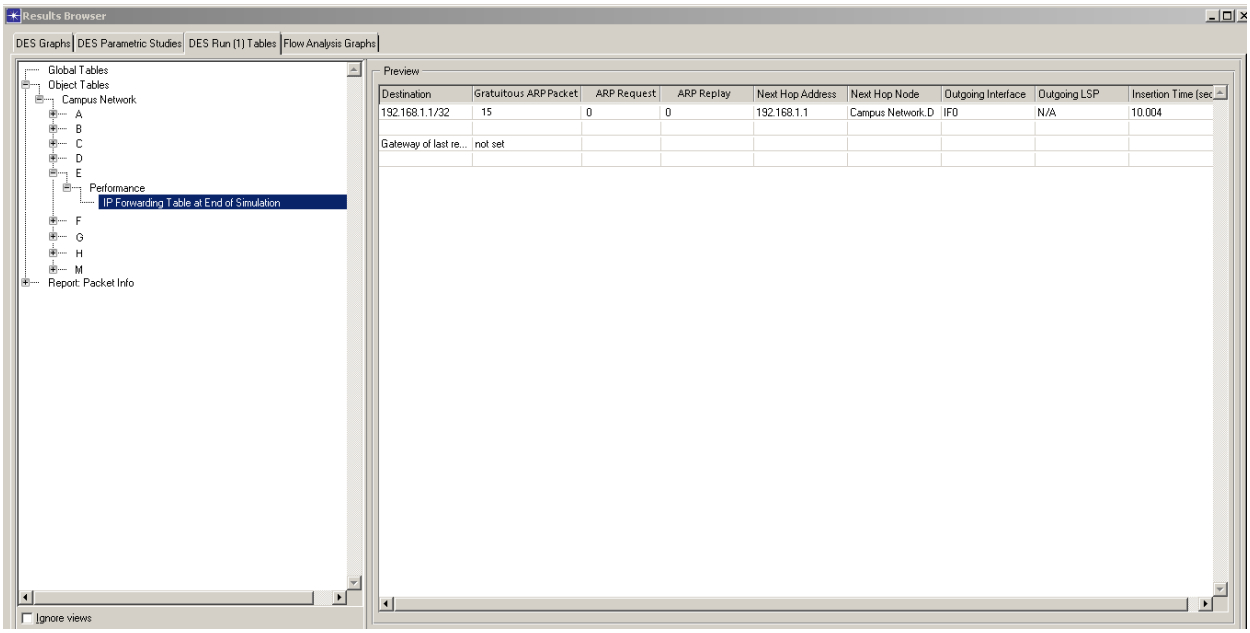


Figure 29: Node E Packet Exchange

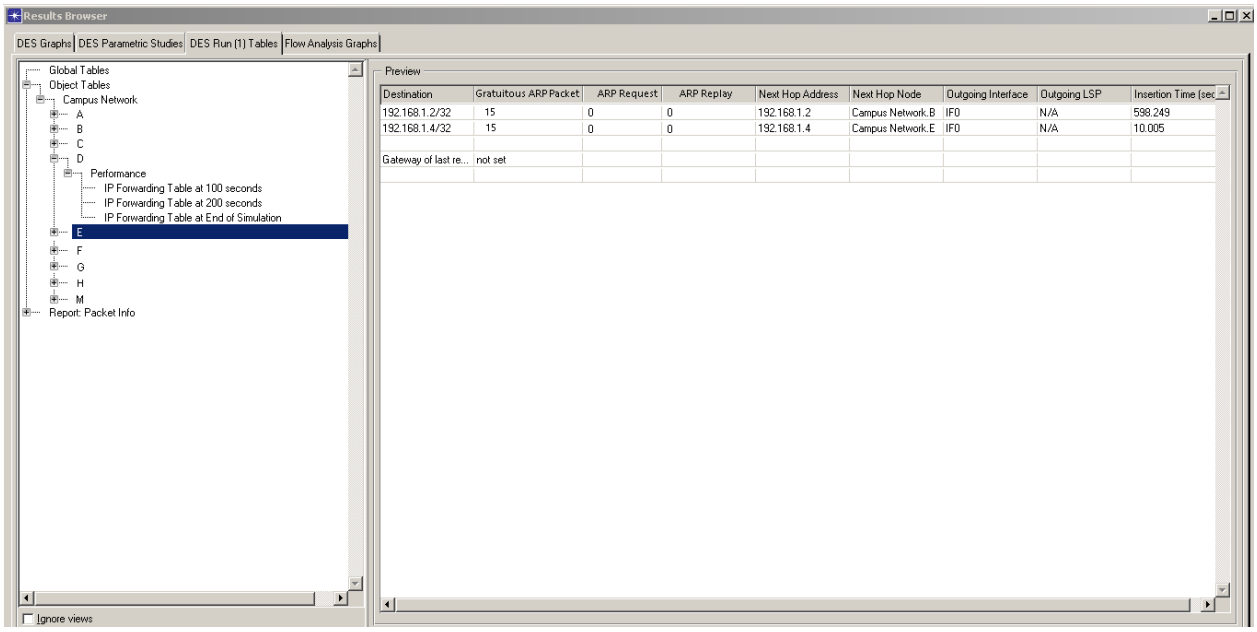


Figure 30: Node F Packet Exchange

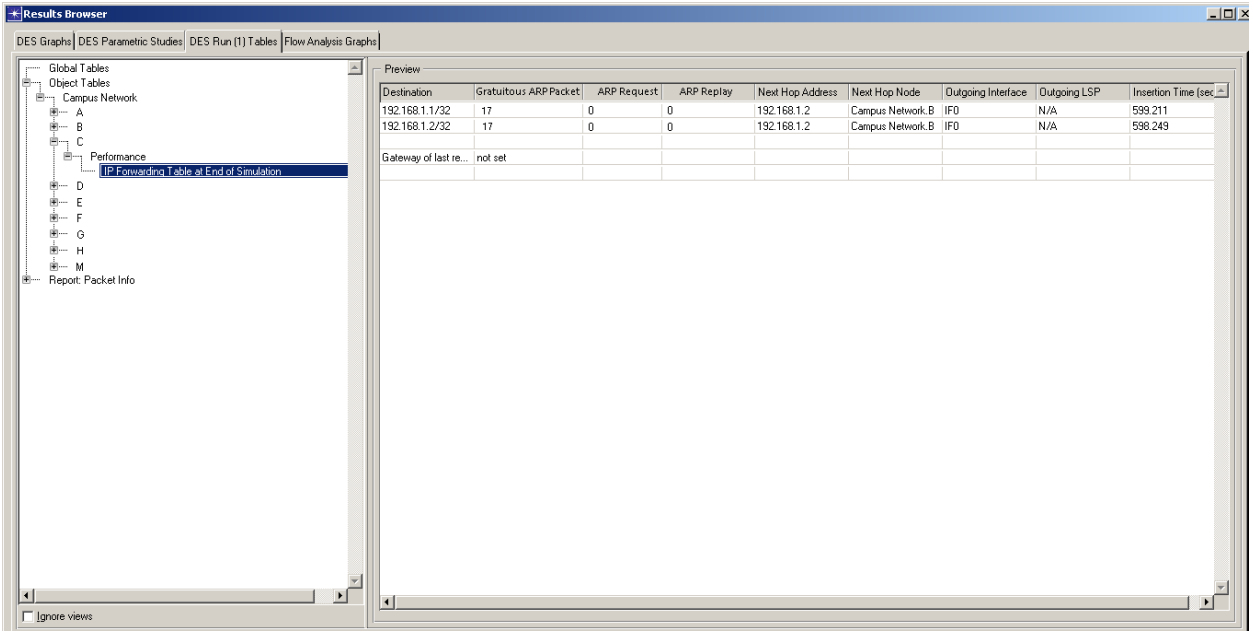


Figure 31: Node G Packet Exchange

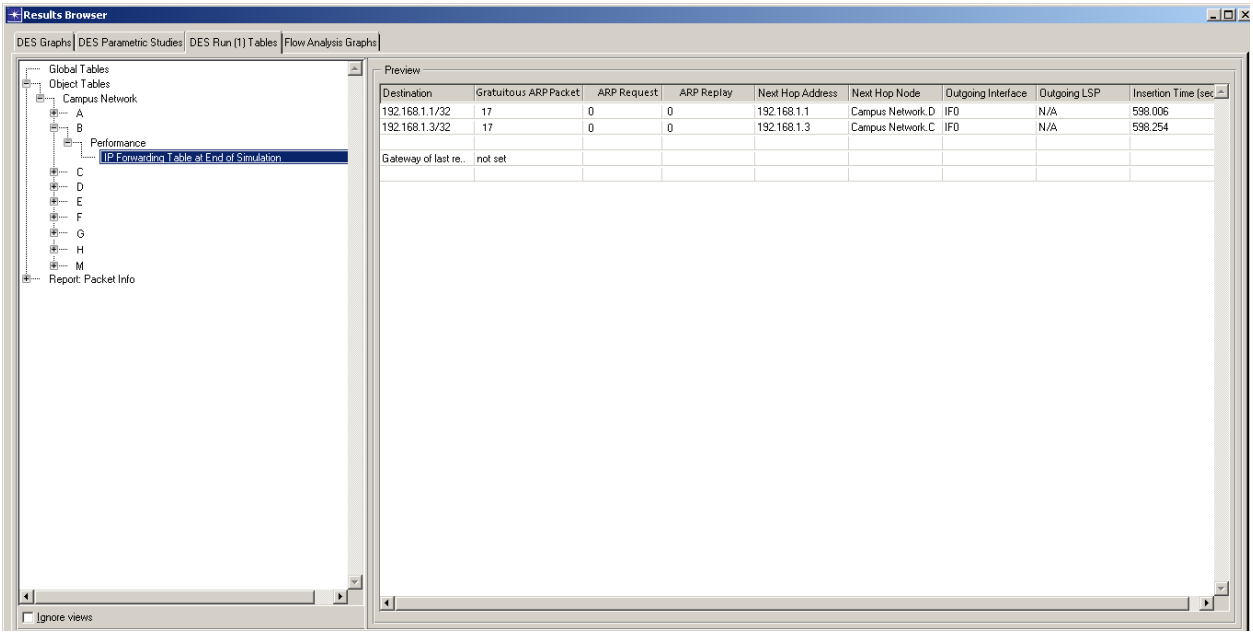


Figure 32: Node H Packet Exchange

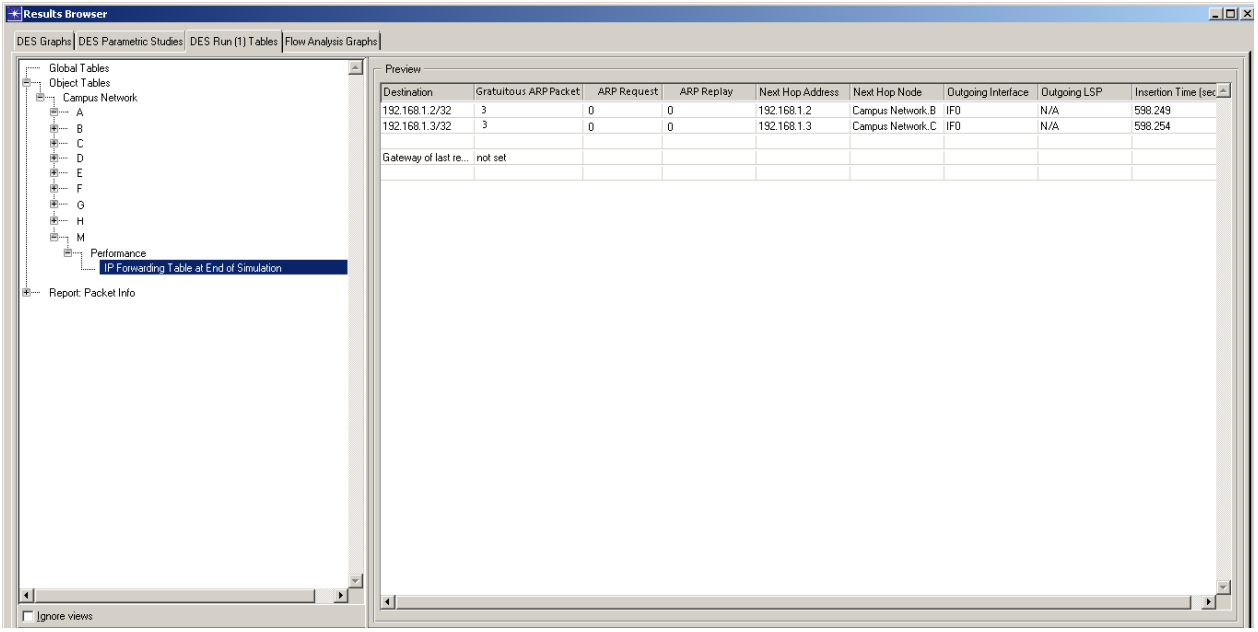


Figure 33: Node M Packet Exchange

B.3. Scenario 3 Results

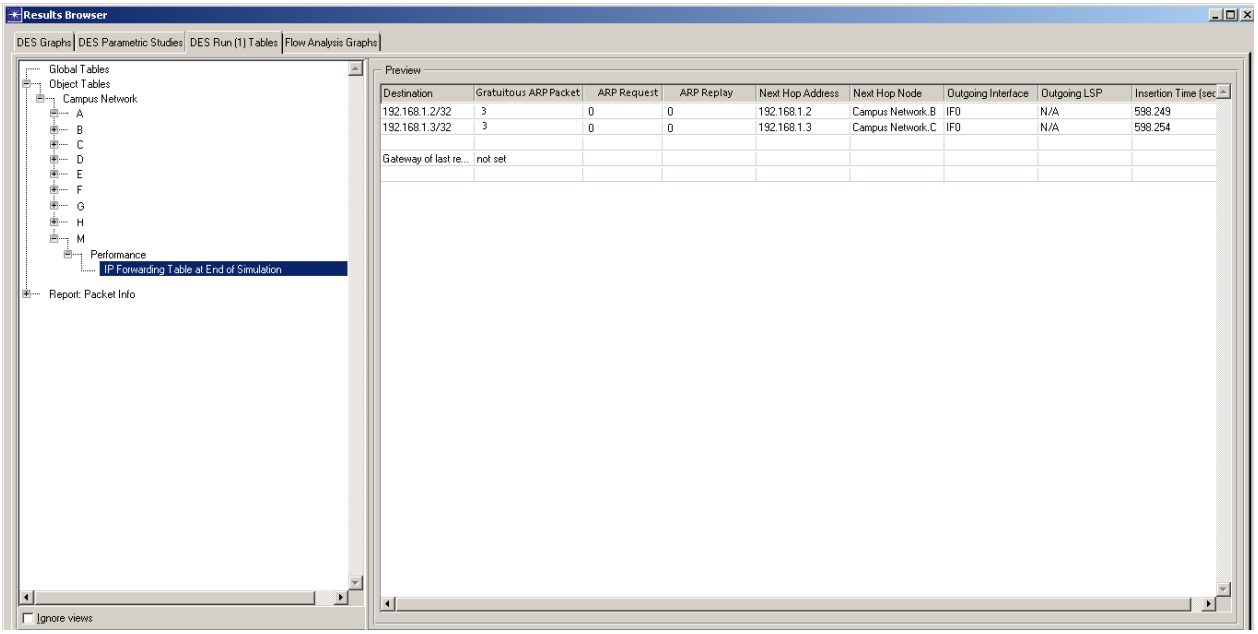


Figure 34: Node A Packet Exchange

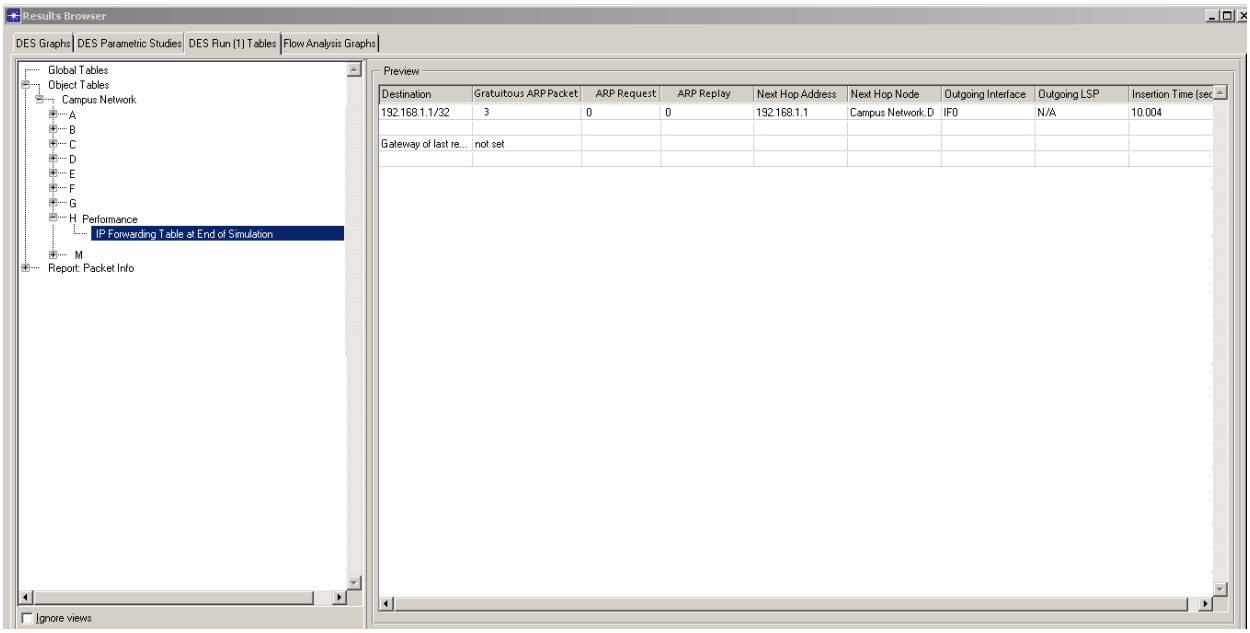


Figure 35: Node B Packet Exchange

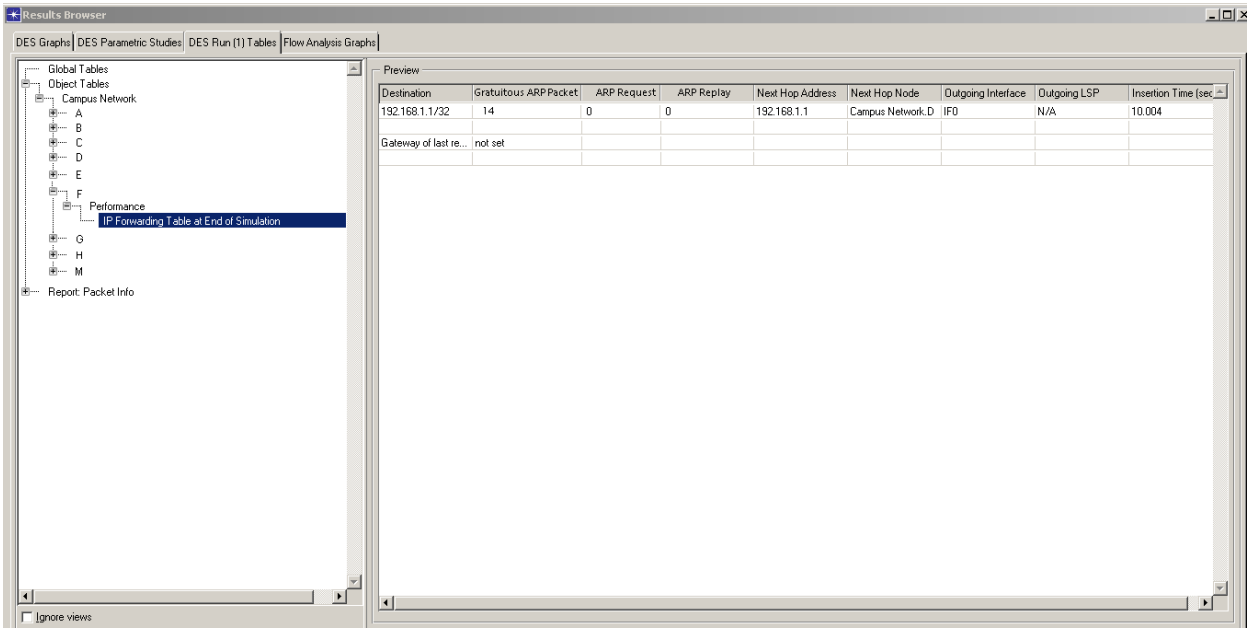


Figure 36: Node C Packet Exchange

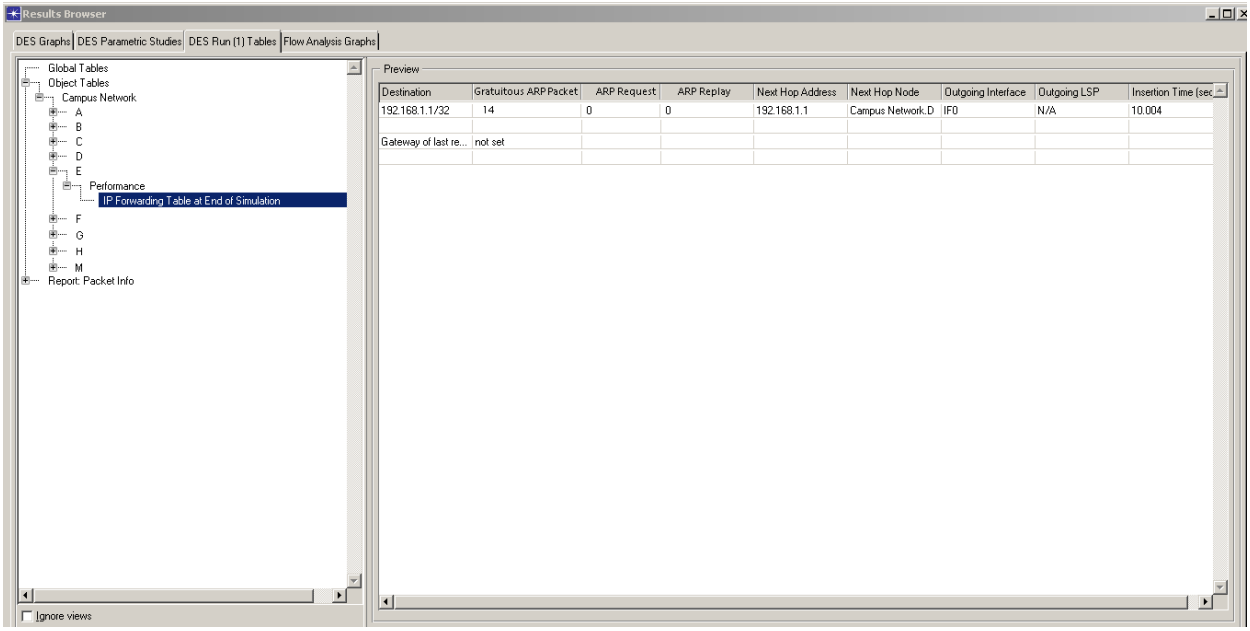


Figure 37: Node D Packet Exchange

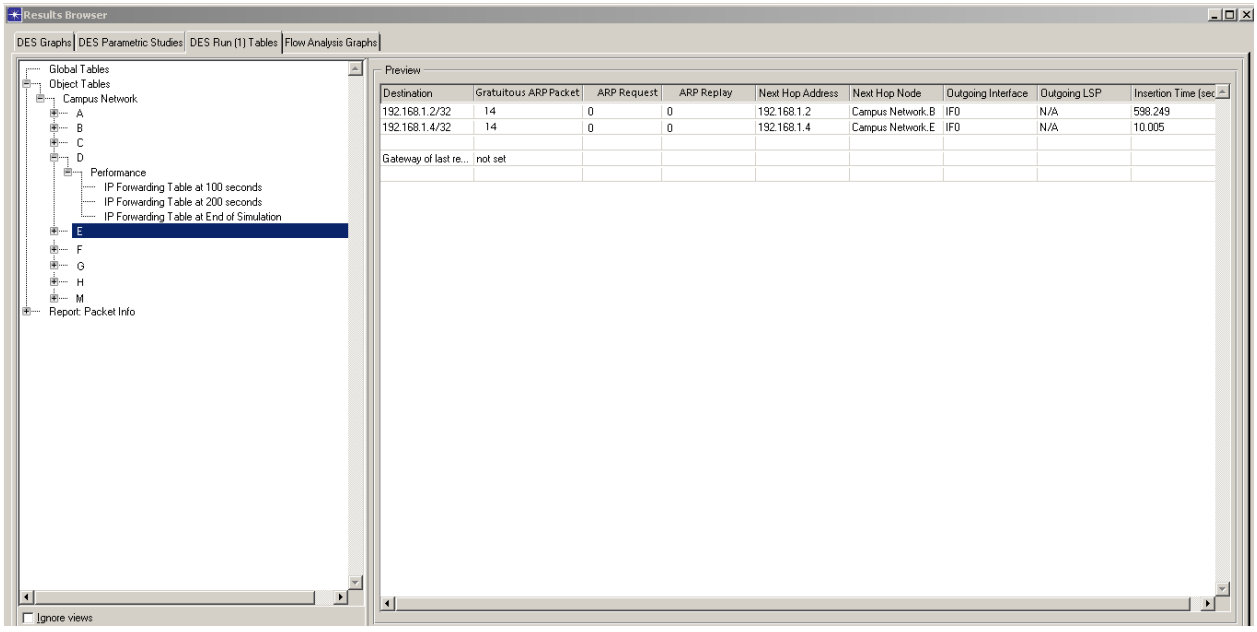


Figure 38: Node E Packet Exchange

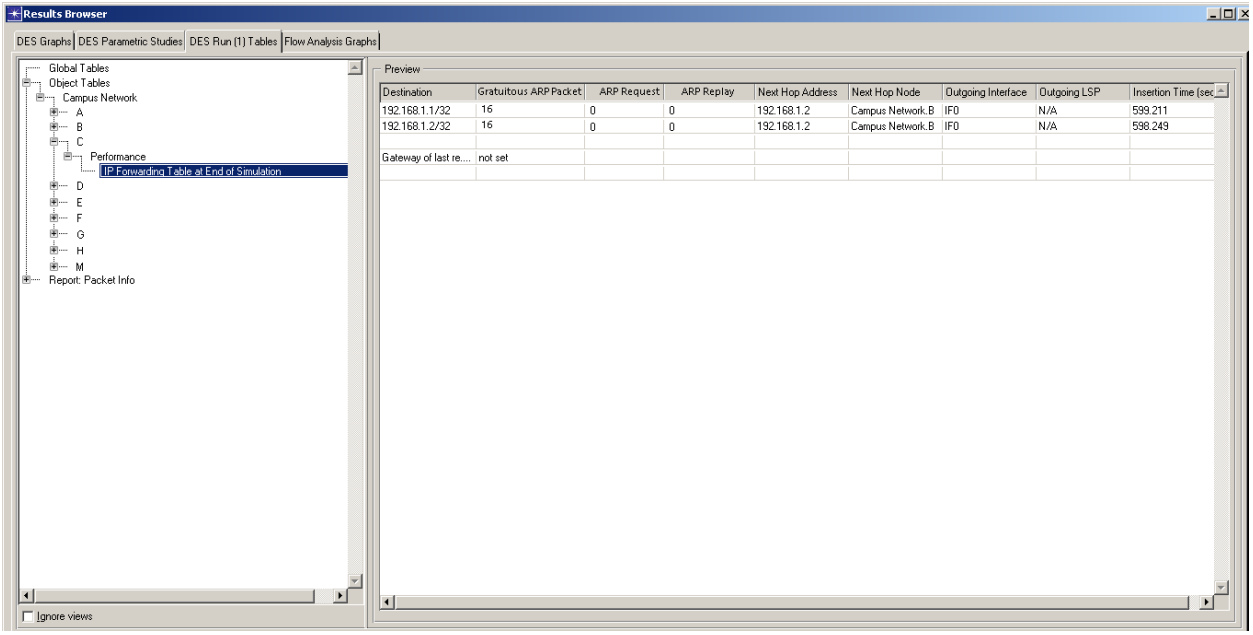


Figure 39: Node F Packet Exchange

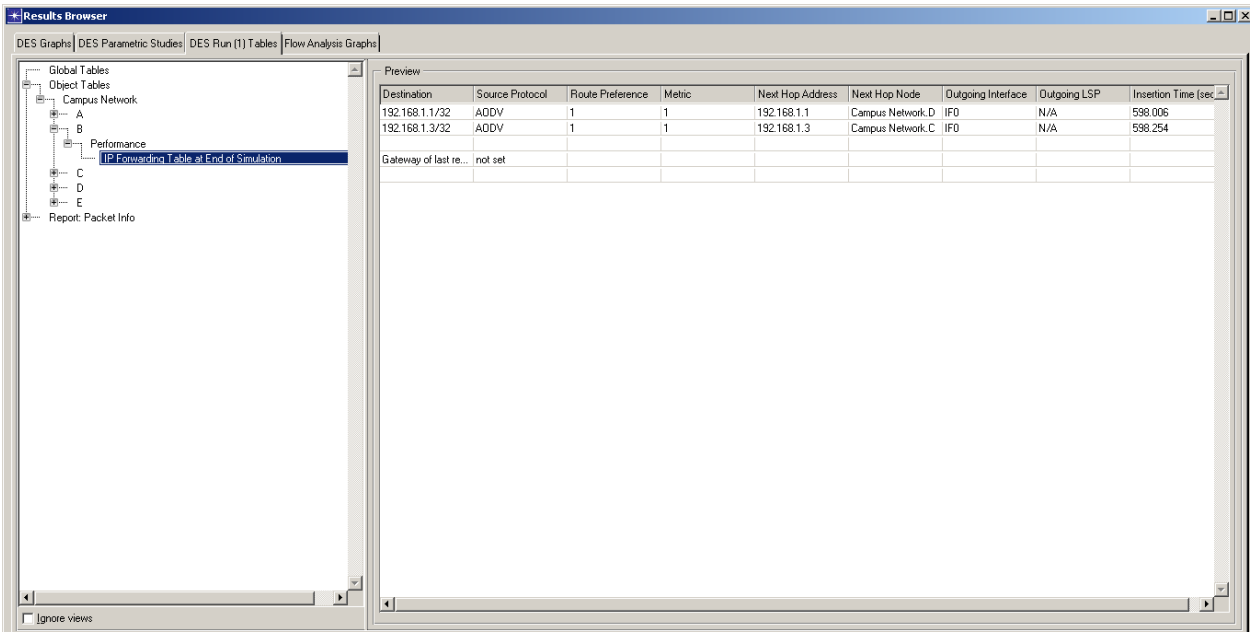


Figure 40: Node G Packet Exchange

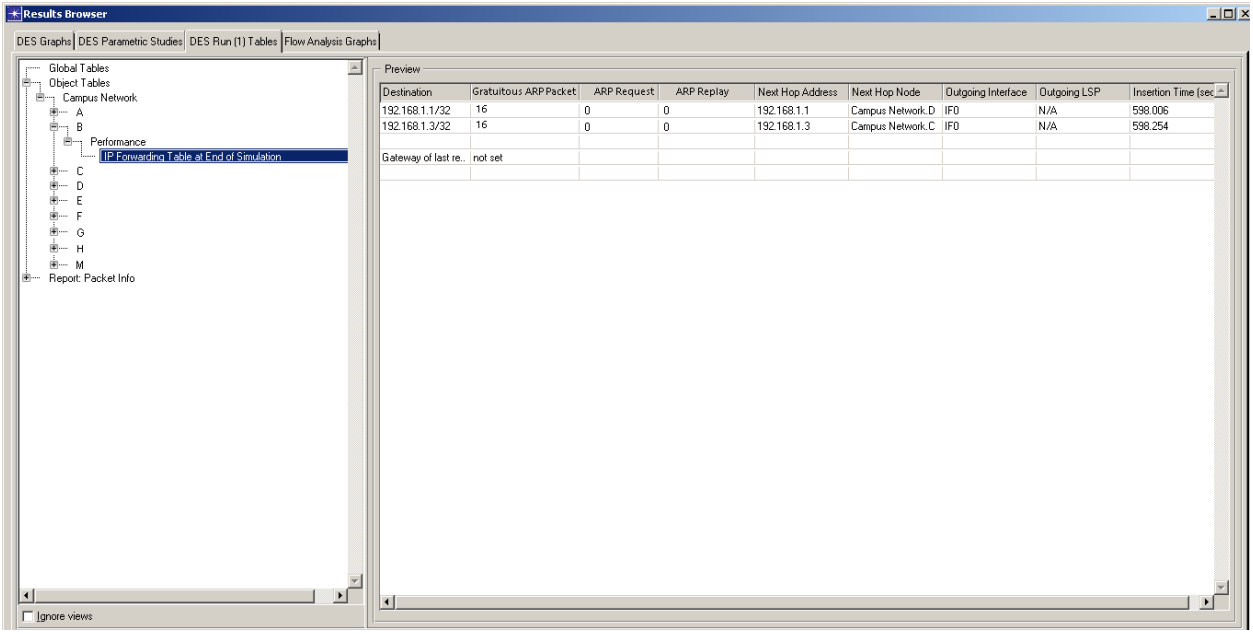


Figure 41: Node H Packet Exchange

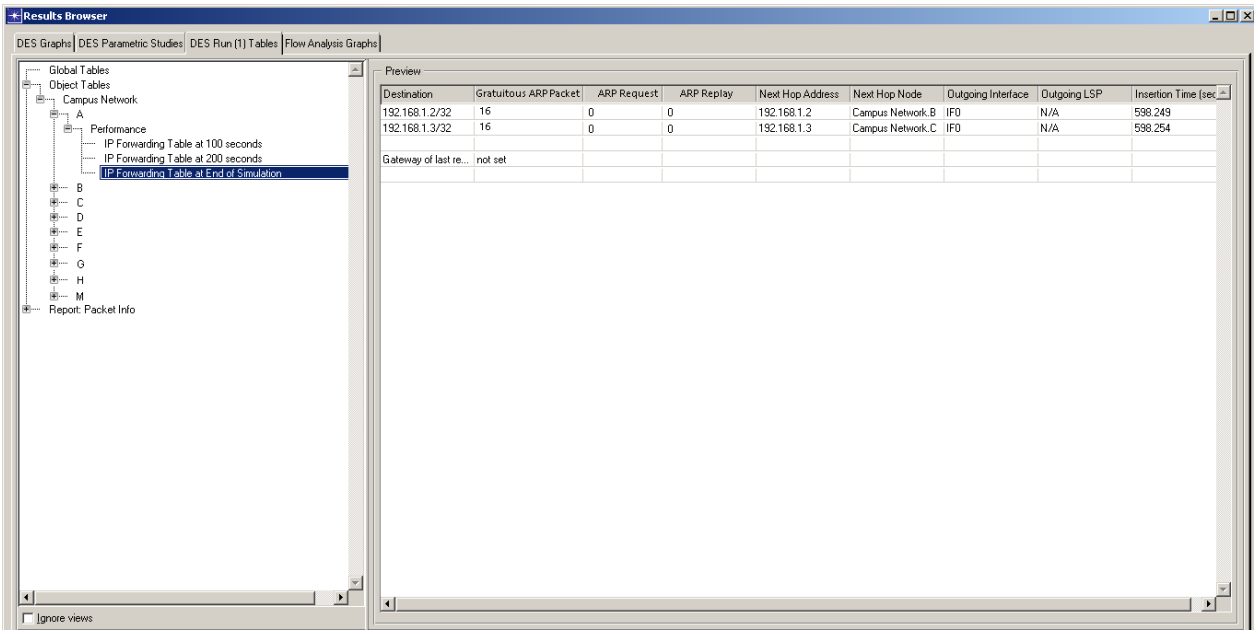


Figure 42: Node M Packet Exchange

B.4. Scenario 4 Results

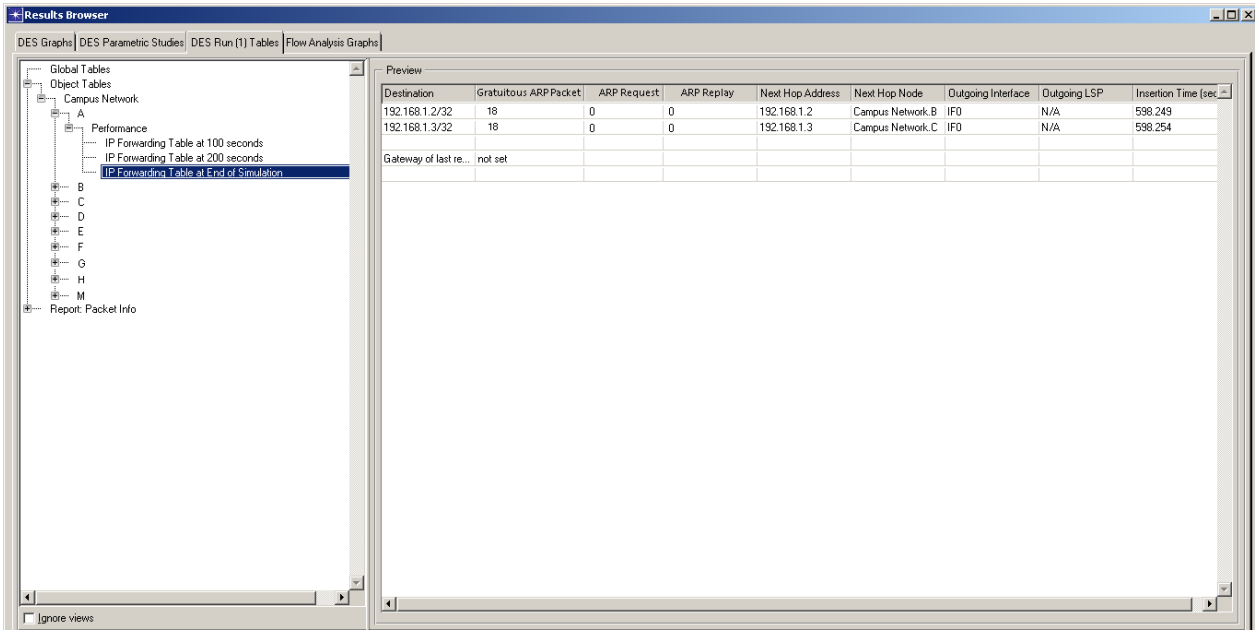


Figure 43: Node A Packet Exchange

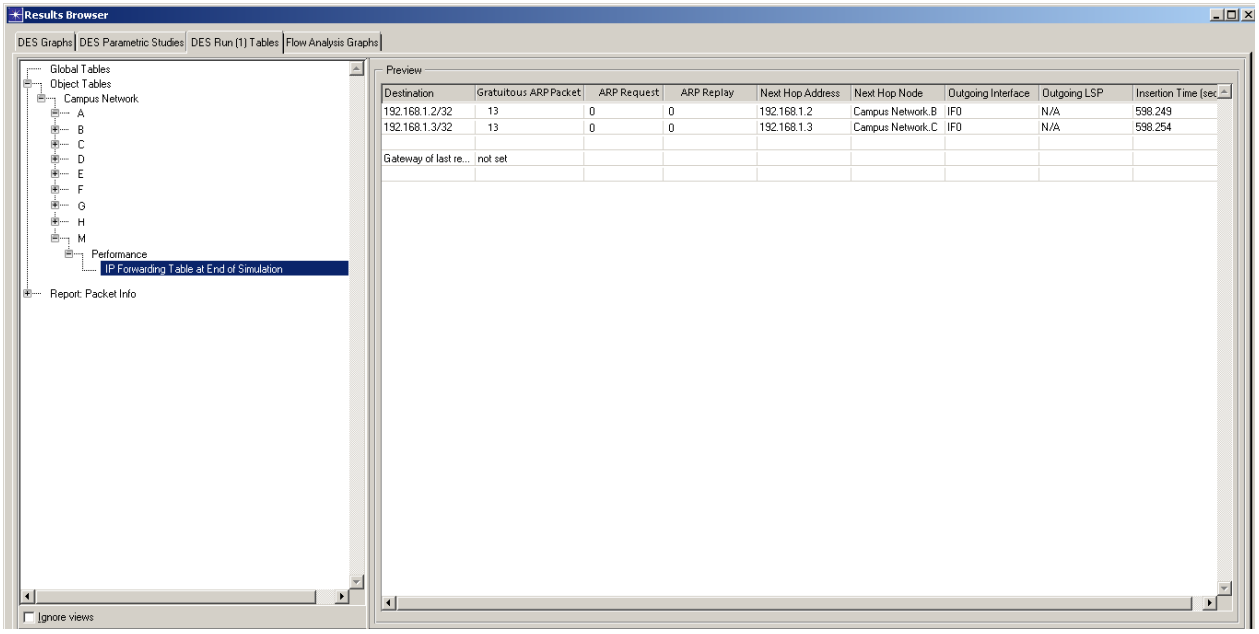


Figure 44: Node B Packet Exchange

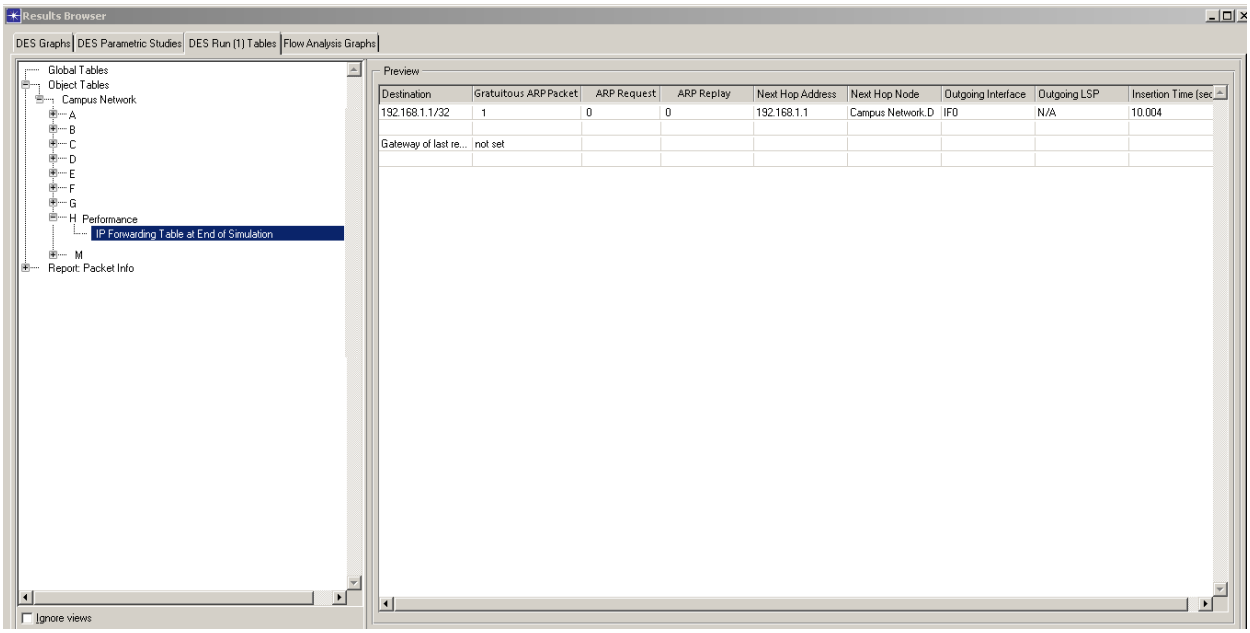


Figure 45: Node C Packet Exchange

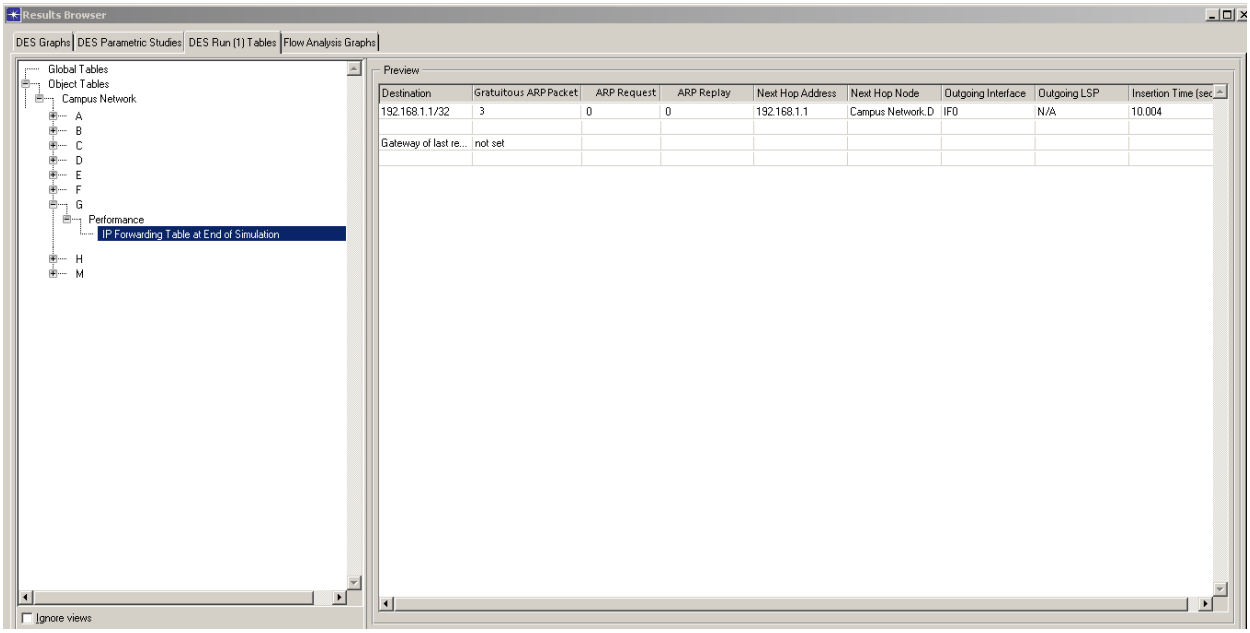


Figure 46: Node D Packet Exchange

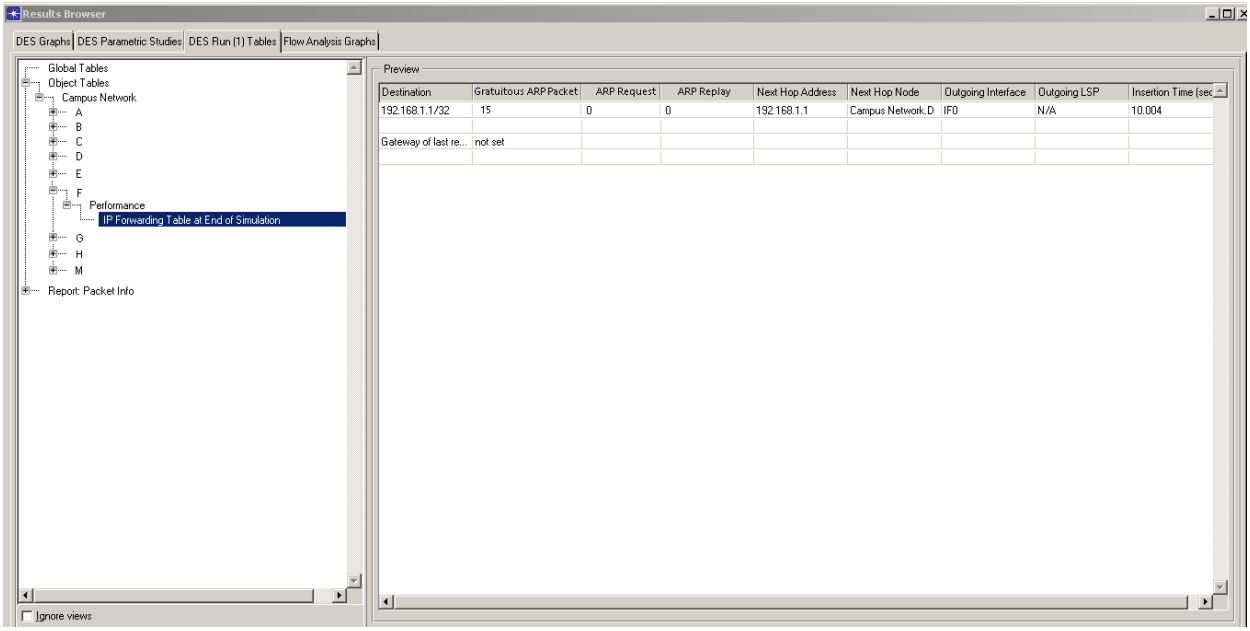


Figure 47: Node E Packet Exchange

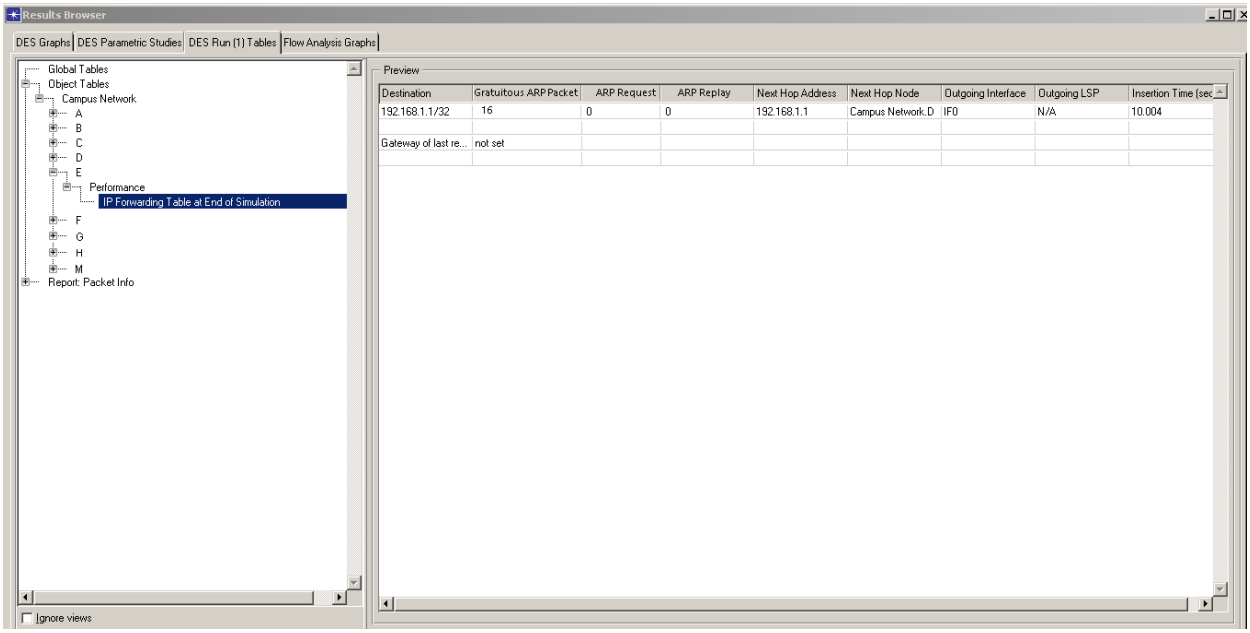


Figure 48: Node F Packet Exchange

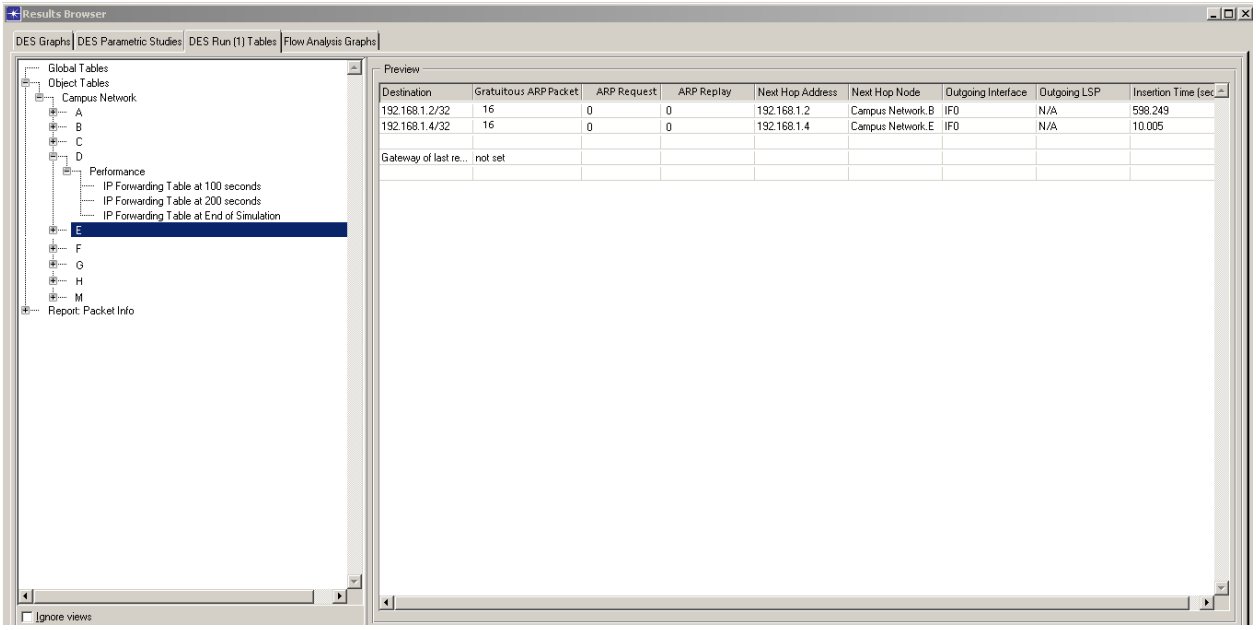


Figure 49: Node G Packet Exchange

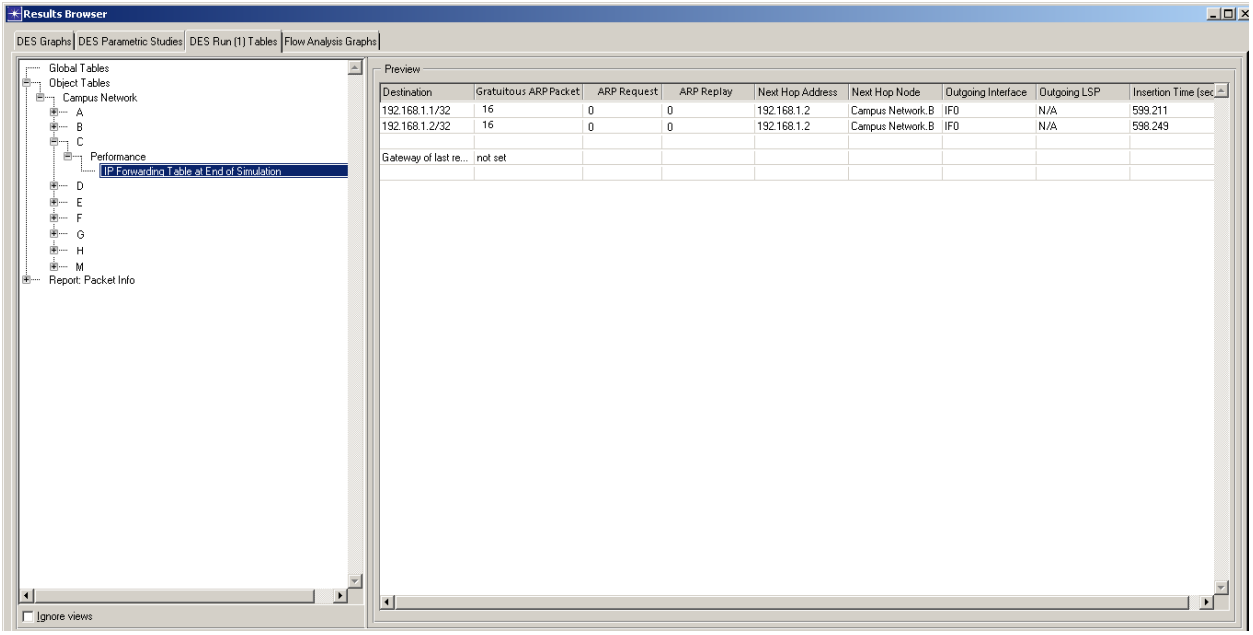


Figure 50: Node H Packet Exchange

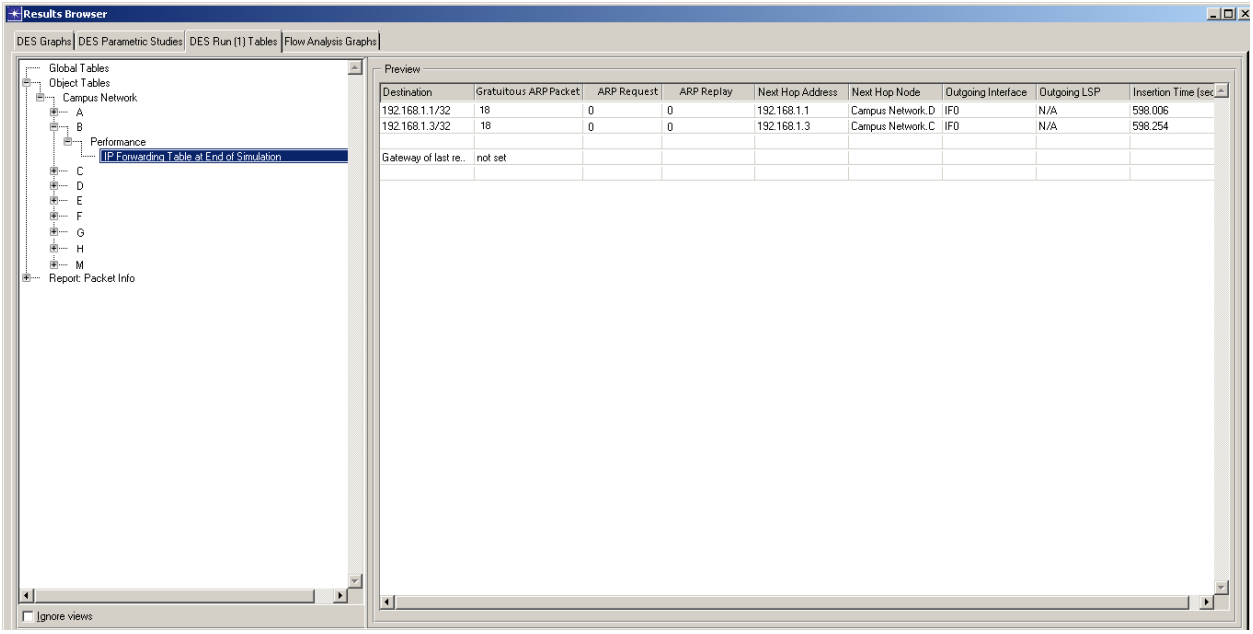


Figure 51: Node M Packet Exchange

B.5. Scenario 5 Results

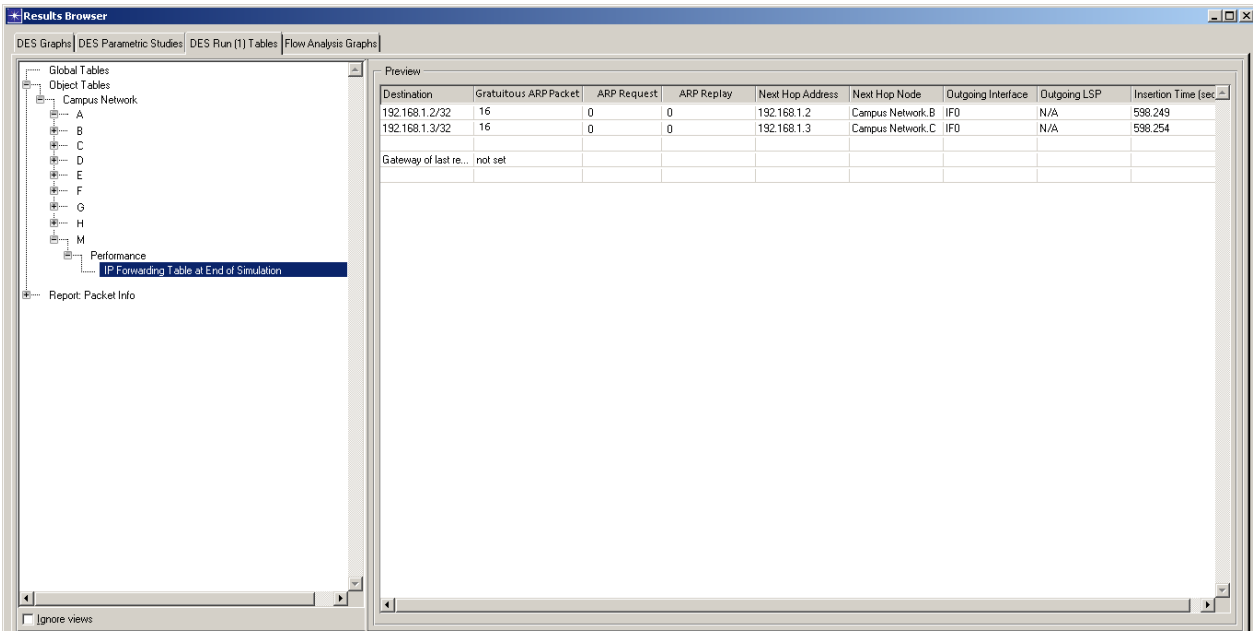


Figure 52: Node A Packet Exchange

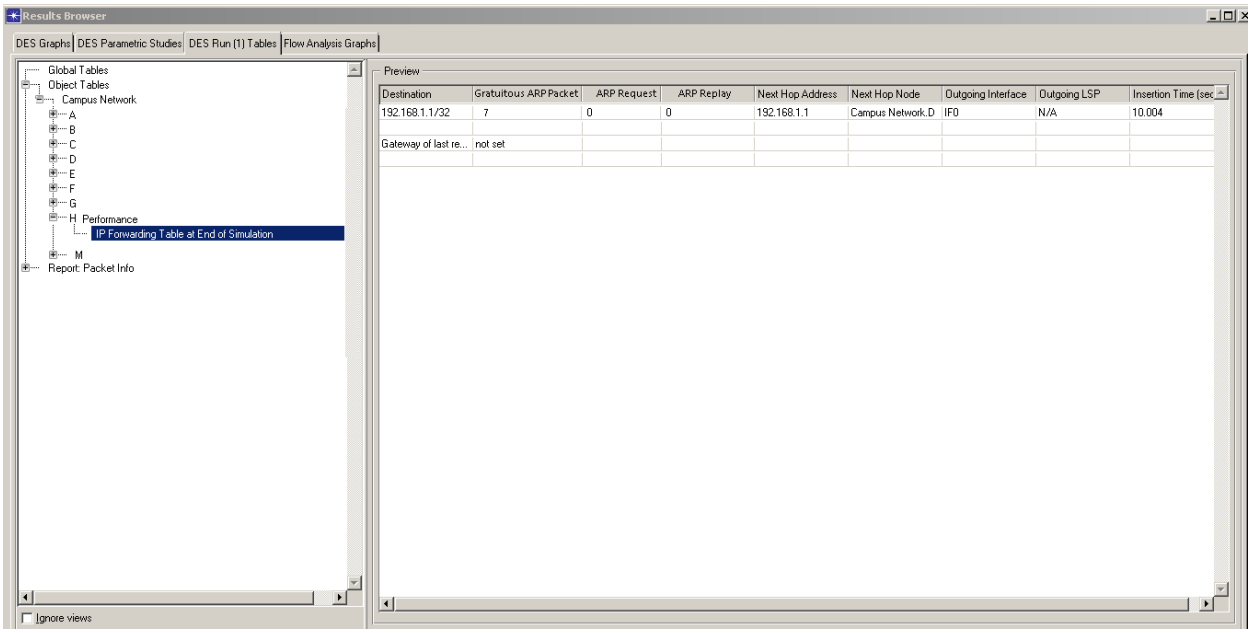


Figure 53: Node B Packet Exchange

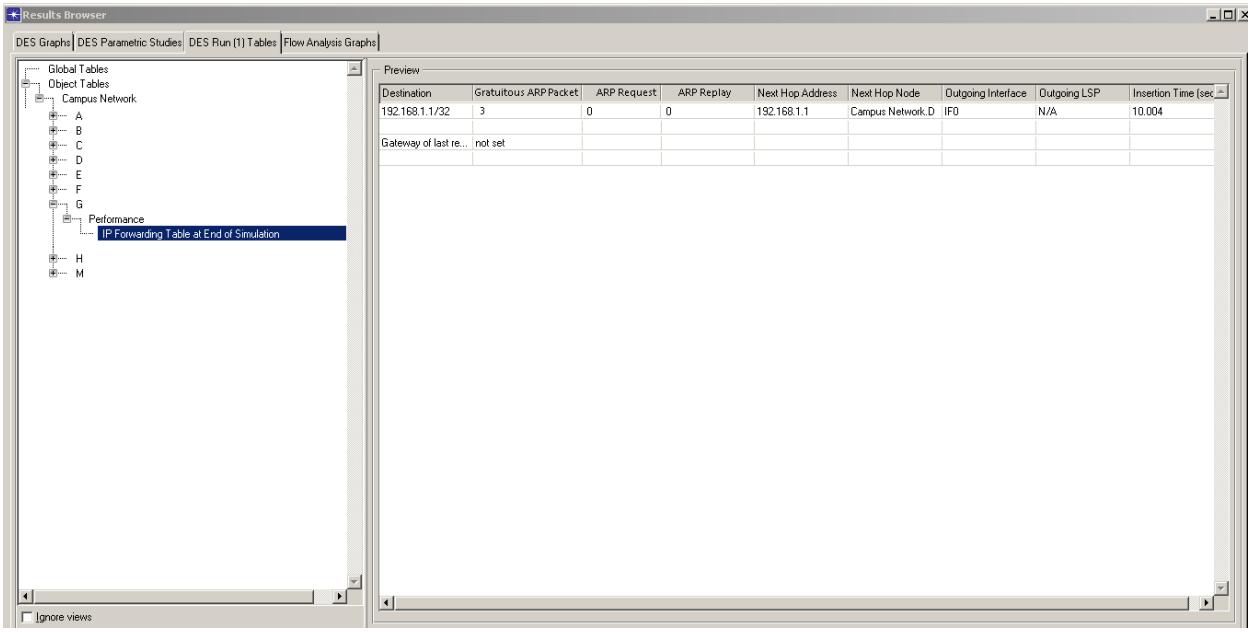


Figure 54: Node C Packet Exchange

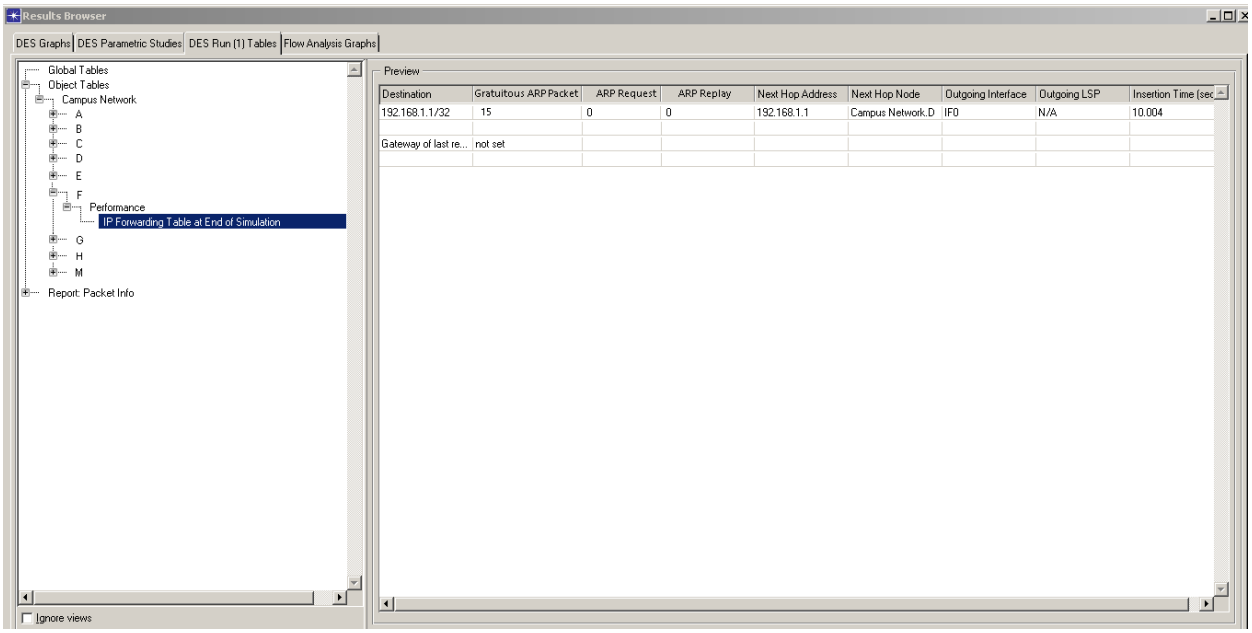


Figure 55: Node D Packet Exchange

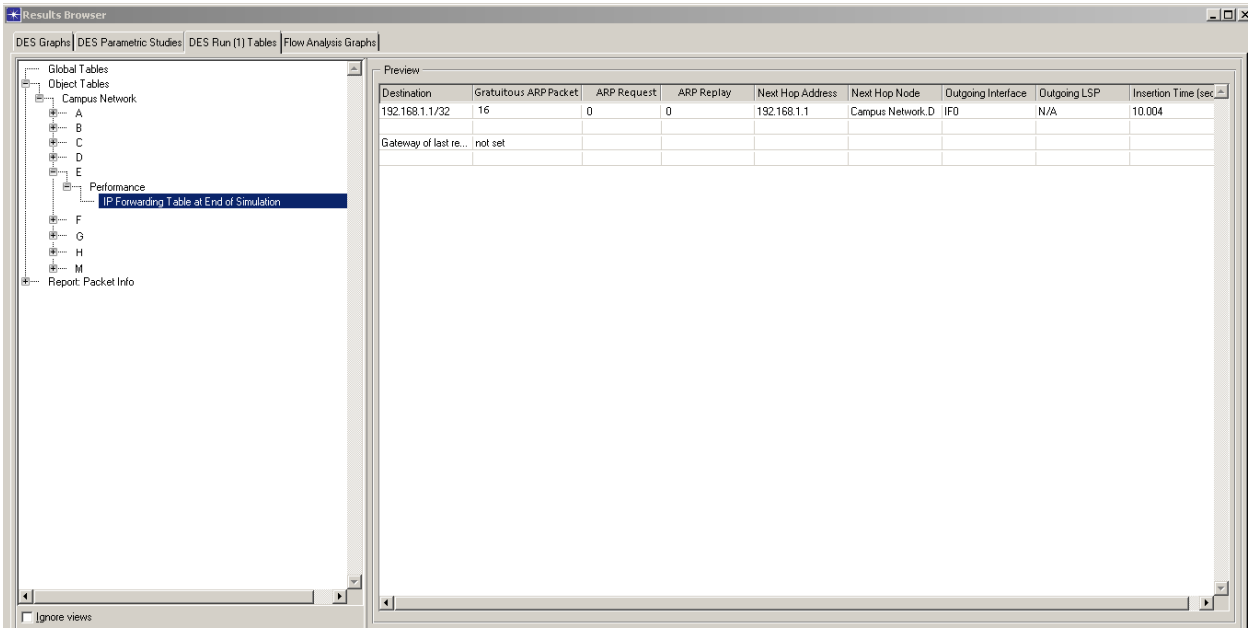


Figure 56: Node E Packet Exchange

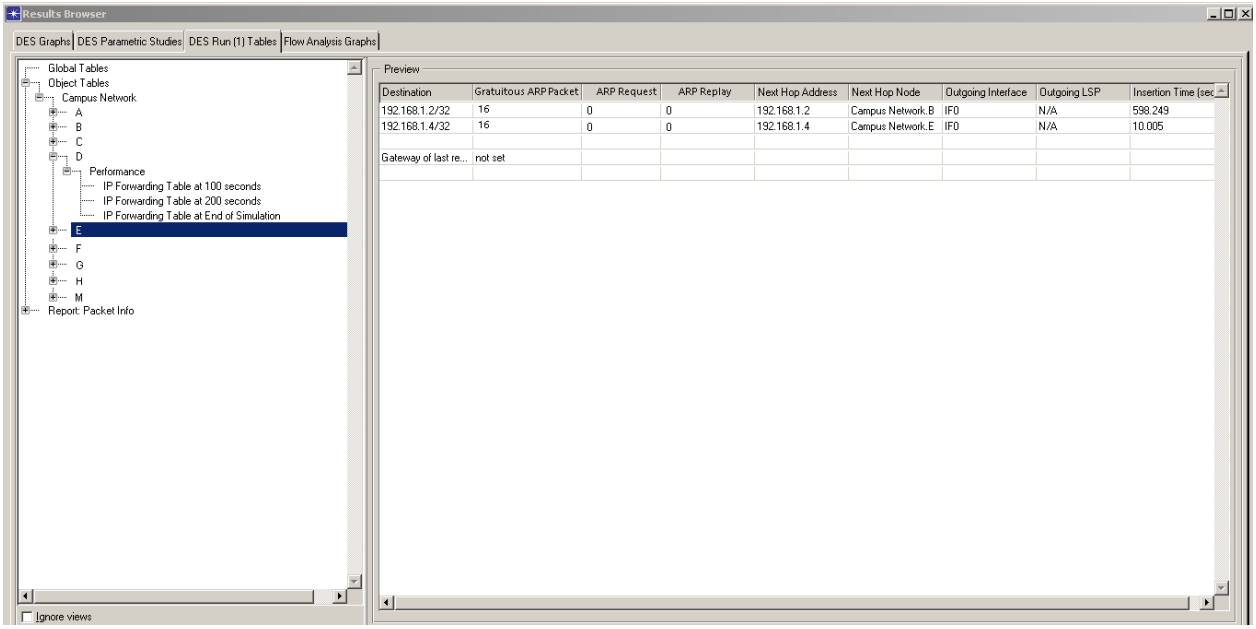


Figure 57: Node F Packet Exchange

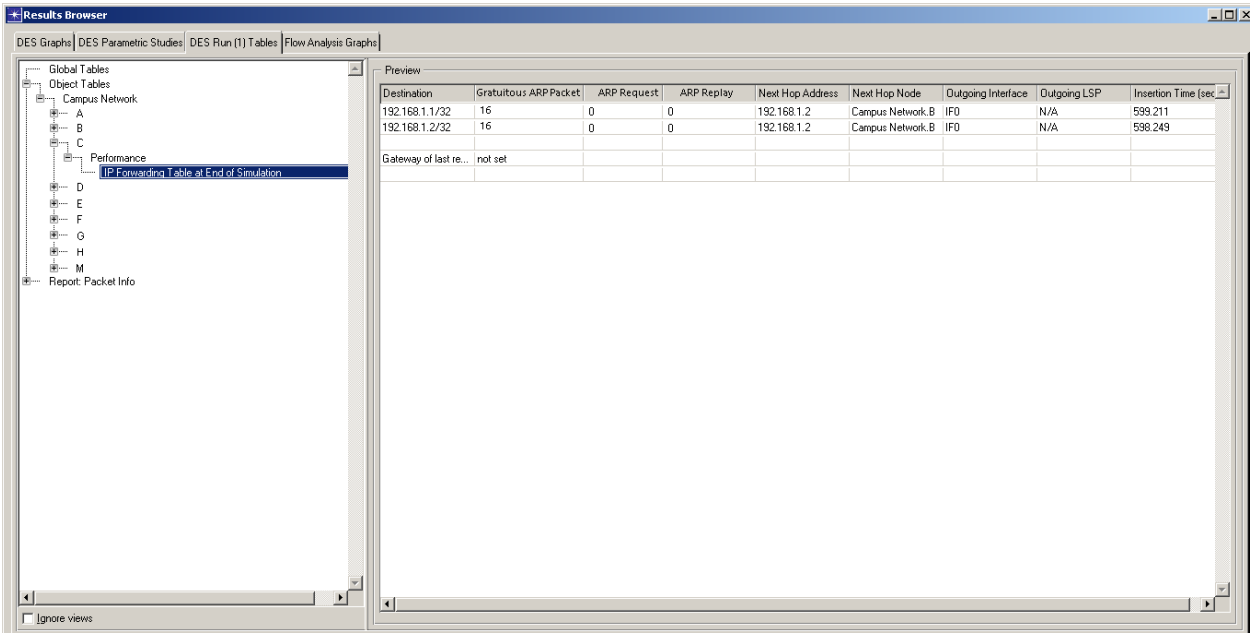


Figure 58: Node G Packet Exchange

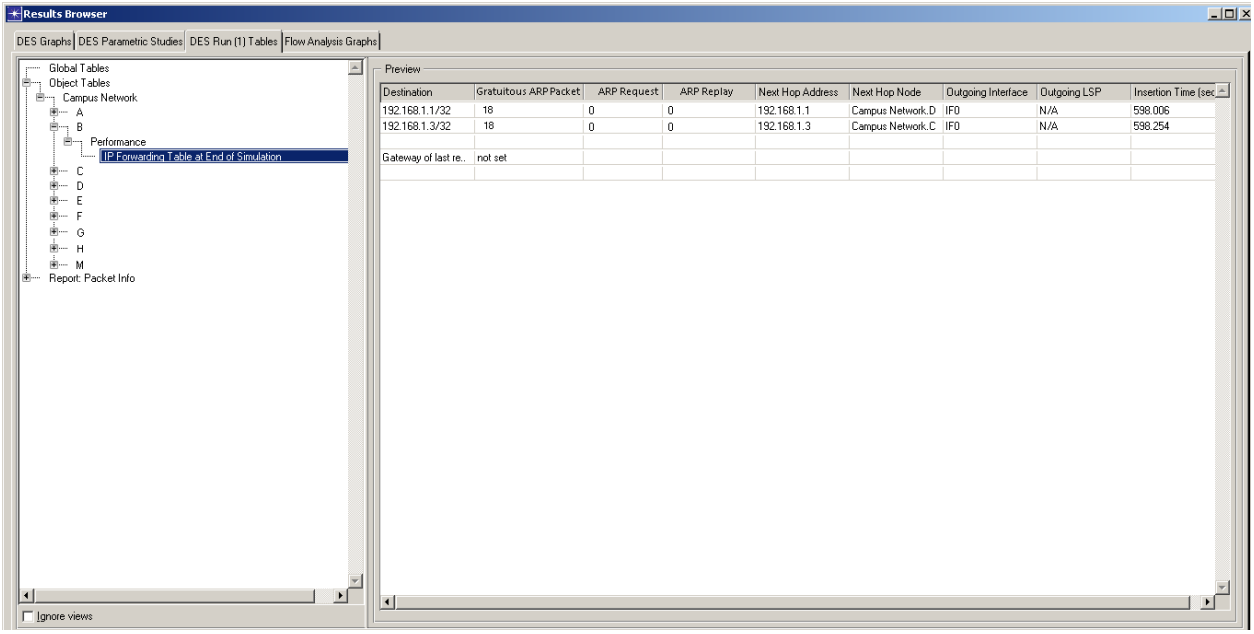


Figure 59: Node H Packet Exchange

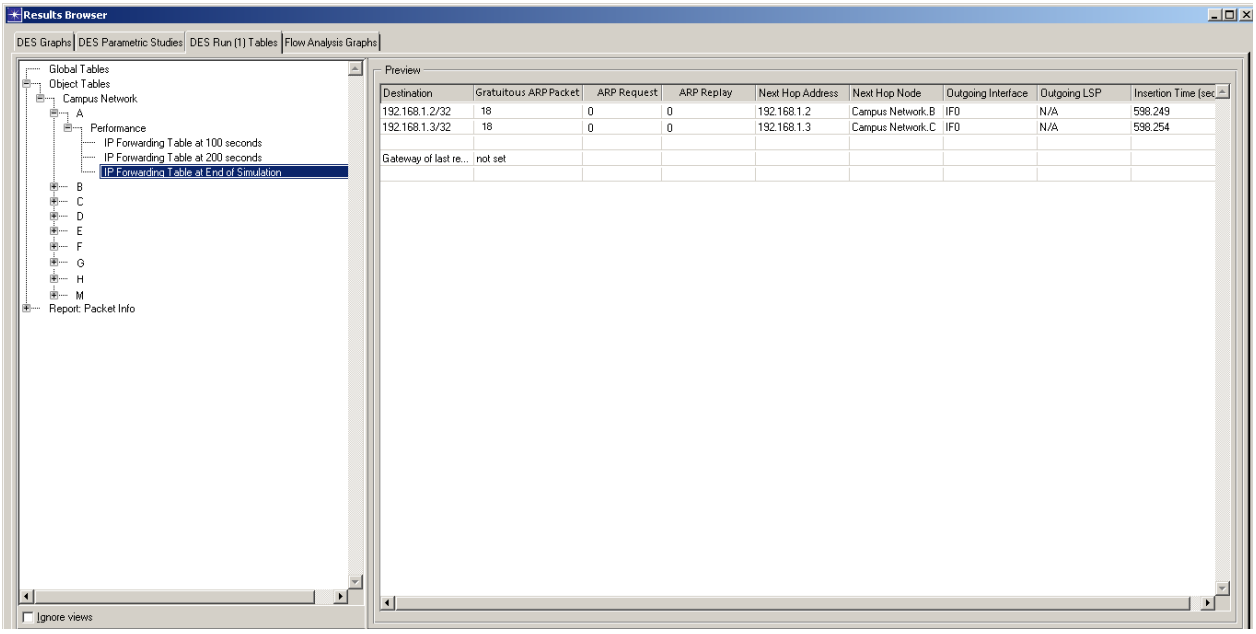


Figure 60: Node M Packet Exchange

References

[1] Mukesh Barapatre1, Prof. Vikrant Chole2, Prof. L. Patil, A Review on Spoofing Attack Detection in Wireless Ad hoc Network

[2] G.Bianchi, G.Neglia, V.Mancuso, A Review on Direct Datagram Forwarding: Address Resolution Protocol

[3] Heejo Lee and Kihong Park, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack," In Proceedings IEEE INFOCOM 2000, August 2000.

[4] Maurizio A. Spirito, "On the Accuracy of Cellular Mobile Station Location Estimation," IEEE Transactions on Vehicular Technology, Vol. 50, No. 3, May 2001.

[5] C. Hsu and C. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," IEEE Trans. Neural Networks, vol. 13, no. 2, Mar. 2002.

[6] T. Roos, P. Myllymaki, H.Tirri, P. Misikangas, and J. Sievanen, "A probabilistic approach to WLAN user location estimation," International Journal of Wireless Information Networks, vol. 9, no. 3, July 2002.

[7] Daniel B. Faria and David R. Cheriton, "DoS and Authentication in Wireless Public Access Networks," In Proceedings of the First ACM Workshop on Wireless Security (WiSe'02), September 2002.

[8] Bellardo and S. Savage, "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", Proc. USENIX Security Symp., August 2003.

[9] Mathias Bohge and Wade Trappe, "An Authentication Framework for Hierarchical Ad Hoc Sensor Networks," IEEE Trans. Ad Hoc Sensor Networks, WiSE'03, September 19, 2003.

[10] Pradeep Kyasanur and Nitin Vaidya, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks," In Proceedings the International Conference on Dependable Systems and Networks, San Francisco, CA, June 2003.

[11] P. Tao, A. Rudys, A. Ladd, and D. S. Wallach, "Wireless LAN Location-Sensing for Security Applications," In Proc. of the Second ACM Workshop on Wireless Security (WiSe'03), Sept. 2003.

[12] J.Hall, M.Barbeau, and E.Kranakis, "Enhancing Intrusion Detection in Wireless Networks Using Radio Frequency Fingerprinting," In Proc. of The IASTED Conference on Communications, Internet and Information Technology, Nov. 2004.

[13] A.Wool, "Lightweight Key Management for IEEE 802.11 Wireless Lans With Key Refresh

and Host Revocation," ACM/Springer Wireless Networks, vol. 11, no. 6, 2005.

[14] B.Wu, J.Wu, E.Fernandez and S.Magliveras, "Secure and Efficient Key Management in Mobile Ad Hoc Networks," Proc. IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS), 2005.

[15] Yang Xiao, Chaitanya Bandela and Yi Pan, "Vulnerabilities and Security Enhancements for the IEEE 802.11 WLANs," proc. IEEE Global Telecommunications Conference (GLOBECOM), 2005.

[16] F. Guo and T. Chiueh, "Sequence Number-Based MAC Address Spoof Detection," Proc. Eighth Int'l Conf. Recent Advances in Intrusion Detection, 2006.

[17] Xiao L and Trappe W, "Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication," Proceedings of IEEE International Conference on Communications (ICC), 2007.

[18] Qing Li and Wade Trappe, "Detecting Spoofing and Anomalous Traffic in Wireless Networks via Forge-Resistant Relationship," IEEE Transactions on Information Forensics and Security, Vol. 2, No. 4, December 2007.

[19] Ruiliang Chen, Jung-Min Park and Randolph Marchany, "A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of- Service Attacks," IEEE Transactions on Parallel and Distributed Systems, Vol. 18, No. 5, May 2007.

[20] Jie Yang and Yingying, "A Theoretical Analysis of Wireless Localization Using RF-based Fingerprint Matching," IEEE International Symposium on Parallel and Distributed Processing, (IPDPS), page 1-6, 2008.

[21] Zhenhai Duan, Xin Yuan and Jaideep Chandrashekar, "Controlling IP Spoofing through Interdomain Packet Filters," IEEE Transactions on Dependable and Secure Computing, Vol. 5, No. 1, March 2008.

[22] Kavitha Muthukrishnan, Berend Jan van der Zwaag, and Paul Havinga, "Inferring Motion and Location Using WLAN RSSI," proc. IEEE INFOCOM, 2009.

[23] J. H. Lee and R. M. Buehrer, "Location estimation using differential RSS with spatially correlated shadowing," in Proc. IEEE Global Commun. Conf. (GLOBECOM), November 2009.

[24] Yang, Y. Chen, and W. Trappe, "Detecting Spoofing Attacks in Mobile Wireless Environments," Proc. Ann. IEEE Comm. Soc.Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON), 2009.

[25] G Äuenther Lackner, Udo Payer and Peter Teu, “Combating Wireless LAN MAC-layer Address Spooing with Fingerprinting Methods,” proc IEEE International Journal of Network Security, Vol.9, No.2, Sept. 2009

[26] Gayathri Chandrasekaran, John-Austen Francisco, Vinod Ganapathy, Marco Gruteser and Wade Trappe, “Detecting Identity Spoofs in IEEE 802.11e Wireless Networks,” proc. IEEE GLOBECOM, 2009.

[27] Qi Zeng, Husheng Li and Lijun Qian, “GPS Spoofing Attack on Time Synchronization in Wireless Networks and Detection Scheme Design,” proc. IEEE MILCOM, 2012.

[28] Zhenghao Zhang, Matthew Trinkle, Lijun Qian and Husheng Li, “Quickest Detection of GPS Spoofing Attack,” Proc. IEEE Military Communications Conference (MILCOM), 2012.

[29] P. Kiruthika Devi, Spoofing Attack Detection and Localization in Wireless Sensor Network: A Review, Department of Computer Applications, K.S.Rangasamy College of Arts and Science

[30] Ahmed M.AbdelSalam ,Wail S.Elkilani and Khalid M.Amin, “An Automated approach for Preventing ARP Spoofing Attack using Static ARP Entries” Proc IEEE (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 1, 2014.

[31] M. Loganathan and V.Navaneethakrishnan, ”Detecting and Localizing Wireless Spoofing Attacks” IEEE Trans. International Journal of Advanced Research in Computer Science and Software Engineering vol4 Iss2, February 2014.

[33] Archana Shelar and M.D.Ingale, ”Modeling Security with Localization of Multiple Spoofing Attackers in Wireless Network,” The International Journal of Engineering And Science (IJES), Volume 3, Issue 01, 2014.

[34] Rehan Akbani, S. K. Upadhyay, “Different Types of Attacks on Integrated MANET-InternetCommunication.”

[35] Abhay Kumar Rai, Mukesh Barapatre, Prof. Vikrant Chole and Prof. L. Patil, “A Review on Spoofing Attack Detection in Wireless Adhoc Network”,International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 6, December 2013.

[36] Y. Chen, J. Francisco, W.Trappe, and R.P. Martin, “A Practical Approach to Landmark Deployment for Indoor Localization,” Proc. IEEE Int’l Conf. Sensor and Ad Hoc Comm. and Networks (SECON), Sept. 2006.

[37] A. Mitrokotsa, Archana Shelar and M.D.Ingale, ”Modeling Security with Localization of Multiple Spoofing Attackers in Wireless Network,” The International Journal Of Engineering And Science (IJES), Volume 3, Issue 01, 2014.

[38] B.Wu, J.Wu, E.Fernandez and S.Magliveras, “Secure and Efficient Key Management in Mobile Ad Hoc Networks,” Proc. IEEE Int’l Parallel and Distributed Processing Symp. (IPDPS), 2005.

[39] P.Goyal, Hao Yang, Haiyun Luo, Yi Yang, Songwu Lu and Lixia Zhang, “HOURS: Achieving DoS Resilience in an Open Service Hierarchy,” proc. IEEE Global Telecommunications Conference (GLOBECOM), 2003.

[40] Bart Wernik, “Arp Spoof Attack Mitigation and Threat Analysis”, In Mobile Ad Hoc Networks (Manets), Carleton University, August 2012