

**DETECTION OF MALICIOUS NODE  
IN MANET  
THROUGH FATIH**

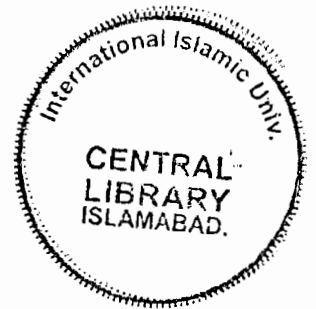
**MS THESIS**



**Nyla Khadam**  
(167-CS/MS/2003)

**Supervised By:**

**Mr. Khalid Hussain**  
**Mr. Mata-ur-Rehman**



**Department of Computer Science**  
**Faculty of Basic and Applied Sciences**  
**International Islamic University Islamabad**  
**2008**

**Department of Computer Science  
Faculty of Basic & Applied Sciences  
International Islamic University Islamabad**

Dated: 24-07-2008

**Final Approval**


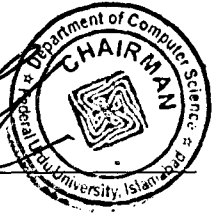
It is certified that we have read the Thesis Report submitted by Nyla Khadam (Registration number 167-CS/MS/03) and it is our judgment that this Thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the Masters of Science in Computer Science.

**Committee**

**External Examiner**

**Dr. M.A. Ansari**

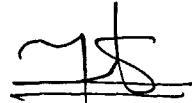
Chairman,  
Department of Computer Science,  
Federal Urdu University of Arts, Science & Technology  
Islamabad

**Internal Examiner**

**Dr. Muhammad Sher**

Chairman,  
Department of Computer Science,  
Faculty of Basic and Applied Sciences,  
International Islamic University,  
Islamabad

  
\_\_\_\_\_

**Internal Supervisor**

**Mr. Mata-ur-Rahman**

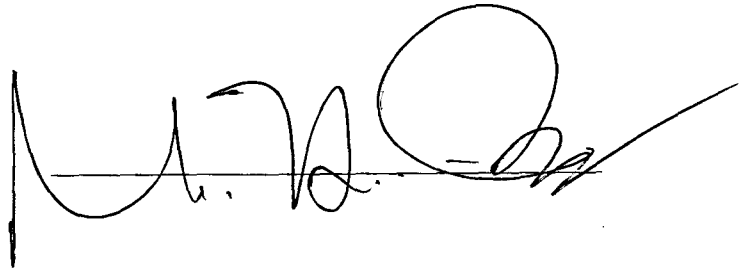
Assistant Professor,  
Department of Computer Science,  
Faculty of Basic and Applied Sciences,  
International Islamic University,  
Islamabad

  
\_\_\_\_\_

**External Supervisor**

**Mr. Khalid Hussain**

Director,  
IT & Communication,  
Crystal Communications,  
Wah Cantt

  
\_\_\_\_\_

A Dissertation submitted in partial fulfillment of the  
requirement for the degree of Master of Science in  
Computer Science

# Acknowledgements

All praise to Almighty Allah, the Most Merciful and Compassionate, Who enabled me to complete this work.

I would like to sincerely thank my Supervisors, Mr. Khalid Hussain and Mr. Mata-ur-Rehman who have made this work possible. I am extremely thankful for both the time and intellectual stimulation that Mr. Faraz Ahsan has graciously shared with me during this work.

I am grateful to all my friends for their care and guidance without which I would never be able to accomplish this task. I would especially like to thank Umair Masood, Khalil Afzal, Zubair Ahmed, Umair Naru, Ghazala Shaheen, Nadra Inam and Amna Farheen who always helped me in every way possible. I would also like to mention here the cooperation of all my colleagues and students throughout this thesis work.

Finally, I wish to thank my Family, who are the biggest supporters in my life. Their love and moral support have fostered and encouraged me to pursue my personal and academic interests. It is to them that I dedicate this work.

**NYLA KHADAM**  
**(167-CS/MS/03)**

## **ABSTRACT**

Rapid growth in Mobile Adhoc Networks is steadily escalating the number of mobile users and hence demands for improved security solutions also increased. Users can efficiently get advantages of an Adhoc network if it has strong security mechanism. But this is a fact that Adhoc Network is open for several types of attacks like wormhole, black hole and Byzantine attack. Therefore research in the domain of security solutions for Adhoc networks is gaining popularity day by day. Author have focused to counter black hole attack in wireless networks and have successfully simulated active Blackhole attack using NS-2.31 and Passive Blackhole attack using OMNeT++ environment. An enhancement in existing AODV routing protocol is suggested to avoid formation of black hole (sink) in the network. For this two different algorithms are designed to detect active and passive blackhole attack in Adhoc network environment working under AODV protocol.

**NYLA KHADAM**  
**(167-CS/MS/03)**

# TABLE OF CONTENTS

<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 MOTIVATIONS & CHALLENGES	1
1.2 BACKGROUND	2
1.2.1 Computer Networks	2
1.2.2 Adhoc Networks	2
1.2.3 Applications of Adhoc Networks	3
1.2.4 Characteristics of Adhoc Networks	4
1.2.5 Adhoc Routing Protocols	4
1.2.5.1 AODV	5
1.2.5.2 DSR	6
1.2.5.3 OLSR	8
1.2.6 Security Challenges	8
1.2.7 Security Attacks	9
1.2.8 Security Solutions	9
1.2.8.1 Preventive Mechanism	10
1.2.8.2 Reactive Mechanism	10
1.2.8.3 Handling Misbehaving Nodes	10
1.3 PROBLEM DOMAIN	11
1.4 PROPOSED APPROACH	12
1.5 THESIS OUTLINE	13
<b>CHAPTER 2: LITERATURE SURVEY</b>	<b>14</b>
2.1 RELATED WORK	14
2.1.1 AODV Protocol	14
2.1.2 Security Solutions for Wired Networks	16
2.1.3 Security Solutions for Wireless Networks	18
2.1.4 Security Solutions for Black hole Attacks in MANET	20
2.3 LIMITATIONS	22
2.4 SUMMARY	23
<b>CHAPTER 3: REQUIREMENT ANALYSIS</b>	<b>24</b>
3.1 NETWORK LAYER ATTACKS	24
3.1.1 Attacks on Particular Routing Protocols	25
3.2 PROBLEM SCENARIOS	26
3.3 FOCUS OF RESEARCH	28
3.4 SUMMARY	30
<b>CHAPTER 4: SYSTEM DESIGN</b>	<b>31</b>
4.1 DESIGN REQUIREMENTS	31
4.2 REFERENCE ARCHITECTURE	31
4.2.1 AODV Implementation	31
4.3 DESIGN METHODOLOGY	35
4.3.1 Algorithm	35
4.4 UML DIAGRAMS	36

4.4.1 Use Case Diagram	36
4.4.2 Class Diagram	37
4.4.3 Sequence Diagram	38
4.5 SUMMARY	39
<b>CHAPTER 5: IMPLEMENTATION</b>	<b>40</b>
5.1 NETWORK SIMULATOR 2	40
5.1.1 Class Hierarchies	40
5.1.2 TCL Linkage	41
5.1.3 Scheduler	41
5.2 OMNET ++	41
5.2.1 Hierarchical Modules	42
5.2.2 Module Types	43
5.2.3 Modeling of Packet Transmissions	43
5.2.4 Parameters	44
5.3 ACTIVE BLACKHOLE ATTACK IMPLEMENTATION	44
5.4 DETECTION MODEL FOR ACTIVE BLACKHOLE ATTACK	45
5.4.1 Mathematical Model	46
5.5 PASSIVE BLACKHOLE ATTACK IMPLEMENTATION	46
5.6 DETECTION MODEL FOR ACTIVE BACKHOLE ATTACK	47
5.6.1 Mathematical Model	48
5.7 SUMMARY	50
<b>CHAPTER 6: SIMULATION AND RESULTS</b>	<b>51</b>
6.1 SIMULATION SCENARIOS IN NETWORK SIMULATOR 2	51
6.1.1 Scenario I	51
6.1.2 Scenario II	53
6.1.3 Scenario III	56
6.2 SIMULATION SCENARIOS IN OMNET ++	59
6.2.1 Scenario I	59
6.2.2 Scenario II	61
6.2.3 Scenario III	62
6.3 RESULTS	64
<b>CHAPTER 7: CONCLUSION AND FUTURE WORK</b>	<b>65</b>
7.1 CONCLUSION	65
7.2 FUTURE WORK	65
<b>REFERENCES:</b>	<b>66</b>
<b>APPENDIX:</b>	

# TABLE OF FIGURES

Figure 1: Adhoc Networks	2
Figure 2: AODV Route Discovery	6
Figure 3: Route Discovery Process for DSR	7
Figure 4: Misbehaving node dropping packets	11
Figure 5: AODV Route Discovery Mechanism	15
Figure 6: AODV Route Maintenance Mechanism	15
Figure 7: Illustration of routing attacks	24
Figure 8: Transit packet byte controller	27
Figure 9: Black hole problem	29
Figure 10: Packet Flow	32
Figure 11: AODV Initialization	33
Figure 12: Transmission Packet Flow	33
Figure 13: Packet receiving in AODV	34
Figure 14: Use case diagram	36
Figure 15: Class diagram	37
Figure 16: Sequence diagram	38
Figure 17: Flowchart for Active Blackhole Attack Implementation	44
Figure 18: Flowchart for Active Blackhole Attack Detection Model	45
Figure 19: Flowchart for Passive Blackhole Attack Implementation	47
Figure 20: Flowchart for Passive Blackhole Attack Detection Model	48
Figure 21: Snapshot of Simulation Scenario I	51
Figure 22: Number of Packets sent at all the nodes	52
Figure 23: Number of Packets received at all the nodes	52
Figure 24: Number of Packets dropped at all the nodes	53
Figure 25: Throughput of dropping packets at Black hole Node	53
Figure 26: Snapshot of Simulation Scenario II	54
Figure 27: Number of Packets sent at all the nodes	54
Figure 28: Number of Packets received at all the nodes	55
Figure 29: Number of Packets dropped at all the nodes	55
Figure 30: Throughput of dropping packets at Black hole Node	56
Figure 31: Snapshot of Simulation Scenario III	56
Figure 32: Number of Packets sent at all the nodes	57
Figure 33: Number of Packets received at all the nodes	57
Figure 34: Number of Packets dropped at all the nodes	58
Figure 35: Throughput of dropping packets at Black hole Node	58
Figure 36: Topology of Simulation Scenario I	59
Figure 37(a): Packets sent and received at all the nodes	60
Figure 37(b): Packets dropped during transaction	60
Figure 38: Topology of Simulation Scenario II	61
Figure 39(a): Packets sent and received at all the nodes	62
Figure 39(b): Packets dropped during transaction	62
Figure 40: Topology of Simulation Scenario II	63
Figure 41(a): Packets sent and received at all the nodes	63
Figure 41(b): Packets dropped during transaction	64



# CHAPTER 01

## INTRODUCTION

# Chapter 1: Introduction

## 1.1 Motivations and Challenges

Wireless networks are experiencing unprecedented growth in the recent years. The primary reason being the greater user convenience promised by mobile computing. The wide proliferation of laptops, PDAs, and mobile phones means that there is an increasing number of devices on the move and hence there is a greater need to support such devices. In the wired networking world, a static network infrastructure is implicitly assumed to exist, with a host having the same point of attachment into the larger network over time. The user convenience promised by wireless networks allows a node to change its point of attachment over time and requires a number of additions to the existing network-layer architecture [1].

Advances in wireless technology and portable computing along with demands for greater user mobility have provided a major thrust toward development of an emerging class of self-organizing, rapidly deployable network architectures referred to as ad-hoc networks. An ad-hoc network is comprised of wireless nodes and requires no fixed infrastructure. Any device with a microprocessor, whether highly mobile or stationary, is a potential node in an ad-hoc network. This includes mobile telephones, motor vehicles, roadside information stations, satellites, and desktop or hand-held computing devices. Unlike existing commercial wireless systems and fixed infrastructure networks, ad-hoc networks cannot rely on specialized routers for path discovery and traffic routing. Consequently, mobile end-systems in an ad-hoc network are expected to act cooperatively to route traffic and adapt the network to the highly dynamic state of its links and its mobility patterns [2].

Ad hoc networks have a large number of potential applications. Military uses such as connecting soldiers or other military units to each other on the battlefield or creating sensory arrays with thousands of sensors are two typical examples. Ad hoc networks provide a possibility of creating a network in situations where creating the infrastructure would be impossible or prohibitively expensive. Unlike a network with fixed infrastructure, mobile nodes in ad hoc networks do not communicate via access points (fixed structures). Each mobile node acts as a host when requesting/providing information from/to other nodes in the network, and acts as router when discovering and maintaining routes for other nodes in the network [3].

Unique characteristics of mobile ad-hoc networks pose various challenges to the security design, such as open peer-to-peer network architecture, a shared wireless medium and a high dynamic topology. These challenges raised the requirement of developing security solutions that achieve wider protection and desirable network performance. The wireless channel in a mobile ad-hoc network is accessible to both legitimate network users and malicious attackers. The presence of malicious nodes [1] poses a grave threat to the very existence of an ad-hoc network. It is imperative to handle such nodes to prevent the legitimate nodes from being hit and to enable the ad-hoc network deliver its services.

## 1.2 Background

### 1.2.1 Computer Networks

A computer network is an interconnected collection of autonomous computers that are able to exchange information. The interconnection between these autonomous computers can be accomplished via copper wire, microwaves, infrared, fiber optics, or communication satellites. Computer networks enable resource sharing (for example, access to information regardless of the physical location of the resource and the user), provide a powerful communication medium (via e-mail, video-conferencing etc) and entertainment (via video on-demand, network gaming etc). Also, it is worth noting that the Internet is not just a single network but a collection of interconnected networks [17].

### 1.2.2 Ad Hoc Networks

An ad-hoc network is one that comes together as needed to meet the communication needs of the moment without relying on the existence of any preinstalled infrastructure to deliver its services. Each node in an ad-hoc network, if it volunteers to carry traffic, participates in the formation of network topology. The nodes in an ad-hoc network may be mobile so that two nodes within communication range at one point of time may be out of range some time later. Also, the nodes assist each other in the process of delivering packets of data as not all of them are within the range of each other. An example ad-hoc network is shown in Figure 1.

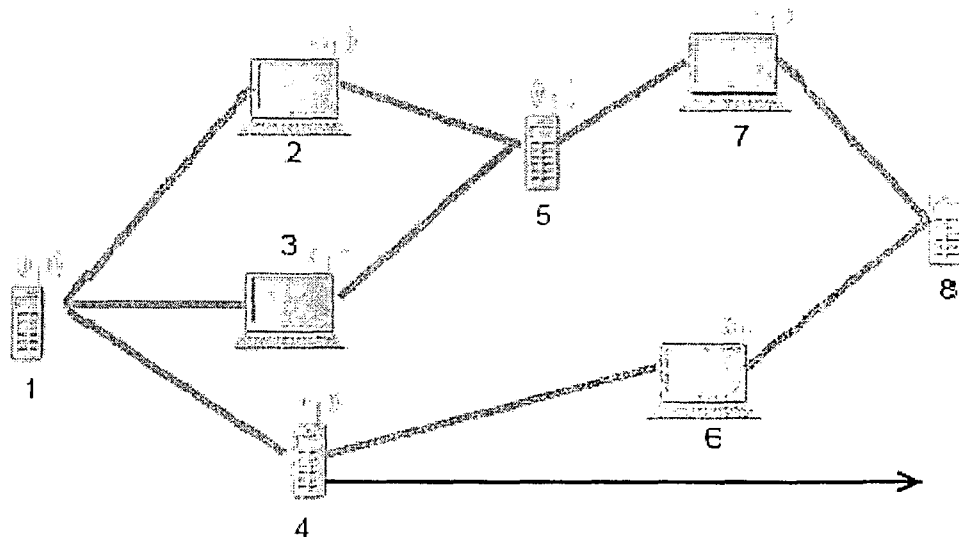


Figure 1: Adhoc Network

In an ad-hoc network, nodes are able to move relative to each other; as this happens, existing links may be broken and new links established [17].

### **1.2.3 Applications of Adhoc Networks**

Possible applications of MANET include: Soldiers relaying information for situational awareness on the battlefield, business associates sharing information during a meeting; attendees using laptop computers to participate in an interactive conference; and emergency disaster relief personnel coordinating efforts after a fire, hurricane, or earthquake [22].

Some important applications [1] of ad-hoc networks include:

#### **Spontaneous Networking:**

Ad hoc networking facilitates collaborative computing through “mobile conferencing” which allows mobile computer users to meet outside normal office environment and work towards a particular collaborative project. An ad-hoc network might be more desirable even when the Internet infrastructure is available. This results from the likely overhead required when utilizing infrastructure links, which might entail drastically sub optimal routing back and forth between separated office environments.

#### **Emergency Services:**

As the Internet grows in importance, the loss of network connectivity during natural disasters will become ever more noticeable and network applications will become increasingly important for emergency services. Ad hoc networks help to overcome network impairment during such emergencies. They can greatly aid in the “search-and-rescue” operations following the disaster.

#### **Military Applications:**

One of the main motivations for ad-hoc networks in military is the need for battlefield survivability. It is necessary to coordinate group actions avoiding single points of failure such as centralized control stations. Also, military cannot rely on preplaced communication infrastructure especially in jungles, deserts etc.

#### **Personal Area Networks:**

The idea of a personal area network (PAN) is to create a very localized network populated by some network nodes that are closely associated with a single person. These devices need to communicate with one another while they are associated with their users’ activities and mobility is not of significance in this scenario. But mobility becomes significant for inter-PAN communications and the methods of establishing communications between nodes on separate PANs could benefit from the technologies of ad-hoc networks.

## Sensor Networks:

Sensors are tiny, inexpensive devices used for gathering detailed information about a terrain or dangerous environmental conditions. These sensors can form an ad-hoc network and cooperate to gather the desired information. For example, they can be used to gather the chemical concentration level after a chemical explosion etc.

### 1.2.4 Characteristics of Ad hoc Networks

Mobile ad hoc networks exhibit properties different from fixed networks or infrastructure-based wireless networks. These properties make it harder to implement security services or even exhibit vulnerabilities to different and additional security attacks:

**Unreliable wireless links** are vulnerable to jamming and by their inherent broadcast nature facilitate eavesdropping.

#### Constraints in

- **bandwidth** are caused by the limits of the air interface with fading and noise.
- **computing power** in mobile devices require security mechanisms to be low in computation overhead.
- **battery power** in mobile devices can lead to application specific trade-offs between security and longevity of the device.

**Mobility/Dynamics** make it hard to detect behavior anomalies such as advertising bogus routes since routes in this environment change frequently. It is difficult to employ mechanisms like firewalls, because the border between being inside or outside the network is blurred.

**Self-organization** is a key property of ad hoc networks. They can not rely on central authorities and infrastructures. Therefore, trust management has to be distributed and adaptive [13]. On the bright side, self-organization leads to inherent better fault tolerance thanks to the absence of the potential bottleneck of centralized authorities [23].

### 1.2.5 Adhoc Routing Protocols

Ad hoc routing protocols of ad hoc networking are typically subdivided into two main categories:

- Proactive (Table-Driven) Routing Protocols
- Reactive (On-Demand) Routing Protocols

Proactive routing protocols are derived from legacy Internet distance-vector and link-state protocols. They maintain tables that store routing information. And for any change in network they triggers propagating updates throughout the network in order to maintain a consistent network view. This can cause substantial overhead affecting bandwidth

utilization, throughput as well as power usage. The advantage is that routes to any destination are always available without the overhead of a route discovery but such protocols cannot perform properly when the mobility rate in the network is high or when there are a large number of nodes in the network. Protocols in this category differ in the number of tables they contain as well as on the details of how they are updated. For example, nodes in Destination-Sequenced Distance Vector (DSDV) algorithm maintain route information to every other node in the network. As the network status changes full updates are exchange among all nodes. The Wireless Routing Protocol (WRP) localizes the updates to the immediate neighbors. When a new node A moves into range of a node B and a hello message is received from it, A is added to B's routing table and sent a full copy of the table. When a link fails, a node sends updates to its neighbors. The Cluster Gateway Switch Routing (CGSR) protocol reduces the size and amount of information propagation by having each cluster of nodes elect a cluster head. Network-wide information is only exchanged among the cluster heads. While the amount of information propagation is reduced, this results in inefficient routes. The Fisheye State Routing Protocol has been recently suggested, this differ from others in that the update frequency is inversely related to the distance between any two nodes.

Reactive (On-Demand) routing protocols are characterized by a path discovery mechanism that is initiated when a source needs to communicate with a destination that it does not know how to reach. The Route Discovery is usually in the form of query flood. Generally, on-demand routing requires less over-head than table-driven routing; but it incurs a path discovery delay whenever a new path is needed.

The differences between on-demand protocols are in the implementation of the path discovery mechanism and optimizations of it. Dynamic Source Routing (DSR) uses source routing, with every packet carrying the full path information with it [19,20]. Similarly, Ad hoc On-Demand of neighbors. Flooding is robust and well suited to Distance Vector Routing. AODV [21, 22] is an on-demand version of DSDV where the path results in exchange of the portions of the routing table necessary for establishing the route. Other on-demand algorithms include Temporally Ordered Routing Algorithm (TORA) that discovers multiple paths from a source to destination and re-initiates discovery only when all of them have failed [18].

#### **1.2.5.1 AODV**

The Ad hoc on-demand distance vector (AODV) routing protocol [5,7] is intended for use by mobile nodes in an ad hoc network. The AODV algorithm enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. One distinguishing feature of AODV is its use of destination sequence number for each route entry.

Operation of the AODV protocol can be divided in two functions – route discovery and route maintenance.

### Route discovery:

When a route to a new destination is needed, a RREQ (Route Request) packet is broadcast throughout the MANET with a search ring technique. On receipt of RREQ, the node creates a reverse routing entry towards the originator of RREQ, which is used to forward replies later. The destination or the intermediate node, which has a valid route towards the destination, answers with a RREP (Route Reply) unicast packet. On receipt of RREP, the reverse routing entry towards the originator of RREP is also created, similar to the processing of RREQ. When a link break in an active route is detected, a RERR message is used to notify other nodes that the loss of that link has occurred. The RERR message indicates those destinations, which are no longer reachable by way of the broken link. In order to enable this reporting mechanism, each node keeps a "precursor list". This list contains the upstream nodes, which use the node itself towards the same destinations.

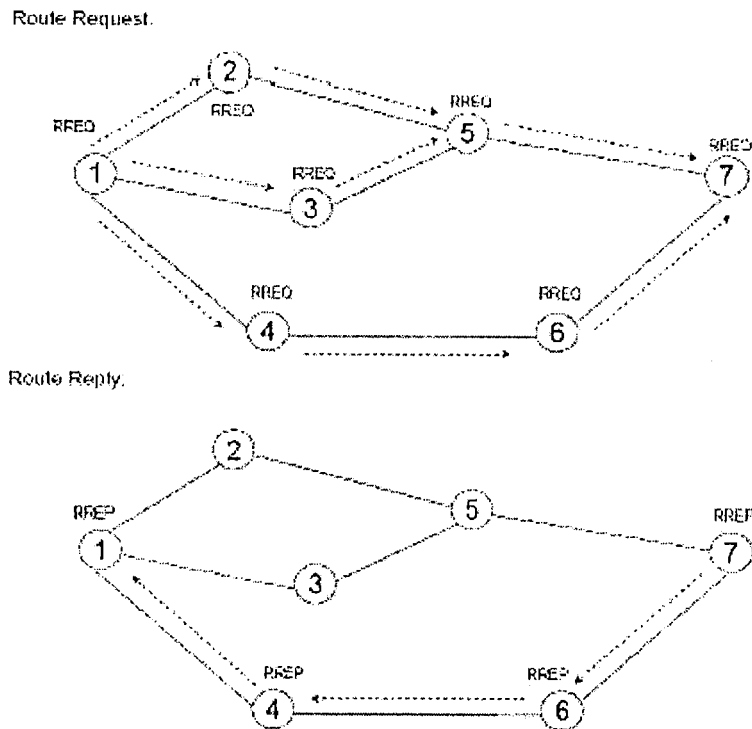


Figure 2: AODV Route Discovery Mechanism

### Route Maintenance:

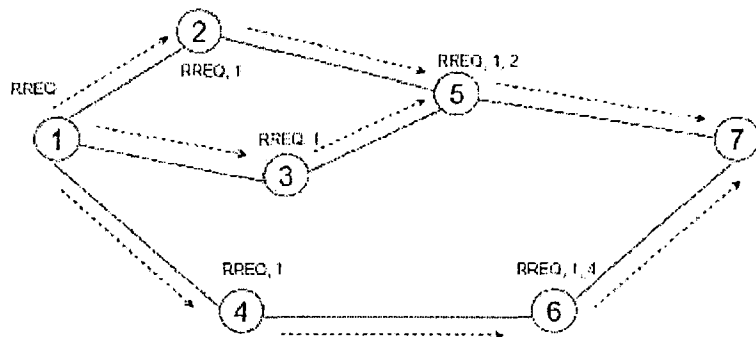
A node may offer connectivity information by broadcasting local HELLO messages. A node SHOULD only use hello messages if it is part of an active route. If the node does not receive a HELLO message or a data packet from a neighbor for a while, the link between itself and the neighbor is considered to be broken. If the destination with this neighbor as the next hop is believed not to be far away, local repair mechanism may be launched to rebuild the route towards the destination; otherwise, a RERR (Route Error) packet is sent to the neighbors in the precursor list associated with the routing entry to inform them of the link failure.

### 1.2.5.2 DSR

The Dynamic Source Routing (DSR) protocol [6,7] is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. The DSR protocol is designed mainly for mobile ad hoc networks of up to about two hundred nodes and is designed to work well even with very high rates of mobility.

The DSR protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network:

Route Request:



Route Reply:

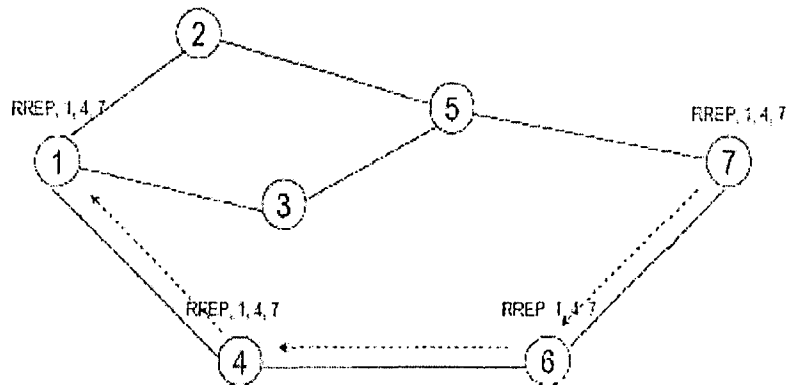


Figure 3: Route Discovery Process for DSR

- 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.



- Route Maintenance is the mechanism by which node S is able to 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, S can attempt to use any other route it happens to know to D, or it can invoke Route Discovery again to find a new route for subsequent packets to D. Route Maintenance for this route is used only when S is actually sending packets to D.

In DSR, Route Discovery and Route Maintenance each operate entirely "on demand". In particular, unlike other protocols, DSR requires no periodic packets of any kind at any layer within the network.

### 1.2.5.3 OLSR

Optimized link state routing (OLSR) protocol [7,8] aims at large and dense MANETS. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs are selected nodes, which forward broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared to a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message.

Nodes, which have been selected as multipoint relays by some neighbor node(s), announce this information periodically in their control messages. Thereby a node announces to the network, that it has reachability to the nodes, which have selected it as an MPR. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. Furthermore, the protocol uses the MPRs to facilitate efficient flooding of control messages in the network.

OLSR may optimize the reactivity to topological changes by reducing the maximum time interval for periodic control message transmission. Furthermore, as OLSR continuously maintains routes to all destinations in the network, the protocol is beneficial for traffic patterns where a large subset of nodes are communicating with another large subset of nodes, and where the [source, destination] pairs are changing over time. The protocol is particularly suited for large and dense networks, as the optimization done using MPRs works well in this context. The larger and more dense a network, the more optimization can be achieved as compared to the classic link state algorithm.

### 1.2.6 Security Challenges

Security is an essential service for wired and wireless network communications. The success of mobile ad hoc networks (MANET) strongly depends on people's confidence in its security. However, the characteristics of MANET pose both challenges and opportunities in achieving security goals, such as confidentiality, authentication, integrity, availability, access control, and non-repudiation [22]. One of the primary concerns in a Mobile Ad Hoc Network is to provide secure communication between

mobile hosts in a hostile environment. There is no standard security mechanism in a mobile ad-hoc network from the security design perspective to address this issue.

The nature of wireless adhoc networks makes them very vulnerable to an adversary's malicious attacks. First of all, the use of wireless links renders a wireless adhoc network susceptible to attacks ranging from passive eavesdropping to active interfering. Unlike wired networks where an adversary must gain physical access to the network wires or pass through several lines of defense at firewalls and gateways, attacks on a wireless ad-hoc network can come from all directions and target at any node. Damages can include leaking secret information, message contamination and node impersonation. All these mean that a wireless ad-hoc network will not have a clear line of defense, and every node must be prepared for encounters with an adversary directly or indirectly [2].

### 1.2.7 Security Attacks

The attacks in MANET can roughly be classified into two major categories, namely passive attacks and active attacks, according to the attack means. A passive attack obtains data exchanged in the network without disrupting the operation of the communications, while an active attack involves information interruption, modification, or fabrication, thereby disrupting the normal functionality of a MANET. Table 1 shows the general taxonomy of security attacks against MANET. Examples of passive attacks are eavesdropping, traffic analysis, and traffic monitoring. Examples of active attacks include jamming, impersonating, modification, denial of service (DoS), and message replay [22]. I will discuss network layer attacks in detail in section 3.1.

Layer	Attacks
Application layer	Repudiation, data corruption
Transport layer	Session hijacking, SYN flooding
Network layer	Wormhole, blackhole, Byzantine, flooding, resource consumption, location disclosure attacks
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11), WEP weakness
Physical layer	Jamming, interceptions, eavesdropping
Multi-layer attacks	DoS, impersonation, replay, man-in-the-middle

Table 1: Security Attacks

### 1.2.8 Security Solutions

A variety of security mechanisms have been invented to counter malicious attacks. The conventional approaches such as authentication, access control, encryption, and digital signature provide a first line of defense. As a second line of defense, intrusion detection systems and cooperation enforcement mechanisms implemented in MANET can also help to defend against attacks or enforce cooperation, reducing selfish node behavior.

### **1.2.8.1 Preventive mechanism:**

The conventional authentication and encryption schemes are based on cryptography, which includes asymmetric and symmetric cryptography. Cryptographic primitives such as hash functions (message digests) can be used to enhance data integrity in transmission as well. Threshold cryptography can be used to hide data by dividing it into a number of shares. Digital signatures can be used to achieve data integrity and authentication services as well.

It is also necessary to consider the physical safety of mobile devices, since the hosts are normally small devices, which are physically vulnerable. For example, a device could easily be stolen, lost, or damaged. In the battlefield they are at risk of being hijacked. The protection of the sensitive data on a physical device can be enforced by some security modules, such as tokens or a smart card that is accessible through PIN, passphrases, or biometrics.

Although all of these cryptographic primitives combined can prevent most attacks in theory, in reality, due to the design, implementation, or selection of protocols and physical device restrictions, there are still a number of malicious attacks bypassing prevention mechanisms.

### **1.2.8.2 Reactive mechanism:**

An intrusion detection system is a second line of defense. There are widely used to detect misuse and anomalies. A misuse detection system attempts to define improper behavior based on the patterns of well-known attacks, but it lacks the ability to detect any attacks that were not considered during the creation of the patterns; Anomaly detection attempts to define normal or expected behavior statistically. It collects data from legitimate user behavior over a period of time, and then statistical tests are applied to determine anomalous behavior with a high level of confidence. In practice, both approaches can be combined to be more effective against attacks. Some intrusion detection systems for MANET have been proposed in recent research papers.

### **1.2.8.3 Handling Misbehaving Nodes**

The presence of malicious nodes poses a grave threat to the very existence of an ad-hoc network. It is imperative to handle such nodes to prevent the legitimate nodes from being hit and to enable the ad-hoc network deliver its services. Detecting malicious behavior is the very first step in handling malicious nodes. Once malicious behavior is detected, the next step would be to identify the misbehaving node(s) in the ad-hoc network and then to finally isolate them so that the ad-hoc network can start functioning in accordance with its intended purpose without any performance hit.

There are three main steps in handling a malicious node.

- *Detection*

The first step in handling a malicious node is to detect the presence of any malicious nodes. This is done by looking for any distinct or peculiar network behavior such as increased packet drops or TCP timeouts at the source node.

- *Identification*

Once the presence of malicious node(s) is detected, the next step is to identify the misbehaving nodes(s). For example, a trace route mechanism can be used to identify a malicious node. After the successful identification of misbehaving node(s), all the nodes participating in the ad-hoc network should be informed so that they can avoid those nodes in their communication routes.

- *Isolation*

Once all the nodes in the ad-hoc network are aware of the malicious node(s), they can cooperate to isolate those nodes by denying to provide them with any kind of service (For example, denying packet forwarding on behalf of such nodes.) [1].

### 1.3 Problem Domain

Most current ad hoc routing protocols assume that the wireless network is benign and every node in the network strictly follows the routing behavior and is willing to forward packets for other nodes. Most of these protocols cope well with the dynamically changing topology. However, they do not address the problems when misbehavior nodes present in the network.

A commonly observed misbehavior is packet dropping. In a practical MANET, most devices have very limited computing and battery power while packet forwarding consumes a lot of such resources. Thus some of the mobile devices would not like to forward the packets for the benefit of others and they drop packets not destined to them. On the other hand, they still make use of other nodes to forward packets that they originate. These misbehaved nodes are very difficult to identify because we cannot tell that whether the packets are dropped intentionally by the misbehaved nodes or dropped due to the node having moved out of transmission range or other link error. Packet drop significantly decreases the network performance.

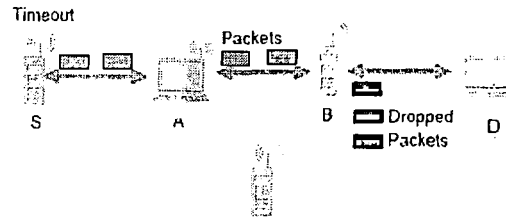


Figure 4: Misbehaving node dropping the packets

Traditional security mechanisms are generally not suitable for MANET because:

- 1) The network lacks central infrastructure to apply traditional security mechanism such as access control, authentication and trusted third party.
- 2) Limited bandwidth, battery lifetime, and computation power prohibits the deployment of complex routing protocols or encryption algorithms. New security models or mechanisms suitable for MANET must be found.
- 3) Network topologies and memberships are constantly changing. Thus new intrusion detection system and entity recognition mechanisms that are suitable for mobile ad hoc networks must be designed to avoid or mitigate the behavior to the networks [9].

In the case of routing, the correct transport of the packets on the network relies on the authenticity of the information given by the other nodes. The emission of false routing information by a host could thus create bogus entries in routing tables throughout the network making communication difficult. Furthermore, the delivery of a packet to a destination is based on a hop-by-hop routing and thus needs total cooperation from the intermediate nodes. A malicious host could, by refusing to cooperate, quite simply block or modify the traffic traversing it. By fooling the routing algorithm or even by choosing a strategic geographic positioning, a host can thus control the traffic to and from entire parts of the network [4].

In the context of anomalous routing behavior, a compromised router can pose two types of threats. Attack on control plane (by means of routing protocol) and attack on data plane (by having the router violate the forwarding decisions). Despite the fact that this type of attack presents a wider set of opportunities to the attacker the latter (*malicious forwarding*) has received comparatively less attention [16].

## 1.4 Proposed Approach

As we know that in wired networks the central point of communication is the router. All the data is forwarded from router to router towards its final destination. And if any of the intermediate router is compromised it starts misbehaving either by dropping, altering,

delaying or corrupting the packets. The problem of detecting and isolating the compromised routers is an emerging research topic. The researchers have so far presented many types of security solutions both for wired and wireless protocols but in some or other way they have deficiencies.

Unlike wired networks, there are no special routers in Mobile Adhoc networks. Every node in MANETS acts as a router and provides the mean to forward the data through it. Thus if a node misbehaves in MANET then it would be difficult to detect and isolate the malicious node.

I have studied different techniques related to security solutions in wired networks. Mizrak et al [15] presented a detection mechanism for wired networks and named it Fatih. His work is based on Conservation of Flow principle and he divided the problem into three sub problems: traffic validation, distributed detection and response.

Due to lack of collusion detection mechanism, even in an ideal wireless environment, there is collusion and packet drop. However, there are a number of factors on which successful packet transmission relies, mainly the environmental factors and limited resources of wireless devices. Both the said factors result in congestion and packet drop. Packet drop in wireless network is may be because of two reasons; lossy channel or malicious node. Any of the reason consequently decreases QOS and network performance. In terms of lossy channel, there does not exist a cutting edge on the basis of which a lossy channel is identified. Once, identified we can later further investigate what was the underlying reason for the loss of packets; through different existing algorithms for specific purposes. Blackhole attack adversely affects the performance of a network working under AODV protocol. I have focused to implement blackhole attack in AODV protocol as AODV routing protocol is vulnerable to this type of attack. I have also designed an algorithm which can identify blackhole attack in AODV protocol.

## **1.5 Thesis Outline**

The rest of the thesis is organized as follows. In chapter 2, the review of the related research in the area of wireless security is presented. Additionally, it discusses the techniques for malicious node identification and isolation. In chapter 3, problem domain is described in detail along with problem scenarios. The architecture and methodology of proposed solution is presented in chapter 4. The implementation and testing details are discussed in chapter 5 and 6. In chapter 7, final remarks are presented and the future scope of this work is discussed.

## **CHAPTER 02**

# **LITERATURE SURVEY**

## Chapter 2: Literature Survey

With the growth in wireless network usage, problems related to network security are increasing day by day. The secure functioning of routers and routing protocols is an important network security challenge. Wireless network routing is vulnerable to the attacks such as denial of service, surveillance and man in the middle attack. As the network routers play a key role in distributed system so they are attractive targets for attackers. A compromised router may drop, delay, reorder, corrupt or re-route any of the packets passing through it. The detection of compromised routers and their removal from the routing framework is becoming a new research goal.

Computer networks are susceptible to attacks, which is true even in the case of wired networks. But the unique features of the mobile ad-hoc networks such as the lack of fixed infrastructure, dynamic topology, and shared wireless medium make them all the more vulnerable to attacks when compared with their wired counterparts. So, the security techniques that have been proven useful for wired networks may not be directly applicable for ad-hoc networks [1].

Most research effort has been put in the routing protocols since the advent of the MANET as the traditional routing protocols used in hardwired networks, such as distance vector protocols (e.g. RIP) and link state protocols (e.g., OSPF) cannot be applied in the MANET directly, as stated in [12], for the following reasons:

- (1) There may be uni-directional links between nodes;
- (2) There are more than one eligible path between two nodes;
- (3) The consumption of bandwidth and power supply incurred by periodic routing information updates is considerable;
- (4) The routing fabrics converge slowly in contrast to rapid topology change.

In MANET environment, there is no guarantee that a path between two nodes would be free of malicious nodes, which would not comply with the employed protocol and attempt to harm the network operation. The mechanisms currently incorporated in *MANET* routing protocols cannot cope with disruptions due to malicious behavior [11].

### 2.1 Related Research

#### 2.1.1 AODV Protocol

The Ad hoc on-demand distance vector (AODV) routing protocol [5,7] is intended for use by mobile nodes in an ad hoc network. The AODV algorithm enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. One distinguishing feature of AODV is its use of destination sequence number for each route entry.

Operation of the AODV protocol can be divided in two functions – route discovery and route maintenance.



### Route discovery:

When a route to a new destination is needed, a RREQ (Route Request) packet is broadcast throughout the MANET with a search ring technique. On receipt of RREQ, the node creates a reverse routing entry towards the originator of RREQ, which is used to forward replies later. The destination or the intermediate node, which has a valid route towards the destination, answers with a RREP (Route Reply) unicast packet. On receipt of RREP, the reverse routing entry towards the originator of RREP is also created, similar to the processing of RREQ. When a link break in an active route is detected, a RERR message is used to notify other nodes that the loss of that link has occurred. The RERR message indicates those destinations, which are no longer reachable by way of the broken link. In order to enable this reporting mechanism, each node keeps a "precursor list". This list contains the upstream nodes, which use the node itself towards the same destinations.

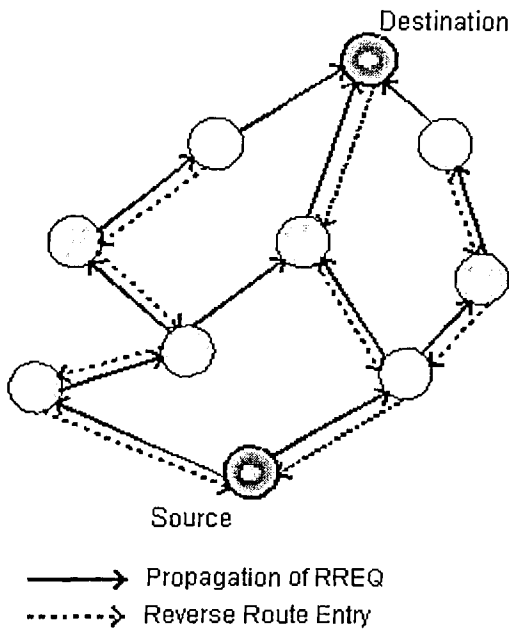


Figure 5: AODV Route Discovery Mechanism

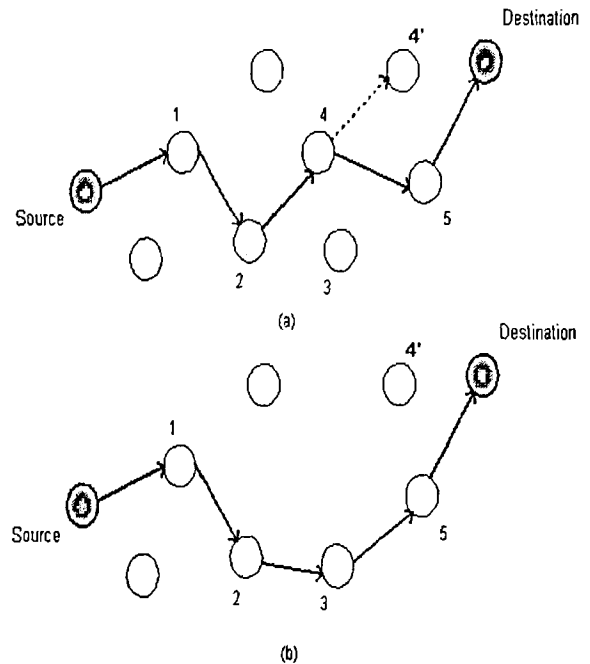


Figure 6: AODV Route Maintenance

### Route Maintenance:

A node may offer connectivity information by broadcasting local HELLO messages. A node SHOULD only use hello messages if it is part of an active route. If the node does not receive a HELLO message or a data packet from a neighbor for a while, the link between itself and the neighbor is considered to be broken. If the destination with this neighbor as the next hop is believed not to be far away, local repair mechanism may be launched to rebuild the route towards the destination; otherwise, a RERR (Route Error) packet is sent to the neighbors in the precursor list associated with the routing entry to inform them of the link failure.

## 2.1.2 Security Solutions for Wired Networks

### SATS

SATS detects malicious routers on the data plane. SATS can detect set of misbehaving routers when packets do not follow their predicted paths. It uses the mechanism of packet sampling. They also assume that the first and last router on a path is correct. Between these correct routers SATs can detect set of malicious routers. SATS mechanism uses split range assignment function to assign multiple hash ranges to the routers. In next step, routers using assigned sampling range sample subset of packets. Hash labels and keys of sampled packets are reported to backend engine. The work of routers ends here. Rest of the functions is performed by backend engine. After each interval, the trajectories of sampled packets reported from routers are reconstructed. Aggregation of trajectories with same ingress router and destination routing prefix pair is done. Then SATS identifies inconsistent trajectories from their predicted trajectories. Upon the identification of inconsistent trajectory, SATS locates a set of suspicious routers [31].

### HOP-BY-HOP Acknowledgements

Perlman does the earliest work about the security mechanism in networks. Perlman first proposed the idea of hop-by-hop acknowledgements. In her developed protocol a source routers first computes a route based on its local dab and then sends packet along the chosen route. Source needs to receive an acknowledgement from destination node and from each router on the path from source to the destination. If the source node does not receive acknowledgement from the data packet from the destination node, then it detects that the chosen route is not reliable and computes a new route [32].

### WATCHERS

Bradley et al in [13] presented a distributed monitoring protocol designed to detect and isolate malicious routers. The protocol is basically based on Law of Conservation of Flow which states that all data bytes sent into a node and not destined for that node are expected to exit the node. According to WATCHERS a faulty router is defined to be one that drops or misroutes packets or that behaves in an arbitrary manner.

WATCHERS is designed to work in networks with following conditions.

- Link state condition:  
Good routers agree on the exact topology of the network
- Good neighborhood condition:  
Each outer is a neighbor to at least one good router.
- Good path condition:  
Each pair of good routers has at least one path of only good routers connecting them.
- Majority good condition:  
A majority of the routers are good. This is required to prevent faulty routers from triggering a new round of the protocol.

WATCHERS maintain two types of counters for transit traffic and misrouted traffic. Each router counts how many bytes it has received and forwarded through each link during a specified time interval. Each router then floods the snapshots of its counters. Counter values are exchanged among the routers using request, receive and respond sub protocol. After the counter values are exchanged two-phase protocol is used to determine which routers are faulty.

Validation is the mechanism via which a router  $a$  compares, for each neighbor  $b$ , its counters for the  $a \leftrightarrow b$  link with those of  $b$ . If the counters do not agree, it detects its neighbor as faulty. Similarly, for each neighbor  $b$  and each of its neighbor  $c$ ,  $a$  compares the  $b \leftrightarrow c$  link counters of  $b$  with those of  $c$ . If these counters do not agree, then  $a$  knows that at least one of  $b$  and  $c$  is faulty, and so  $a$  does nothing further with  $b$ ; it assumes that  $b$  will detect  $c$  as faulty or vice versa.

If the validation phase is passed successfully, then  $a$  checks if each neighbor  $b$  preserves Conservation of flow test (CoF). It does so by computing the incoming transit flow  $I_b$  and the outgoing transit flow  $O_b$  of router  $b$ :

$$I_b = \sum_{c|b \leftrightarrow c} (S_{c,b} + T_{c,b}) \quad O_b = \sum_{c|b \leftrightarrow c} (D_{b,c} + T_{b,c})$$

If  $|I_b - O_b| > T$  for some threshold  $T$  then  $a$  diagnoses  $b$  as *faulty*.

Each router maintains six counters for each of its neighbors. Since all counters are compared over the same time interval, all of the routers periodically synchronize with each other.

## FATIH

According to A.Mizrak et al [15], the problem of detecting and removing compromised routers can be broken into three distinct sub problems: Traffic validation, distributed detection and response. Traffic information is the basis of detecting anomalous behavior. When the monitored traffic leaving one part of the network differs significantly from what is expected, anomalous behavior is detected. Traffic validation mechanism can be represented as a predicate

$$TV(\pi, info(r_i, \pi, \tau), info(r_j, \pi, \tau))$$

TV mechanism requires buffers to store packets information. It also consumes bandwidth for resending this packet information. All this makes this approach practically complex. Bad behavior of routers is divided into five different threats: Packet loss, packet fabrication, packet modification, packet reordering and time behavior. When the metrics for all of these threats are zero then no router is found forwarding traffic in a faulty manner.

Synchronizing the collection of traffic information and distributing the results for detection purposes is the second problem in detecting compromised routers. The solution

to this problem is a protocol, which is presented in this paper and is named  $\pi(k+2)$ . Since routers collect the information upon which traffic validation is based, there will be some uncertainty in determining which router is faulty. Since this detector is based on evaluating traffic collected over a period of time, so failure detector report pairs  $(r, \tau)$  are used, which means that  $r$  was suspected as being faulty during the time interval  $\tau$ . As the failure detector is based on traffic information collected by untrustworthy routers. This results in detections that are imprecise. So, the failure detector returns a pair  $(\pi, \tau)$  where  $\pi$  is a path segment. This restricted the detection to a router being faulty with respect to forwarding traffic along  $\pi$ . For a detection protocol a simple approach is to use consensus to distribute the traffic information and then have each router use  $TV$  to implement its failure detector.

In  $\pi(k+2)$   $TV$  is applied just for the end nodes of each path segment in  $Pr$ . A router needs only keep track of each  $x$ -path segment of which it is one of the end nodes, for some value of  $x$ . After a path segment  $\pi$  is detected as containing compromised routers the proper countermeasures should be taken. It would be to remove all of the routers in  $\pi$  from the routing framework or to remove the path segment  $\pi$  from the routing framework.

### 2.1.3 Security Solutions for Wireless Networks

#### Watchdog and Pathrater

S. Marti et al presented a mechanism for detecting misbehaving nodes and reducing their performance impact in wireless adhoc networks. They proposed a mechanism consisting of two components. Watchdog is used to identify misbehaving nodes. It maintains a buffer of recently sent packets and compares each overheard packet with the packet in the buffer. If packets match, then its entry in the buffer is removed since it has been forwarded on. Watchdog might not perform good detection in the presence of ambiguous collisions, receiver, collisions, limited transmission power etc. In path rater mechanism, each node maintains a rating for every other node it knows about in the networks. A path metric is calculated by averaging the node ratings in the path [33].

#### MobIDS

This paper is focused on detection of selfish nodes. For this purpose, different type of sensors are used to find selfish nodes, they named the mechanism as Mob IDS. This mechanism enhanced the existing over hearing and also developed new sensors so that they can be used in parallel with old schemes to achieve high rate of detection accuracy. The sensors, which are used, generate two types of observations. Positive behavior is observed if the value is positive where as faulty or non-cooperative behavior is observed if the value is negative. Sensor rating is calculated by using all local observations of a node  $K_i$  and a sensor  $S$  regarding another node  $K_j$  at time  $t$ . After this, all these sensor ratings are combined into a local rating. Local rating is the judgment made by a node  $k_i$  regarding node  $K_j$  at time  $t$ . Distribution of local ratings to neighboring nodes is done by flooding in a certain diameter around a node. When a node starts receiving local ratings it

takes an average of these rating and finally generates global rating. Nodes are isolated or removed from the network depending on the global rating. The classical overhearing sensor is improved by using activity based overhearing. And binary probing is replaced by iterative probing. To resolve probing dilemma, iterative probing is combined with overhearing [34].

### **Intrusion Detection**

An intrusion detection and response mechanism has been proposed Zhang et al in [2]. The authors of this paper emphasize on the point that ID and response systems should be both distributive and cooperative. A majority voting mechanism is used to classify the malicious behavior by consensus. Intrusion response can be to reinitialize communication channels between nodes or to isolate compromised nodes.

### **Confidant**

Sonja et al presented a protocol for detecting and isolating misbehaving nodes in Adhoc networks. In this mechanism, each node is responsible of monitoring the behavior of its next hop neighbors. When a node detects anomalous behavior, the information is given to the reputation system (a module of CONFIDANT). The event is checked for a predefined threshold value. This threshold value helps to distinguish between malicious behavior and collisions. When threshold value exceeds, the reputation system updates the misbehaving node's rating. When this rating level becomes high then the information is passed to the path manager then the path manager deletes all those paths from the path cache containing misbehaving nodes. A trust manager can warn the node by sending an alarm message containing the information like type of protocol violation, number of occurrences observed, address of reporting node, address of observed node and the destination address.

This ALARM message is received by the Monitor component of a node. It then passes it to the trust manager where source of the message is assessed. The ALARM table is updated if the source is partially trusted. And if the node is suspected to be malicious then the information is sent to reputation system so that it can evaluate it once again [35].

### **Unobtrusive Monitoring**

S. Medidi et al [1] presented a solution to locate malicious or faulty nodes that drop packets. The mechanism is based on a detection manager, which uses information from different network levels to detect faulty nodes. The detection manager is implemented on the source nodes requesting service. Local messages like route request, error messages and time outs are stored in the detection manager. The data collection component gathers useful information from detection manager when required and passes this data to the data storage component. Data analysis component deals with only related and relevant data and detects any malicious activity. If there is any, the detection manager intimates the node for further action.

Khalid et al [21] have presented a method to evaluate an ideal conservation of flow within a wireless mesh network. They have implemented metrics to measure conservation of flow in terms of end-to-end data delivery. Additionally, if a node fails to send a packet forward, for any reason, it is being logged and the corresponding initiator of the packet is updated. Thus, the total packets sent were either delivered to the intended destination or dropped (which is traceable), but is conserved within the network; resulting in conservation of flow.

In [27] Faraz et al have presented a scheme that integrates or incorporates error rate in wireless channel for mesh network environment. The proposed idea is based on error rate in a wireless channel that can affect conservation of flow. Additionally, an ideal channel bit rate for wireless mesh network has also been implemented to achieve maximum conservation of flow. They have also presented a method to identify a lossy channel within a wireless mesh network.

#### **2.1.4 Security Solutions for Black hole Attack in MANET**

Deng et al [24] studied the security issues related to routing in MANET. Their analysis was focused to one type of attack called “blackhole attack.” They proposed a feasible solution to the blackhole problem in AODV protocol.

In the proposed method, each intermediate node sends the next hop information when it sends back a RREP. When source node gets the RREP, it doesn't send the data immediately. Source node extracts the next hop information from reply message and then sends a “further Request” message to it. This is done to verify that the next hop node has a route to the intermediate node who sends back the reply message. It is also restricted that only the requested next hop can send further reply. Further reply message includes check result. Further reply from the inquired intermediate node is ignored here.

When source node receives Further reply from next hop, it extracts the check result from the reply message. If the result is yes, it means route to destination is secure. So source node begins sending data. If the next hop has no route to the requested intermediate node and destination, source node initiates another routing discovery process and also notifies the network about the malicious node by sending an alarm message.

S. Ramaswamy et al [3] focused on detection of multiple blackholes cooperation with each other as a group. The technique is based on AODV protocol which is modified to have DRI table in addition to the cached and current routing table. In this solution, every node maintains a DRI table in addition to routing table. In the modified AODV, the source node broadcasts RREQ message to discover a secure route to the destination node. The intermediate node generating the RREP has to provide its next hop node and its DRI entry for next hop node. When source node receives RREP from intermediate node, it checks its own DRI table to see whether intermediate node is a reliable node. If source node has used intermediate node before to route data, then it considers it a reliable node and starts routing data through it otherwise intermediate node is unreliable and source

node sends further request message to next hop node to check the identity of intermediate node.

Source node asks next hop node;

1. If intermediate node has routed data packets through next hop node.
2. Who is current next hop node's next hop to destination.
3. Has the current next hop node routed data packets through its own next hop.

Next hop node responds with further request message including

1. DRI entry for intermediate node
2. next hop node of current next hop node
3. DRI entry for the current next hop node's next hop

If source node has routed data through NHN before it means NHN is reliable otherwise it is unreliable. If NHN is unreliable, source node treats current NHN as intermediate node and sends further request to updated IN's next hop node. If NHN is reliable source node will check whether intermediate node is a blackhole or not. If second bit of DRI entry from intermediate node is 1 and first bit of DRI entry from NHN is 0, it means Intermediate node is a blackhole. (It means intermediate node has routed data through NHN but NHN has not routed any data from intermediate node)

If intermediate node is not a blackhole and NHN is a reliable node, route is secure and source node will update its DRI entry for intermediate node with 01 and starts routing data through intermediate node. If intermediate node is blackhole, SN identifies all the nodes along the reverse path from intermediate node to the node that generated RREP as blackhole nodes and ignores any other RREP from blackholes and announces the list of cooperative blackholes.

Bo sun et al [25] worked on detecting blackhole attack in MANET. According to their method, after the completion of normal path discovery procedure in a routing protocol. The source node sends a special control packet (RQNS) to request the destination to send its current neighbor set. Source node sends RQNS (unicasting) to each received RREP. Each node replies to RQNS by sending RPNS (reply neighbor set). When source node receives more than one RPNS in a certain period, it starts comparing neighbor sets. If the difference is larger than the predefined threshold value, source node detects that network has blackhole attack.

S. Dokurer et al [26] simulated blackhole attack on AODV protocol by introducing a new routing protocol called blackhole AODV which simulates the blackhole behavior in NS-2. Nodes which will adopt this new protocol will be marked as blackhole nodes and will behave exactly like blackhole. When a source node broadcasts the RREQ message for any destination, the node acting as blackhole node immediately responds with RREP message containing sequential number. Because of highest sequence number this message is perceived as if it is coming from the destination or from a node having fresh enough route to the destination. Hence the source discards all other RREP messages.

Authors also proposed a solution based on ignoring the first established route to reduce the adverse effects of blackhole node in an Adhoc network using AODV protocol.

## 2.2 Limitations

Up to now, almost all of the mobile ad hoc routing protocols lack built-in security. Retrofitting security mechanisms into these routing protocols are often expensive [26].

Hughes et al in [14] examined the WATCHERS protocol, which was previously presented by Bradley et al for detecting misbehaving routers. As WATCHERS algorithm uses COFP as a test, according to reviewers, it is not suitable to be used as a security mechanism in network protocols. The assumptions made by WATCHERS designers for using Conservation of Flow mechanism are not verified and so they can be defeated.

The mechanism used for reviewing consists of examining the strength of WATCHERS by applying several attacks on routers such as packet modification, packet substitution, ghost routers, source routing and many more. After considering these attacks it was concluded that WATCHERS algorithm states firmly that a modified packet is equivalent to a misrouted packet. It also assumed that all the routers running the protocol know the network topology. Several implicit assumptions made by WATCHERS are also analyzed. The problems that occur because of these implicit assumptions are listed along with possible solutions in this paper. However, memory requirement and performance costs for WATCHERS are still to be calculated. Also diagnosis of super node (group of several nodes) behavior should be done as future work. In addition, broadcast packets are not currently accounted for in the WATCHERS protocol.

Currently, SATS is designed for single administrative domain. Extending it to multiple domains is to be done as a future work. Perlman's security mechanism was quite immature and it can be defeated by consorting bad routers (routers that cooperate to hide malicious behavior). However, at that time many implementation details were left open. The future work of intrusion detection system constitutes the study of detection rate and performance penalties. The work done by Marti et al deals with the throughput ratio. Other factors like latency are still to undergo. Performance analysis of CONFIDANT protocol over time and extension of its implementation for other attacks is stated as future work. In MobIDs Threshold value is to be automatically set; this is left as a future work. The technique presented by S.Medidi only detects packet-dropping nodes. Enhancement of detection manager to detect other kinds of misbehaviors is left as a future work.

Black hole problem in MANETS [24] is a serious security problem to be solved. In this problem, a malicious node uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. In flooding based protocol, if the malicious reply reaches the requesting node before the reply from the actual node, a forged route has been created. This malicious node then can choose whether to drop the packets to perform a denial-of-service attack or to use its place on the route as the first step in a man-in-the-middle attack.



The method proposed by Deng et al assumes that malicious nodes do not work as a group. And they left the work to be done in future. S. Ramaswamy et al [3] focused on detection of multiple blackholes cooperation with each other as a group. The technique is based on AODV protocol which is modified to have DRI table in addition to the cached and current routing table. Maintenance of cross checking DRI table in this solution becomes costly. The cost of cross checking the nodes can be minimized by letting nodes sharing their trusted nodes list (DPS table) with each other.

### **2.3 Summary**

The literature surveyed shows that the issue of detection of malicious routing behavior is gaining popularity day by day. Malicious routing behavior is a serious problem in wireless network routing. Misbehaving nodes try to draw traffic towards them and later fail to forward the traffic correctly. One can identify compromised nodes with the help of correct nodes. Many earlier techniques, which are proportionally large in number, have worked on it but still have some flaws [13,14,15,16]. Most of them tried to detect malicious forwarding in wired networks. The problem is to correctly identify the misbehaving node and then distribute this information to all the correct nodes and at the end as a response of this detection isolate the faulty nodes.

# **CHAPTER 03**

## **REQUIREMENT ANALYSIS**

## Chapter 3: Requirement Analysis

There are a wide variety of attacks that target the weakness of MANET. For example, routing messages are an essential component of mobile network communications, as each packet needs to be passed quickly through intermediate nodes, which the packet must traverse from a source to the destination. Malicious routing attacks can target the routing discovery or maintenance phase by not following the specifications of the routing protocols. There are also attacks that target some particular routing protocols, such as DSR, or AODV. More sophisticated and subtle routing attacks have been identified in recent published papers, such as the blackhole (or sinkhole), Byzantine, and wormhole attacks. Currently routing security is one of the hottest research areas in MANET.

### 3.1 Network layer attacks

Network layer protocols extend connectivity from neighboring 1-hops nodes to all other nodes in MANET. The connectivity between mobile hosts over a potentially multi-hop wireless link strongly relies on cooperative reactions among all network nodes. A variety of attacks targeting the network layer have been identified and heavily studied in research papers. By attacking the routing protocols, attackers can absorb network traffic, inject themselves into the path between the source and destination, and thus control the network traffic flow, as shown in Figure 7, where a malicious node M can inject itself into the routing path between sender S and receiver D.

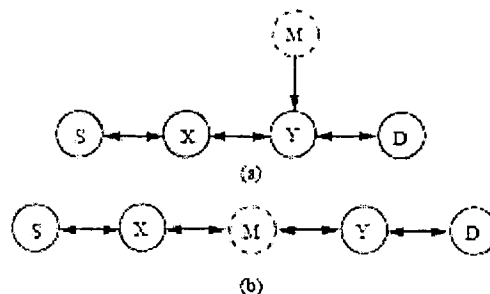


Figure 7: Illustration of routing attacks

The traffic packets could be forwarded to a non-optimal path, which could introduce significant delay. In addition, the packets could be forwarded to a nonexistent path and get lost. The attackers can create routing loops, introduce severe network congestion, and channel contention into certain areas. Multiple colluding attackers may even prevent a source node from finding any route to the destination, causing the network to partition, which triggers excessive network control traffic, and further intensifies network congestion and performance degradation.

### 3.1.1 Attacks on particular routing protocols

There are attacks that target some particular routing protocols. In DSR, the attacker may modify the source route listed in the RREQ or RREP packets. It can delete a node from the list, switch the order, or append a new node into the list. In AODV, the attacker may advertise a route with a smaller distance metric than the actual distance, or advertise a routing update with a large sequence number and invalidate all routing updates from other nodes.

More sophisticated and subtle routing attacks have been identified in recent research papers. The blackhole (or sinkhole), Byzantine, and wormhole attacks are the typical examples, which are described in detail below.

#### **Wormhole attack:**

An attacker records packets at one location in the network and tunnels them to another location. Routing can be disrupted when routing control messages are tunneled. This tunnel between two colluding attackers is referred as a wormhole. Wormhole attacks are severe threats to MANET routing protocols. For example, when a wormhole attack is used against an on-demand routing protocol such as DSR or AODV, the attack could prevent the discovery of any routes other than through the wormhole.

#### **Blackhole attack:**

The blackhole attack has two properties. First, the node exploits the mobile ad hoc routing protocol, such as AODV, to advertise itself as having a valid route to a destination node, even though the route is spurious, with the intention of intercepting packets. Second, the attacker consumes the intercepted packets without any forwarding. However, the attacker runs the risk that neighboring nodes will monitor and expose the ongoing attacks. There is a more subtle form of these attacks when an attacker selectively forwards packets. An attacker suppresses or modifies packets originating from some nodes, while leaving the data from the other nodes unaffected, which limits the suspicion of its wrongdoing.

#### **Byzantine attack:**

A compromised intermediate node works alone, or a set of compromised intermediate nodes works in collusion and carry out attacks such as creating routing loops, forwarding packets through non-optimal paths, or selectively dropping packets, which results in disruption or degradation of the routing services [22].

Since mobile ad-hoc networks do not have any centrally administrated secure routers, chances are high that attackers can easily exploit or possibly disable a mobile ad-hoc network, if no security mechanism is adopted. Due to the dynamic nature of a mobile ad-hoc network, it suffers with frequent topology changes. Due to the absence of an infrastructure, each node in an ad-hoc network also acts as a router. For an ad-hoc network to exist, nodes have to be at least in the reception mode most of the time. Ad hoc

networks should be able to balance traffic load among nodes such that power constrained nodes can be put into a sleep mode while traffic is routed through other nodes [1].

In order to secure the Internet routing infrastructure, the two main planes of network functionality (i.e. control and data) must be protected. The control plane runs intra-domain and inter-domain routing protocols to build forwarding tables at routers. The data plane forwards (or drops) packets according to forwarding tables built by the control plane. Recently, considerable research and industrial efforts have addressed securing routing protocols, e.g. securing Border Gateway Protocol (BGP), the de facto glue for Internet inter-domain connectivity [10]. A secure version of BGP provides path and prefix attestations, which prevent propagation of illegitimate routes. Even in the presence of a secure control plane, however, a compromised router can disregard decisions made by the control plane and act autonomously and maliciously on the data plane. It can modify, drop, delay, reorder, mis-forward valid packets or permit otherwise prohibited packets. Such misbehavior would not be prevented by any secure routing protocol.

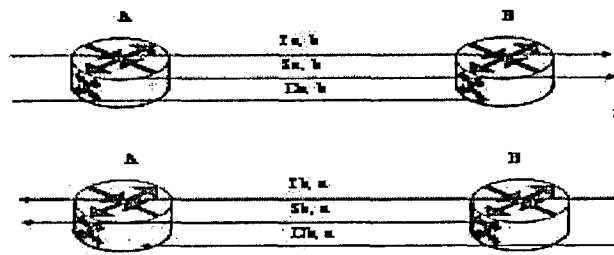
### 3.2 Problem Scenarios

The router is a primary component in the infrastructure of today's Internet, and is therefore an attractive target for attackers. If an attacker can gain control of a router, the attacker can disrupt communication by dropping or misrouting packets passing through the router. We present a protocol that detects and reacts to routers that drop or misroute packets [13].

As we know that in wired networks the central point of communication is the router. All the data is forwarded from router to router towards its final destination. And if any of the intermediate routers is compromised it starts misbehaving either by dropping, altering, delaying or corrupting the packets.

In context of graph theory, amongst the network each edge must be connected with each other so that data should travel accordingly. In directed graph each edge has a capacity, such that the amount of data flow along an edge does not exceed its capacity. A flow must suit the restriction that the amount of flow into a node must equal the amount of data flowing out of it, except when either it is a source, which has only outgoing flow(s), or sink, which has only incoming flow(s).

There are already many protocols being presented for the sake of analyzing incoming and outgoing packet flows for sake of load balancing and security purposes. However, WATCHERS is the one that specially talks about conservation of flow in the network which states that an input must either be engaged or sent on as an output. This is an attractive tool with which to analyze network protocol for security purposes. The functionality of the WATCHER'S algorithm is to detect malicious routers. In this regard, every router has to maintain a set of six vectors for each neighbor node. As shown in Figure 8, these vectors are based on either all the data that is passing through that router, or all information which are being sent by that router or the data which is intended for that router.



**Figure 8: Transit packet byte counter**

Each router performs this test on its neighbors, having received the counters from each neighbor's neighbors. The number of incoming packets minus packets destined for that router is compared to the number of outgoing packets minus packets originating with that router. If this difference exceeds some specified threshold, the tested router is diagnosed as malicious. In order to detect groups of malicious routers that conspire to hide their misbehavior, maintenance requirements are heavily increased [21].

According to A.Mizrak et al [15], the problem of detecting and removing compromised routers can be broken into three distinct sub problems: Traffic validation, distributed detection and response. His work is also based on Conservation of Flow principle. When the monitored traffic leaving one part of the network differs significantly from what is expected, anomalous behavior is detected. Synchronizing the collection of traffic information and distributing the results for detection purposes is the second problem in detecting compromised routers. For a detection protocol a simple approach is to use consensus to distribute the traffic information and then have each router use *TV* to implement its failure detector.

Khalid et al [21] have presented a method to evaluate an ideal conservation of flow within a wireless mesh network. They have implemented metrics to measure conservation of flow in terms of end-to-end data delivery. Additionally, if a node fails to send a packet forward, for any reason, it is being logged and the corresponding initiator of the packet is updated. Thus, the total packets sent were either delivered to the intended destination or dropped (which is traceable), but is conserved within the network; resulting in conservation of flow.

Due to lack of collusion detection mechanism, even in an ideal wireless environment, there is collusion and packet drop. However, there are a number of factors on which successful packet transmission relies, mainly the environmental factors and limited resources of wireless devices. Both the said factors result in congestion and packet drop. Packet drop in wireless network is may be because of two reasons; lossy channel or malicious node. Any of the reason consequently decreases QOS and network performance. In terms of lossy channel, there does not exist a cutting edge on the basis of which a lossy channel is identified. Once, identified we can later further investigate what was the underlying reason for the loss of packets; through different existing algorithms for specific purposes.

Detection of Malicious Node in MANET through Flow...

In [27] Faraz et al have presented a scheme that integrates or incorporates error rate in wireless channel for mesh network environment. The proposed idea is based on error rate in a wireless channel that can affect conservation of flow. Additionally, an ideal channel bit rate for wireless mesh network has also been implemented to achieve maximum conservation of flow. They have also presented a method to identify a lossy channel within a wireless mesh network.

One of the characteristics of an ad-hoc network implies that every node in an ad-hoc network volunteers to forward packets on behalf of other nodes. It is this node cooperation that holds an ad-hoc network together and makes the communication among the nodes possible. But such node cooperation cannot always be taken for granted. There could be situations in which a node might refuse to cooperate. Some of the reasons might be genuine while others indicate malicious or selfish intent. Some possible reasons for a node's non-cooperation include:

**Malicious Intent:** A node might want to disrupt the communication by misrouting, dropping or corrupting data packets. This scenario is very likely to occur in battlefield operations where the enemy nodes are always trying to disrupt the on going communication.

**Selfish Behavior:** Every node in an ad-hoc network must forward packets on behalf of others even if they are not of interest to it. So, a node might not be willing to expend its battery power on behalf of others.

- **Low Battery:** Nodes with reduced battery power might limit their activities to periodically transmitting and receiving emergency or high-priority messages to conserve the remaining battery power and thus extend their duration of operation.

Malicious node behavior is may be because of any of the security attacks listed in section 3.1. One of the common attacks among them is Blackhole attack. Some researchers discuss the vulnerabilities in Ad hoc routing protocols and the attacks that can be mounted. The AODV protocol is vulnerable to the well-known black hole attack. A black hole is a node that always responds positively with a RREP message to every RREQ, even though it does not really have a valid route to the destination node. Since a blackhole does not have to check its routing table, it is the first to respond to the RREQ in most cases. When the data packets routed by the source node reach the black hole node, it drops the packets rather than forwarding them to the destination node [3].

### 3.3 Focus of Research

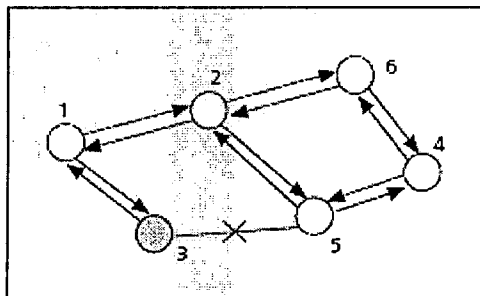
Security always implies the identification of potential attacks, threats and vulnerabilities of a certain system. Deng et al [24] discussed selected types of attacks that can easily be performed against a MANET. Attacks can be classified into *passive* and *active attacks*. A passive attack does not disrupt the operation of a routing protocol, but only attempts to discover valuable information by listening to routing traffic, which makes it very difficult to detect. An active attack is an attempt to improperly modify data, gain authentication,

or procure authorization by inserting false packets into the data stream or modifying packets transition through the network. Active attack can be further divided into external attacks and internal attacks. An *external attack* is one caused by nodes that do not belong to the network. An *internal attack* is one from compromised or hijacked nodes that belong to the network. Internal attacks are typically more severe, since malicious nodes already belong to the network as authorized parties. Therefore, such nodes are protected with the network security mechanisms and underlying services. Next, we describe some types of active attacks easily performed against a MANET in the network layer.

**Black hole:** In this attack, a malicious node uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. We provide a detailed description herein.

**Denial of service:** The DoS attack results when the network bandwidth is hijacked by a malicious node. It has many forms: the classic way is to flood any centralized resource so that the network

According to the original AODV protocol, any intermediate node may respond to the RREQ message if it has a fresh enough route, which is checked by the destination sequence number contained in the RREQ packet. This mechanism is used to decrease the routing delay, but makes the system a target of a malicious node. The malicious node easily disrupts the correct functioning of the routing protocol and makes at least part of the network crash. For example, node 1 wants to send data packets to node 4 in Figure 9, and initiates the route discovery process.



**Figure 9: The Blackhole Problem**

We assume node 3 to be a malicious node with no fresh enough route to destination node 4. However, node 3 claims that it has the route to the destination whenever it receives RREQ packets, and sends the response to source node 1. The destination node and any other normal intermediate nodes that have the fresh route to the destination may also give a reply. If the reply from a normal node reaches the source node of the RREQ first, everything works well; but the reply from malicious node 3 could reach the source node first, if the malicious node is nearer to the source node. Moreover, a malicious node does not need to check its routing table when sending a false message; its response is more likely to reach the source node first. This makes the source node think that the route discovery process is complete, ignore all other reply messages, and begin to send data packets. As a result, all the packets through the malicious node are simply consumed or



lost. The malicious node could be said to form a black hole in the network, and we call this the black hole problem. In this way the malicious node can easily misroute a lot of network traffic to itself, and could cause an attack to the network with very little efforts on its part [24].

The existing techniques for detection of malicious behavior in both wired and wireless scenario are deficient in some or other way. Blackhole attack adversely affects the performance of a network working under AODV protocol. Author have focused to implement active and passive blackhole attacks in AODV protocol as AODV routing protocol is vulnerable to this type of attack and have also designed algorithms which can identify these blackhole attacks in AODV protocol.

### **3.4 Summary**

It is essential to detect malicious behavior of nodes and take suitable action to pass up any unnecessary wastage of network resources like bandwidth, battery power, etc which are utilized repeatedly to retransmit the packets and to exchange control information. Misbehaving nodes slow down the efficient functioning of the ad-hoc network. Detecting malicious behavior is the very first step in handling malicious nodes. Proposed approach will be discussed in the next chapter.

# **CHAPTER 04**

## **SYSTEM DESIGN**

## Chapter 4: System Design

This section briefly illustrates the reference architecture and design methodology used in my work. Author will discuss in detail the AODV protocol implementation details.

### 4.1 Design Requirements

Proactive routing protocols are derived from legacy Internet distance-vector and link-state protocols. They maintain tables that store routing information. And for any change in network they trigger propagating updates throughout the network in order to maintain a consistent network view. This can cause substantial overhead affecting bandwidth utilization, throughput as well as power usage. The advantage is that routes to any destination are always available without the overhead of a route discovery but such protocols cannot perform properly when the mobility rate in the network is high or when there are a large number of nodes in the network. Protocols in this category differ in the number of tables they contain as well as on the details of how they are updated. For example, nodes in Destination-Sequenced Distance Vector (DSDV) algorithm maintain route information to every other node in the network. As the network status changes full updates are exchanged among all nodes. The Wireless Routing Protocol (WRP) [18] localizes the updates to the immediate neighbors. When a new node A moves into range of a node B and a hello message is received from it, A is added to B's routing table and sends a full copy of the table. When a link fails, a node sends updates to its neighbors. The Cluster Gateway Switch Routing (CGSR) [18] protocol reduces the size and amount of information propagation by having each cluster of nodes elect a cluster head. Network-wide information is only exchanged among the cluster heads. While the amount of information propagation is reduced, this results in inefficient routes. The Fisheye State Routing Protocol has been recently suggested, this differ from others in that the update frequency is inversely related to the distance between any two nodes.

Reactive (On-Demand) routing protocols are characterized by a path discovery mechanism that is initiated when a source needs to communicate with a destination that it does not know how to reach. The Route Discovery is usually in the form of query flood. Generally, on-demand routing requires less over-head than table-driven routing; but it incurs a path discovery delay whenever a new path is needed [18].

### 4.2 Reference Architecture

There are attacks that target some particular routing protocols, such as DSR, or AODV. More sophisticated and subtle routing attacks have been identified in recent published papers, such as the black hole (or sinkhole), Byzantine, and wormhole attacks.

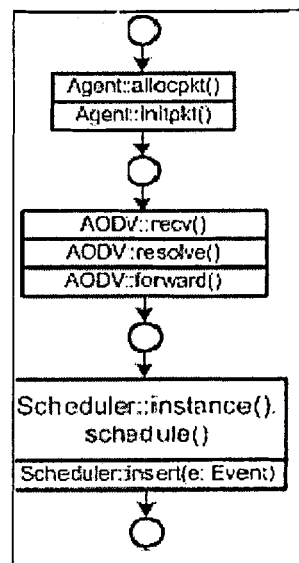
#### 4.2.1 AODV Implementation

In mobile and ad-hoc networks there are several routing protocols available; e.g. DSR, DSDV or AODV. Within NS2 there are implementations for some of the available routing protocols. The AODV implementation in NS2 is the basis of the presented work

and is therefore explained in this section. This AODV implementation is not RFC compliant; e.g. there is no implementation for the blacklist defined in AODV [6]. The AODV implementation mainly consists of two functional units.

- Mechanism to detect the route through a network
- Mechanism to forward packets within this route

Figure 10 is a simple packet flow within a single node in the simulation. An agent consists to specific node. This agent allocates the packet and initializes the content of the packet. Afterwards the packet is handled by the AODV routing mechanism. Initially, the routing handler receives this packet and processes it locally. Afterwards the route for the packet is detected. When a suitable route is found, the packet is forwarded to the next hop within this route.



**Figure 10: Packet Flow**

The call hierarchy in AODV is as follows:

- The `recv()` method is the first method being called.
- `Recv()` further calls first the `resolve()` and then the `forward()` method.
- The `forward()` is the method that calls the `Scheduler::scheduler()` method, which schedules and queues all the packets accordingly.

Figure 11 shows the initialization process of the AODV implementation in a specific simulation. This simulation consists of three hosts that are located in a line, so that Host 0 and Host 2 are not able to communicate directly. Within this simulation one TCP packet is send from Host 0 to Host 2 and the reply is send from Host 2 to Host 0. AODV resolves the route to Host 2 by sending a special request packet. This packet is broadcasted as long as the destination of the packet replies. The reply is also broadcasted through the network. Each forwarded request packet also triggers the intermediate nodes to send additional request packets. Therefore in the start of the simulation there is a high amount of AODV related network traffic. After that phase, there is only a few additional

AODV network traffic required. Detected routes are cached within this implementation, so that not every send packets requires a related AODV route request. After the detection of the route, the packet is send.

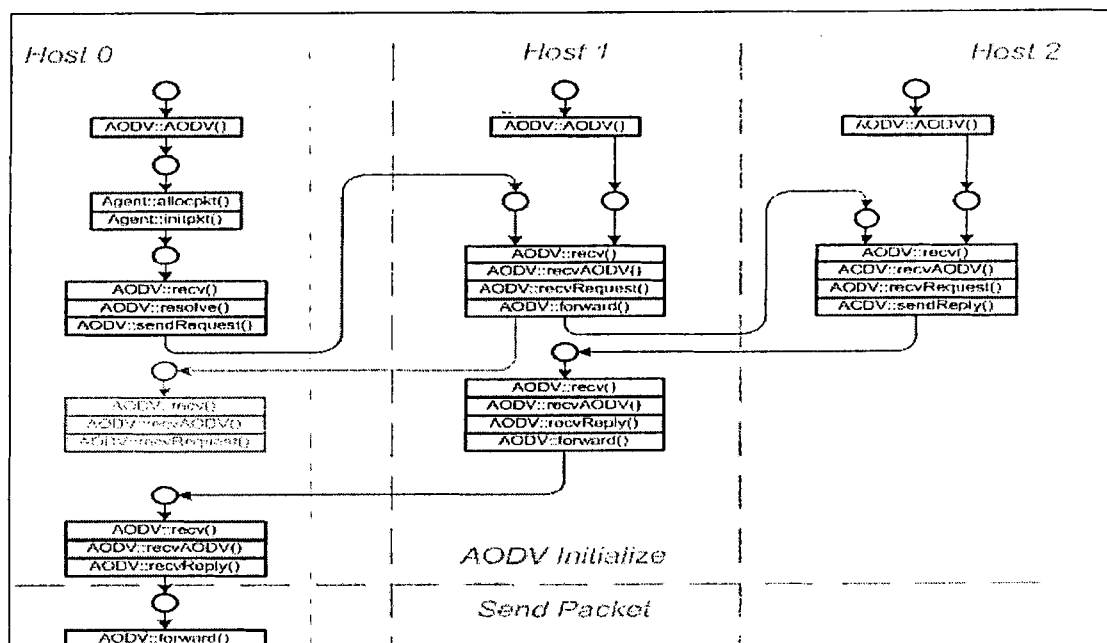


Figure 11: AODV Initialization

Figure 12 shows the process with the related calls. The packet is forwarded by each node on the route. In the forward method the Scheduler is called, to transmit the packets. The resolve() method returns immediately, due to a cached response for the request.

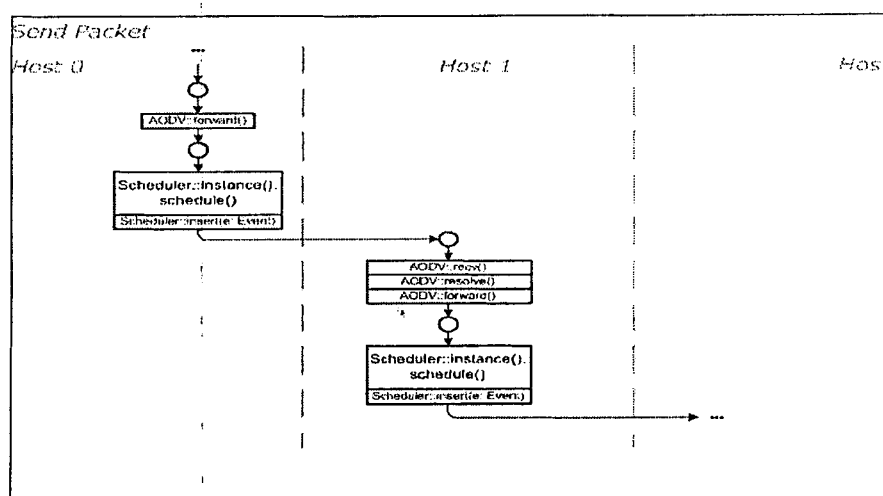
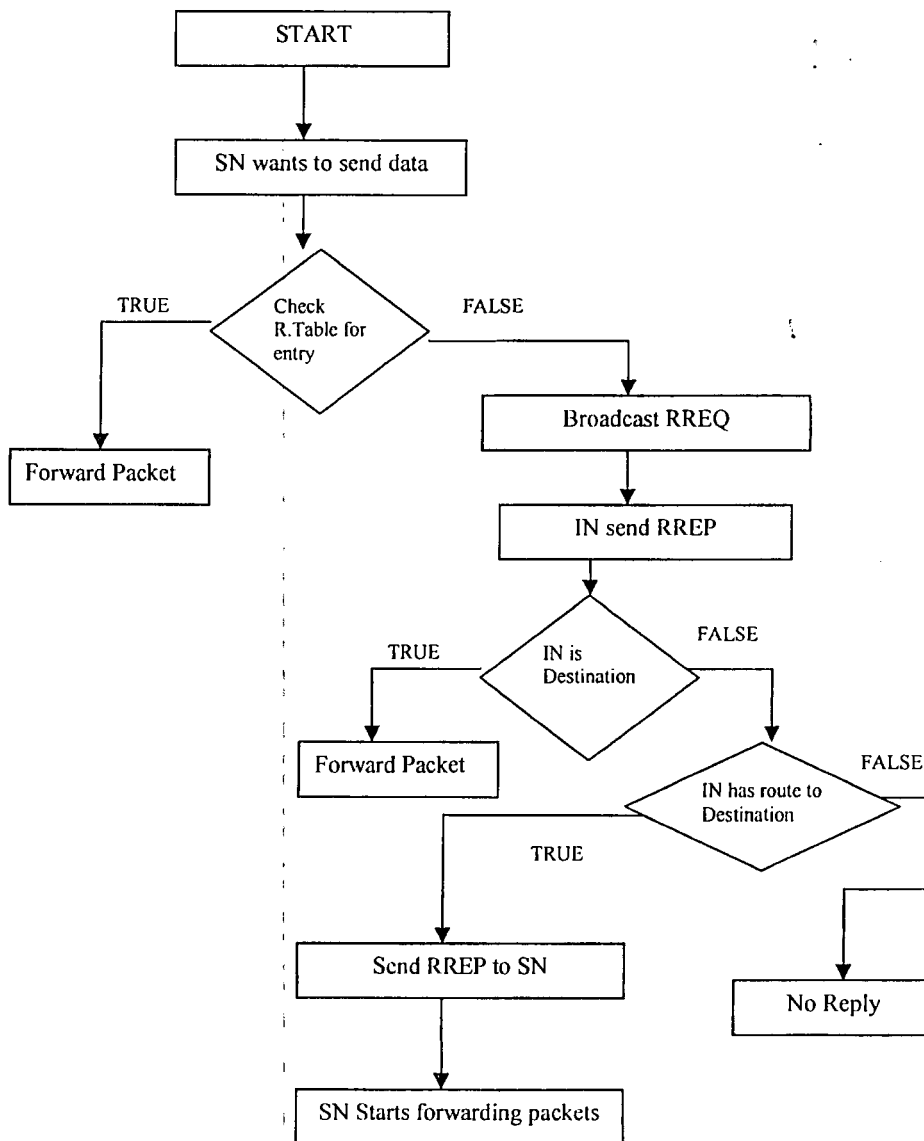


Figure 12: Transmission Packet Flow



**Figure 13: Packet Receiving in AODV**

### 4.3 Design Methodology

This section will be based on the description of my proposed methodology for identifying malicious node by introducing wait and count mechanism. In this mechanism, the source node after broadcasting RREQ message waits for two RREP to come. When source node receives two RREP it calculates the difference between the sequence numbers of the two RREP. A cache table is used to store the information like packet number, path, sequence number and difference of sequence numbers. I will also restrict a node to send RREP to same RREQ message by using a counter with threshold equal to 3. When a node sends more than three reply with same sequence number to same RREQ it is suspected as malicious. Source node will not use this node as intermediate node anymore.

#### 4.3.1 Algorithm

##### Notations:

- *SN: Source Node*
- *DN: Destination Node*
- *IN: Intermediate Node*
- *NH: Next Hop*
- *RREQ: Route Request*
- *RREP: Route Reply*

##### Pseudo-Code:

1. SN broadcasts RREQ msg
2. SN receives RREP msg
3. IF (RREP is from DN) {
4.     Route Data Packets (Secure Route)
5. }
6. ELSE {
7.     SN waits for second RREP
8.     Receive second RREP from an IN
9.     SN calculates the difference between the Sequence numbers of the two RREPs
10.     Save the difference in Cache
11.     also save in Cache (Packet #, NH, Seq #, Difference of Seq #)
12.     Save the entry in the Counter
13.     IF (IN sends a RREP msg with Highest seq #) {
14.         Check Counter
15.         IF (value of Counter > 3) {
16.             IN is a blackhole node
17.             Insecure Route
18.         }
19.         ELSE {
20.             Secure Route
21.             Route Data Packets
22.         }
23.     }

## 4.4 UML Diagrams

### 4.4.1 Use case Diagram

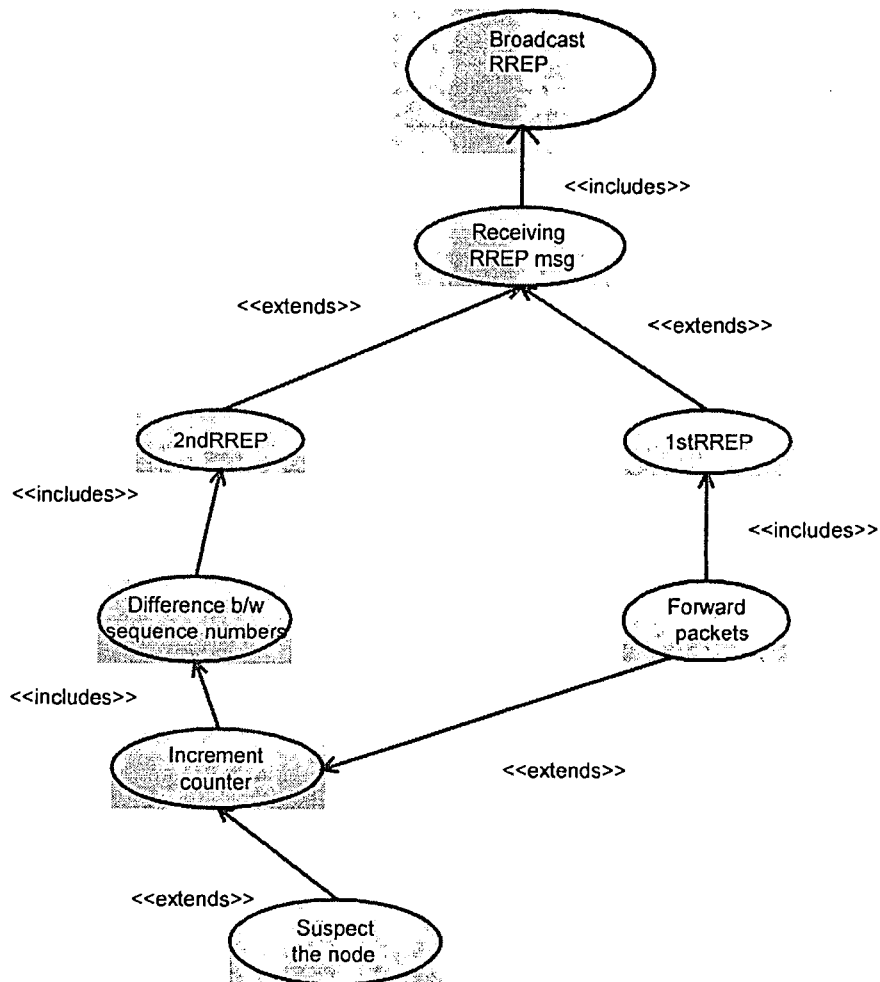


Figure 14: Usecase Diagram



#### 4.4.2 Class Diagram

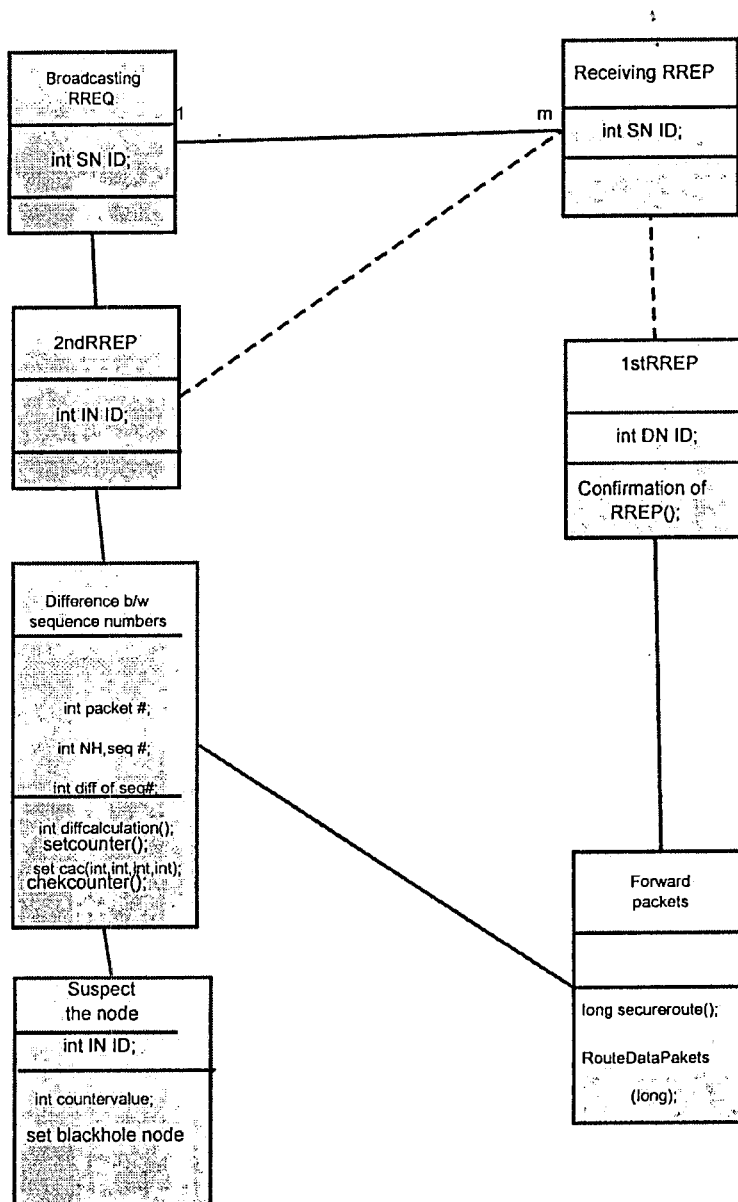


Figure 15: Class Diagram

### 4.4.3 Sequence Diagram

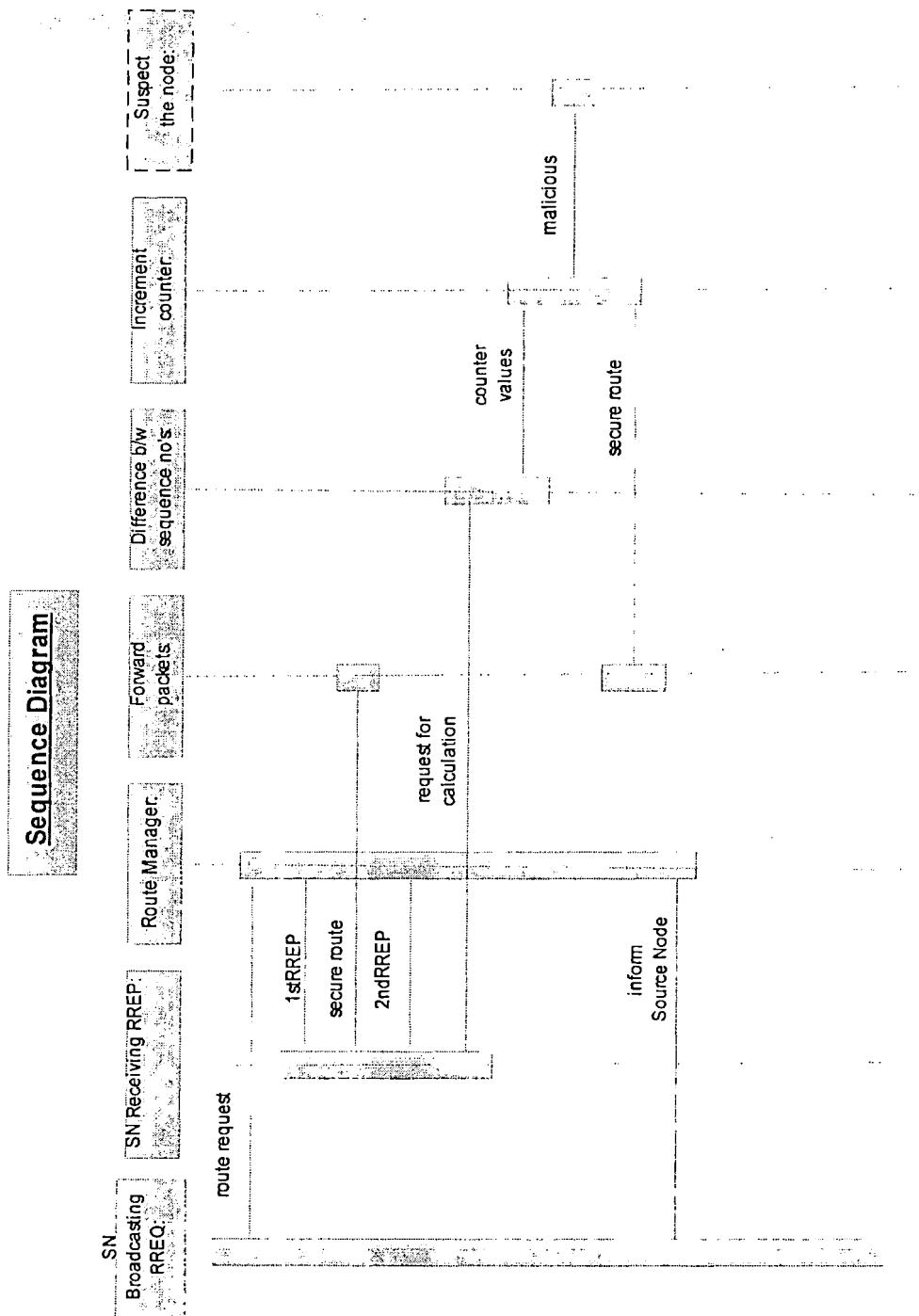


Figure 16: Sequence Diagram

#### 4.5 Summary

In this section author has presented the design architecture, UML Diagrams and designed algorithm. The above are discussed in the context of the AODV routing protocol, which is the basis of the implementation of the packet routing. Details related to implementation are listed in next chapter.

# **CHAPTER 05**

## **IMPLEMENTATION**

## Chapter 5: Implementation

To achieve the goal author has simulated the wireless ad-hoc network scenarios which include Active Black Hole attack using NS Network Simulator [29] program and Passive Blackhole attack using OMNeT++ [30]. To simulate active blackhole attack in a wireless ad-hoc network one has implemented highest sequence number attack that drops data packets after attracting them to itself and to implement passive blackhole attack, selfish node behavior which drops the packets on the basis of probability mechanism in order to save its battery power has been implemented. In this chapter NS, OMNeT++ and author's contribution is presented.

### 5.1 Network Simulator 2

NS [29] is an event driven network simulator program, developed at the University of California Berkley, which includes many network objects such as protocols, applications and traffic source behavior. The NS is a part of software of the VINT project that is supported by DARPA since 1995.

At the simulation layer NS uses OTcl (Object oriented Tool Command Language) programming language to interpret user simulation scripts. OTcl language is in fact an object oriented extension of the Tcl Language. The Tcl language is fully compatible with the C++ programming language. At the top layer, NS is an interpreter of Tcl scripts of the users, they work together with C++ codes.

An OTcl script written by a user is interpreted by NS. While OTcl script is being interpreted, NS creates two main analysis reports simultaneously. One of them is NAM (Network Animator) object that shows the visual animation of the simulation. The other is the trace object that consists of the behavior of all objects in the simulation. Both of them are created as a file by NS. Former is .nam file used by NAM software that comes along with NS. Latter is a ".tr" file that includes all simulation traces in the text format.

#### 5.1.1 Class Hierarchies

NS2 is a network simulator written in C++ and TCL. The NS2 front end is an OTCL interpreter. The simulation descriptions are scripts written in OTCL. The front end interprets this files and the back end simulates the described scenario. The NS2 simulation output files are generated within this process. Within NS2 there are two separate class hierarchies; the Compiled Hierarchy implemented in C++ and the Interpreted Hierarchy implemented in TCL. Furthermore, there are classes that realize the processing of the simulation and there are classes that belong to the simulation itself. Within this document objects based on classes that realize the processing of the simulation are called Processing Objects. The Objects based on the classes that belong to the simulation itself will be called Simulation Objects. Most of the Processing Objects are implemented in the Compiled Hierarchy. For most of the Simulation Objects, there is 1:1 relationship between a base class in the Compiled Hierarchy and a base class in the Interpreted Hierarchy. The implementation of this classes is often separated in two parts,

one located in the Compiled Hierarchy and one located in the Interpreted Hierarchy. This makes modification of Simulation Objects very complicated.

The separation in two class hierarchies is reasonable for dealing with the trade-off between runtime performance and iteration time; which means the required time to change the simulation descriptions and execute another simulation. The iteration time can be minimized by utilization of simple scripting language with simple syntax. The runtime performance can be optimized by using a compiled language. Therefore the two approaches are combined within NS2 by using OTCL as front end and C++ as back end.

### 5.1.2 TCL Linkage

The TCL Linkage is the communication interface between the implementation of the Simulation Objects on the two class hierarchies; the Compiled Hierarchy and the Interpreted Hierarchy. It realizes the communication between the implementation parts; e.g. the transmission of messages from Compiled Hierarchy into Interpreted Hierarchy and vice versa. The TCL Linkage consists of six important classes:

- Tcl
- TclObject
- TclClass
- TclCommand
- EmbeddedTcl
- InstVar

### 5.1.3 Scheduler

The Scheduler is an entity within NS2 responsible for packet transmission. The scheduler is a single instance within the NS2 runtime that collects and transmits messages. Therefore the Scheduler class is realized as static class with static methods. The Scheduler consists of control unit and a so called packet queue. This packet queue is accessible through the nodes within the simulation. Each node can put packets into the packet queue. The Scheduler takes each packet from the queue and dispatches the packet to its destination. The knows the destination of a packet during runtime. The destination of the packet is saved within a special pointer in the packet structure. This pointer is a reference to a NSObject instance where the method `recv()` is called within `Scheduler::dispatch()`.

## 5.2 OMNeT++

OMNeT++ [30] is an object-oriented modular discrete event network simulator. The simulator can be used for:

- Traffic modeling of telecommunication networks
- Protocol modeling
- Modeling queuing networks

- Modeling multiprocessors and other distributed hardware systems
- Validating hardware architectures
- Evaluating performance aspects of complex software systems
- Modeling any other system where the discrete event approach is suitable.

An OMNeT++ model consists of hierarchically nested modules. The depth of module nesting is not limited, which allows the user to reflect the logical structure of the actual system in the model structure. Modules communicate through message passing. Messages can contain arbitrarily complex data structures. Modules can send messages either directly to their destination or along a predefined path, through gates and connections.

Modules can have their own parameters. Parameters can be used to customize module behavior and to parameterize the model's topology. Modules at the lowest level of the module hierarchy encapsulate behavior. These modules are termed simple modules, and they are programmed in C++ using the simulation library. OMNeT++ simulations can feature varying user interfaces for different purposes: debugging, demonstration and batch execution. Advanced user interfaces make the inside of the model visible to the user; allow control over simulation execution and to intervene by changing variables/objects inside the model. This is very useful in the development/debugging phase of the simulation project. User interfaces also facilitate demonstration of how a model works.

The simulator as well as user interfaces and tools are portable: they are known to work on Windows and on several UNIX flavors, using various C++ compilers. OMNeT++ also supports parallel distributed simulation. OMNeT++ can use several mechanisms for communication between partitions of a parallel distributed simulation, for example MPI or named pipes. The parallel simulation algorithm can easily be extended or new ones plugged in. Models do not need any special instrumentation to be run in parallel – it is just a matter of configuration. OMNeT++ can even OMNeT++ Manual – Introduction be used for classroom presentation of parallel simulation algorithms, because simulations can be run in parallel even under the GUI which provides detailed feedback on what is going on.

### 5.2.1 Hierarchical Modules

An OMNeT++ model consists of hierarchically nested modules, which communicate by passing messages to each another. OMNeT++ models are often referred to as networks. The top level module is the system module. The system module contains submodules, which can also contain submodules themselves. The depth of module nesting is not limited; this allows the user to reflect the logical structure of the actual system in the model structure.

Modules that contain sub-modules are termed compound modules, as opposed simple modules which are at the lowest level of the module hierarchy. Simple modules contain

the algorithms in the model. The user implements the simple modules in C++, using the OMNeT++ simulation class library.

### 5.2.2 Module Types

Both simple and compound modules are instances of module types. While describing the model, the user defines module types; instances of these module types serve as components for more complex module types. Finally, the user creates the system module as an instance of a previously defined module type; all modules of the network are instantiated as sub-modules and sub-submodules of the system module. When a module type is used as a building block, there is no distinction whether it is a simple or a compound module. This allows the user to split a simple module into several simple modules embedded into a compound module, or vice versa, aggregate the functionality of a compound module into a single simple module, without affecting existing users of the module type. Module types can be stored in files separately from the place of their actual usage. This means that the user can group existing module types and create component libraries.

Modules communicate by exchanging messages. In an actual simulation, messages can represent frames or packets in a computer network, jobs or customers in a queuing network or other types of mobile entities. Messages can contain arbitrarily complex data structures. Simple modules can send messages either directly to their destination or along a predefined path, through gates and connections.

The “local simulation time” of a module advances when the module receives a message. The message can arrive from another module or from the same module (self-messages are used to implement timers). Gates are the input and output interfaces of modules; messages are sent out through output gates and arrive through input gates.

Each connection (also called link) is created within a single level of the module hierarchy: within a compound module, one can connect the corresponding gates of two sub-modules, or a gate of one sub-module and a gate of the compound module. Due to the hierarchical structure of the model, messages typically travel through a series of connections, to start and arrive in simple modules. Such series of connections that go from simple module to simple module are called routes. Compound modules act as ‘cardboard boxes’ in the model, transparently relaying messages between their inside and the outside world.

### 5.2.3 Modeling of Packet Transmissions

Connections can be assigned three parameters, which facilitate the modeling of communication networks, but can be useful in other models too: propagation delay, bit error rate and data rate, all three being optional. One can specify link parameters individually for each connection, or define link types and use them throughout the whole model. Propagation delay is the amount of time the arrival of the message is delayed by when it travels through the channel.

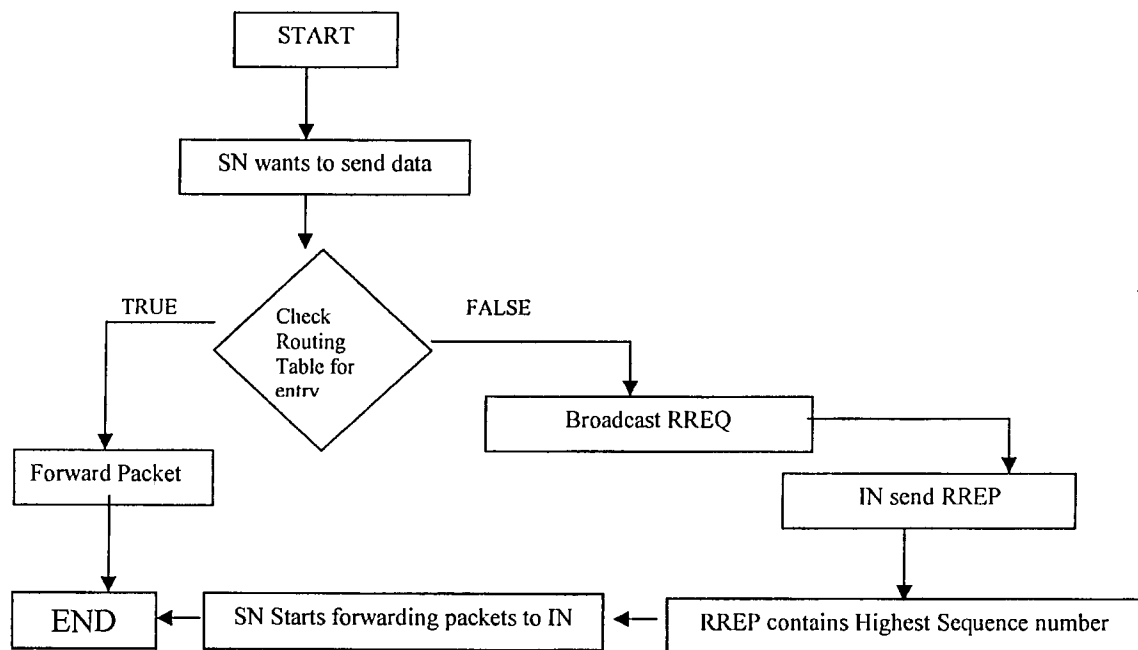


Bit error rate specifies the probability that a bit is incorrectly transmitted, and allows for simple noisy channel modeling. Data rate is specified in bits/second, and it is used for calculating transmission time of a packet. When data rates are in use, the sending of the message in the model corresponds to the transmission of the first bit, and the arrival of the message corresponds to the reception of the last bit. This model is not always applicable, for example protocols like Token Ring and FDDI do not wait for the frame to arrive in its entirety, but rather start repeating its first bits soon after they arrive – in other words, frames “flow through” the stations, being delayed only a few bits. If you want to model such networks, the data rate modeling feature of OMNeT++ cannot be used.

### 5.2.4 Parameters

Modules can have parameters. Parameters can be assigned either in the NED files or the configuration file omnetpp.ini. Parameters may be used to customize simple module behaviour, and for parameterizing the model topology. Parameters can take string, numeric or boolean values, or can contain XML data trees. Numeric values include expressions using other parameters and calling C functions, random variables from different distributions, and values input interactively by the user. Numeric-valued parameters can be used to construct topologies in a flexible way. Within a compound module, parameters can define the number of sub-modules, number of gates, and the way the internal connections are made.

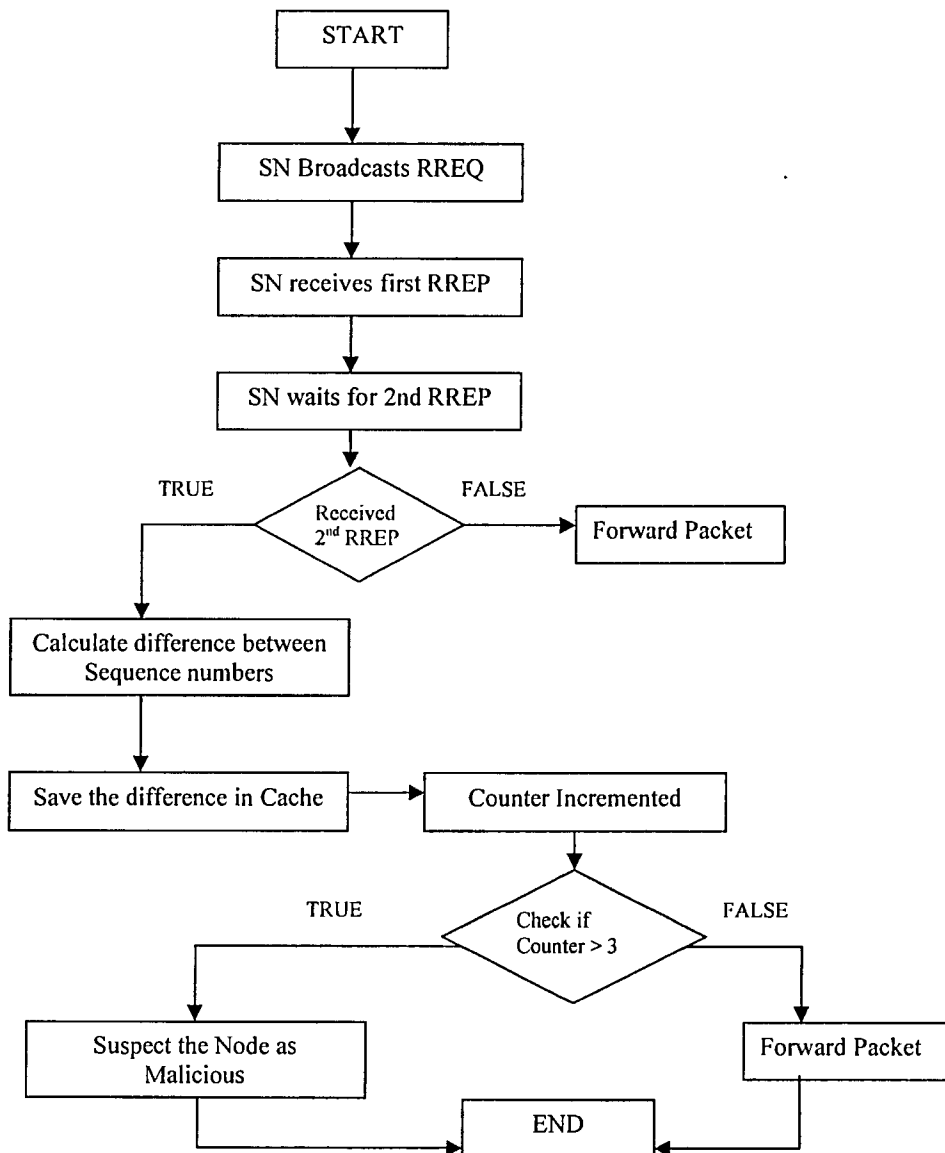
### 5.3 Active Blackhole Attack Implementation



**Figure 17: Flow Chart for Active Blackhole Attack Implementation**

In this mechanism when source node wants to start communication it first checks its routing table if there is a fresh enough route to the destination source node will use it. In other case source node will broadcasts RREQ message. As there is a blackhole node in the network which uses highest possible sequence number (listed in AODV RFC) in RREP message. Because of this highest value of sequence number source node starts sending data to this node thinking it as destination or best intermediate node to the destination. Blackhole node becomes successful as it will receive all the packets and will start dropping the packets.

#### 5.4 Detection Model for Active Blackhole Attack



**Figure 18: Flow Chart for Active Blackhole Attack Detection Model**

In this section, a methodology for identifying malicious node by introducing wait and count mechanism is proposed. In this mechanism, the source node after broadcasting RREQ message waits for two RREP to come. When source node receives two RREP it calculates the difference between the sequence numbers of the two RREP. If there is some blackhole node in the network then this difference would be in millions. A cache table is used to store the information like packet number, path, sequence number and difference of sequence numbers. One will also restrict a node to send RREPs to same RREQ message by using a counter with threshold equal to 3. When a node sends more than three replies with same sequence number to same RREQ it is suspected as malicious. Source node will not use this node as intermediate node anymore.

#### 5.4.1 Mathematical Model

Let  $N$  be the set of nodes  $\{n_1, n_2, n_3, \dots, n_i, n_{i+1}, \dots, n_n\}$  and

SN= source node

DN= destination node

IN= intermediate node

NH= next hope

RREQ= route request

RREP= route reply

pk= packet

Frwd = Route Data Packet

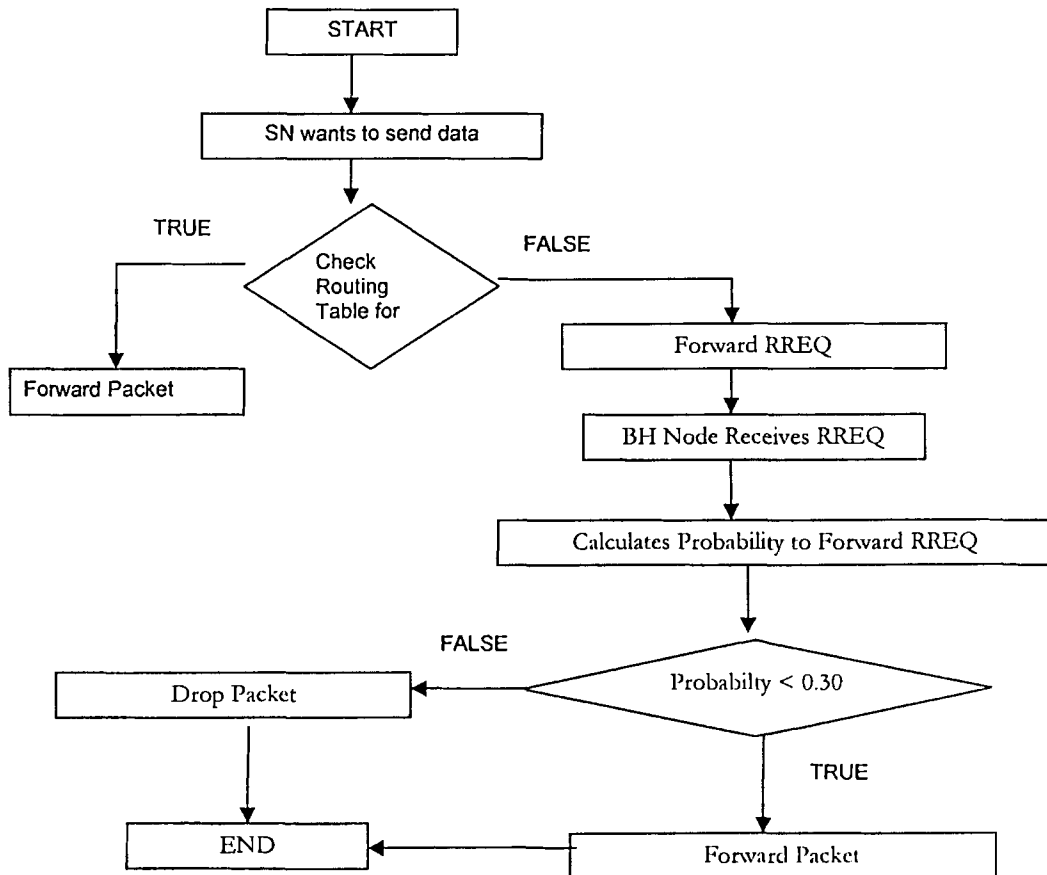
diff= difference between sequence numbers

- 1)  $\sum_1^k \text{RREQ (SN)}$  where  $n \in N$  (Natural number) and  $k \leq N$
- 2)  $\sum_1^k \text{RREQ}$  where  $k \leq N$   
if  $\begin{cases} \text{Frwd (SR)}, & (\text{RREP})^i = \text{DN} \\ \sum_{i+1}^k (\text{RREP}), & \text{wait} \end{cases}$
- 3)  $\text{diff} = |pk_i - pk_j|$
- 4)  $\text{CacheQueue} = \{ \text{diff}_k \wedge (\text{Pk\#}, \text{NH}, \text{Seq\#}, \text{diff})$
- 5) if { check (counter,  $\sum_1^k \text{RREP (IN)} > \text{Seq\#}$
- 6) if  $\begin{cases} pk = \text{insecure}, & \text{counter} > 3 \\ pk = \text{secure} \wedge \text{Frwd (pk)}, & \text{counter} \leq 3 \end{cases}$

#### 5.5 Passive Blackhole Attack Implementation

In this mechanism when source node wants to start communication it first checks its routing table if there is a fresh enough route to the destination source node will use it. In other case source node will broadcasts RREQ message. As there is a passive blackhole node in the network which is not willing to spend its battery power in transmitting packets to other nodes. When it receives RREQ or data packet it will

calculate the probability for forwarding this packet. If the probability is greater than the threshold value the node will drop the packet otherwise it will forward it. In this simulation scenario, passive blackhole node becomes successful as designated passive blackhole node is acting as bridge node between the two logical portions of a network. So it will receive all the packets and will start dropping the packets selectively.

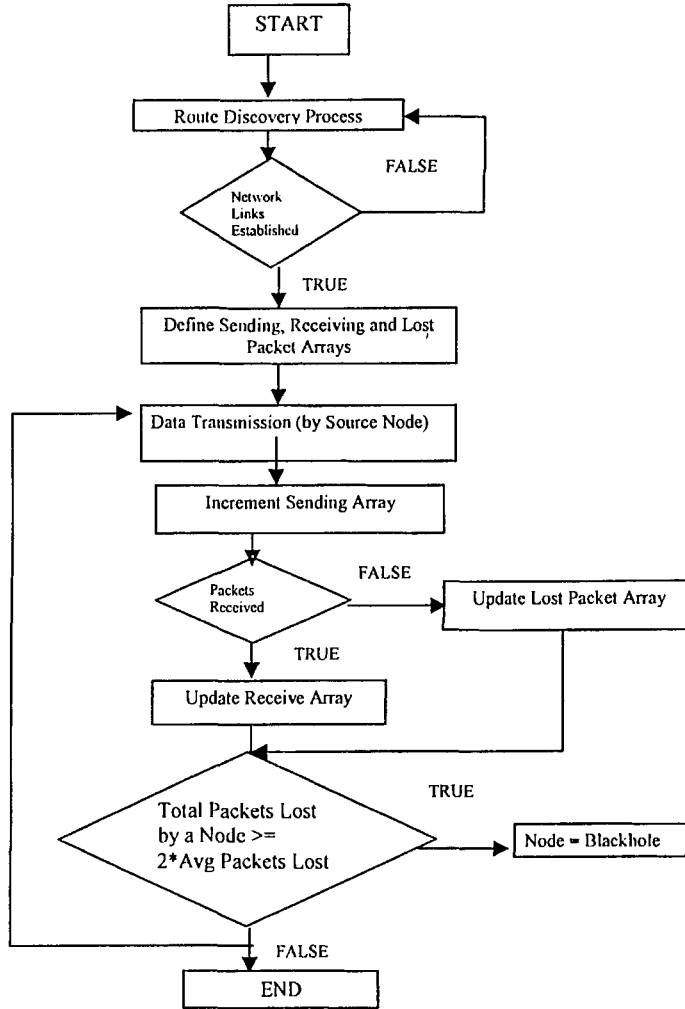


**Figure 19: Flow chart for Passive Blackhole Attack Implementation**

## 5.6 Detection Model for Passive Blackhole Attack

In this section, a detection model for passive blackhole attack in Adhoc networks working under AODV protocol is proposed. When route discovery process is initiated and links are successfully established then three types of matrices are defined to keep track of packets. These matrices include sending, receiving and lost packet arrays. When a source node starts transmitting packets, the sending array will be incremented. When packets will reach the destination the corresponding receiving array is updated. But if a packet originated at a source does not reach the correct destination then it will be traced in lost array. Periodically all these matrices are checked to identify any malicious node in the network. For the sake of simplicity, all packets dropped by a malicious node are

currently incorporated in Lost Packets, against that particular node. So, if the total packets dropped by any node are greater than equal to twice the average packets dropped by all nodes becomes true for any node, one can state that the node truly is identified as the blackhole node.



**Figure 20: Flow Chart of Passive Blackhole Attack Detection Model**

### 5.6.1 Mathematical Model

In a wireless network, consisting of some nodes, we can state:

$$Nodes \{a, b, c, d, e, f, g, h\}$$

Once, the network is established, nodes start to communicate packets to and fro. We can generally say that:

$$Source \in Nodes$$

$$Destination \in Nodes$$

Where  $Source \neq Destination$

Since, within an adhoc network the nodes cooperate with each other to transfer data packets from one end of the network to other, therefore every node also acts as a router or intermediate hop within a defined path for successful packet delivery. Thus,

$$IntermediateNodes = \{I_k \in Nodes \mid I_k \neq Source \wedge I_k \neq Destination \wedge I_k > 0\}$$

The defined path can be represented as:

$$Route = \{(x, y, z) \mid x \in Nodes \wedge y \in IntermediateNodes \wedge z \in Nodes \wedge x \cup \{y\} \cup \{z\}\}$$

In terms of communication amongst the nodes, the number of packets becomes the measure of transmission. In this regard, we denote:

$$TotalPackets = P_T$$

$$SendingPackets = \{P_s \mid s = Source \wedge (0 < P_s \leq TPackets)\}$$

$$RecievingPackets = \{P_r \mid r = Desitnation \wedge (0 < P_r \leq TPackets)\}$$

The  $P_s$  is the number of packets sent by all nodes in a transaction, whereas  $P_r$  is the

number of packets received by the destined nodes. Ideally, it becomes  $\sum P_s = \sum P_r$ .

However, it is always not the case in real world and packet drops are quite frequent. Thus, any packet dropped by any node needs to be reported, so that it can be re-sent. This is done through:

$$LostPackets = \{P_l \mid r = Desitnation \wedge s = Source \wedge P_l \notin P_r\}$$

So far things are easily traceable, but when any node starts forgery then the situation becomes complex; as it drops packets but does not report them in any of the sets: Sending Packets, Receiving Packets and Lost Packets. Therefore, need arises that we introduce the method of counting intermediate packets.

$$IntermediatePackets = \{P_i \mid i \in IntermediateNode \wedge P_i \in P_s \wedge P_i \notin P_r \wedge P_i \in Route\}$$

$$(P_s = P_r + P_l) \Leftrightarrow \{(P_s \neq P_r) \wedge (P_s \rightarrow P_i) \rightarrow (P_i = P_l)\}$$

$$\neg(P_s = P_r + P_l) \rightarrow (P_s = P_r + P_l + P_{bh})$$

For the sake of simplicity, all packets dropped by a malicious node are currently incorporated in Lost Packets, against that particular node.

$$P_{bh} \subseteq P_l$$

Thus, a black node in the network can be identified as follows:

$$P_{ln} \in P_l \wedge P_{ln} \in P_i \wedge \sum P_{ln} \geq 2 * Avg(P_l)$$

i.e. Packet lost by any node is a subset of the set of packet-lost, where packet lost by the node is also a subset of the set of intermediate packets; and if the total packets dropped by any node is greater than equal to twice the average packets dropped by all nodes. If it becomes true for any node, we can state that any node meeting the above criteria, truly is identified as the blackhole node.

## 5.7 Summary

Author has discussed in detail the environment of Network Simulator 2 and OMNET ++. And then the default implementation of AODV protocol in NS2. After that algorithms and the flow charts are presented. Simulation results and scenarios will be covered in the next section.

# CHAPTER 06

## SIMULATION AND RESULTS



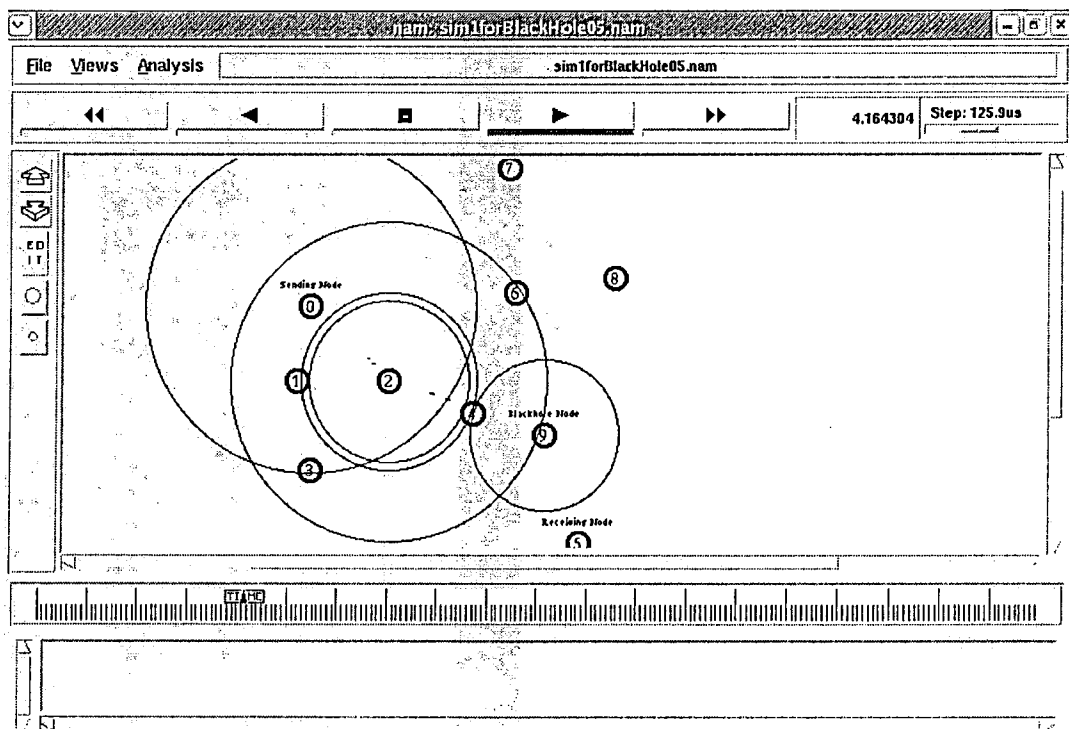
## Chapter 6: Simulation and Results

### 6.1 Simulation Scenarios in Network Simulator 2

In order to test system model for the behavior of black hole behavior three different simulation scenarios are performed in ns-2.31. The three scenarios differ in Blackhole Node Position and Sender Receiver Node Positions.

#### 6.1.1 Scenario I

For a Network of 10 nodes Simulation of 20 Seconds was performed. Source Node is 0, Destination Node is 5 and Blackhole Node is 9. From source to destination there are two intermediate nodes 2 and 4. The topology of the simulation scenario is shown in figure 21.



**Figure 21: Snapshot of Scenario I Simulation**

A total of 120 packets were sent by Node 0. Node 2 being the intermediate node forwarded all the packets to Node 9, this is shown using figure 22. Node 9 received all the packets initiated from source node 0, it can be seen in figure 23. Node 9 acts as a active blackhole node and thus drops all the packets it received as shown in figure 24. Throughput of dropping packets at Blackhole node is shown in Figure 25.

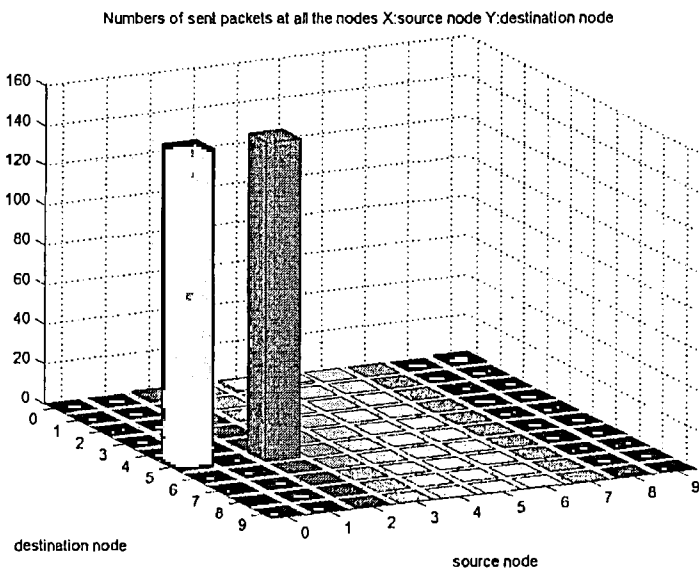


Figure 22: Number of Packets Sent at all the Nodes

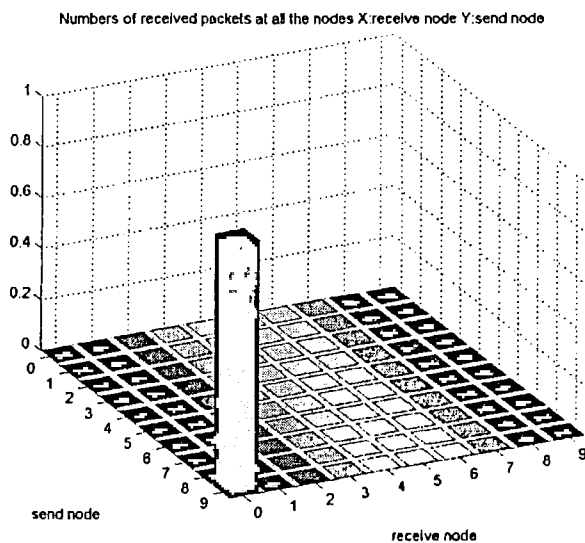
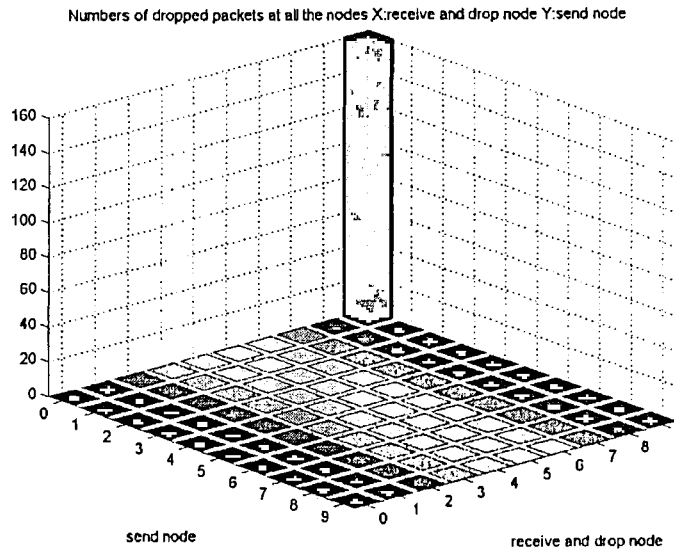
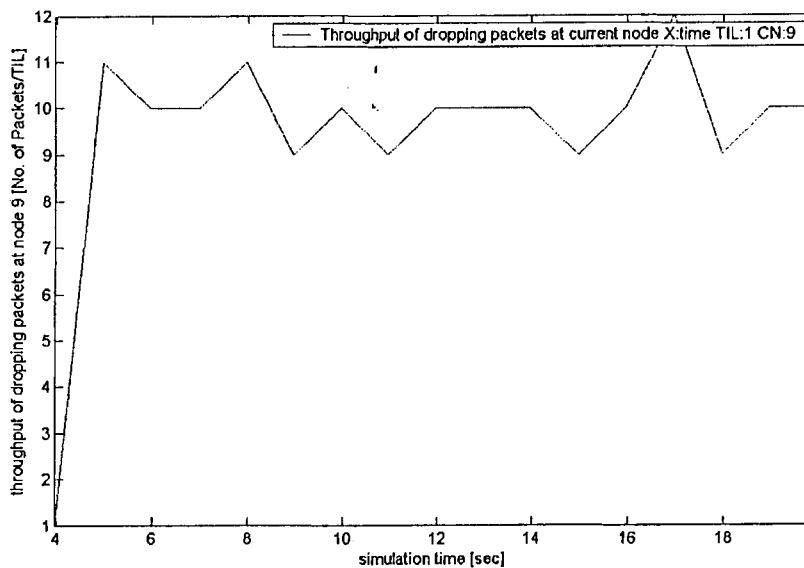


Figure 23: Number of Packets Received at all the Nodes



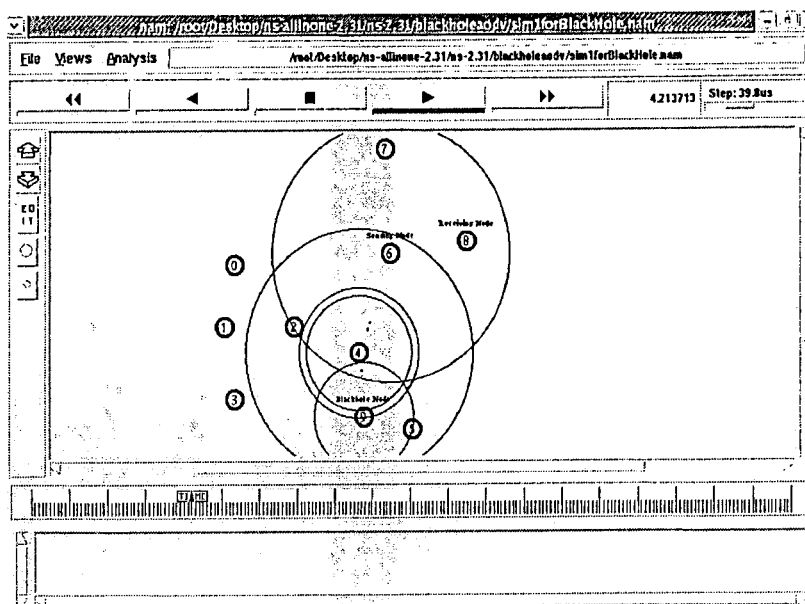
**Figure 24: Number of Packets Dropped at all the Nodes**



**Figure 25: Throughput of Dropping Packets at Blackhole Node (Node 9)**

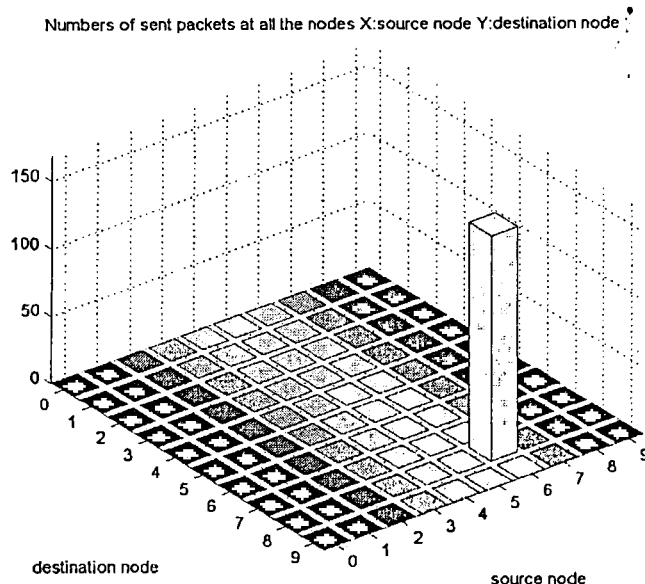
### 6.1.2 Scenario II

For a Network of 10 nodes Simulation of 20 seconds was performed. Source Node is 6, Destination Node is 8 and Blackhole Node is 9. Source and Destination are adjacent nodes. The topology of the simulation scenario is shown in figure 26.



**Figure 26: Snapshot of Simulation Scenario II**

A total of 100 packets were sent by Source Node 6, this is shown using figure 27. Node 9 received all the packets initiated from source node 6, it can be seen in figure 28. Node 9 acts as a active blackhole node and thus drops all the packets it received as shown in figure 29. Throughput of dropping packets at Blackhole node is shown in Figure 30.



**Figure 27: Number of Packets Sent at all the Nodes**

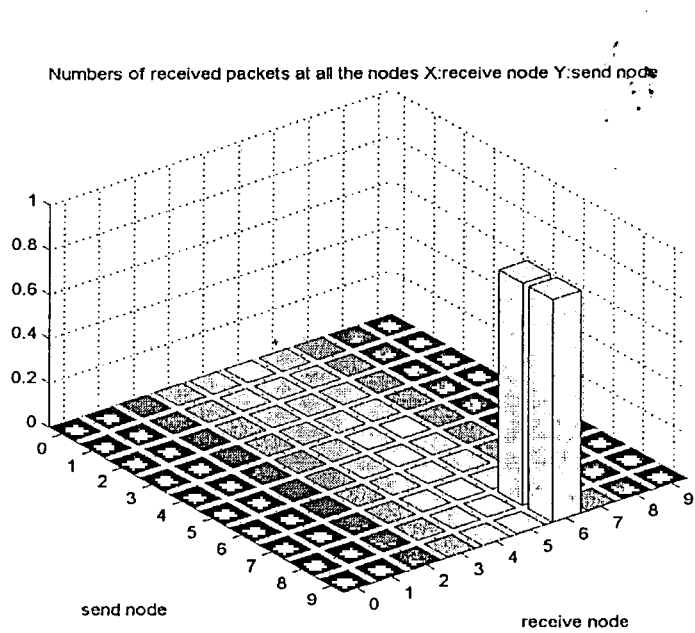


Figure 28: Number of Packets Received at all the Nodes

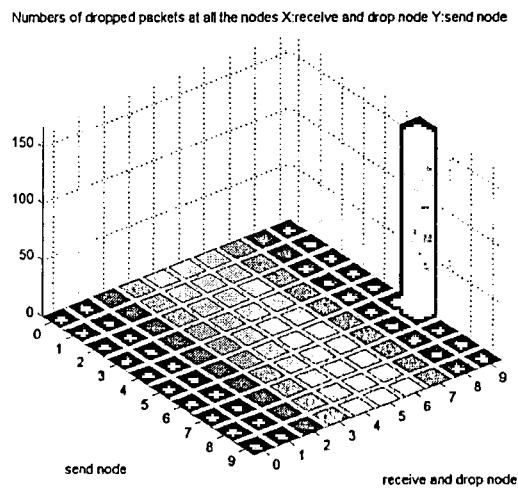
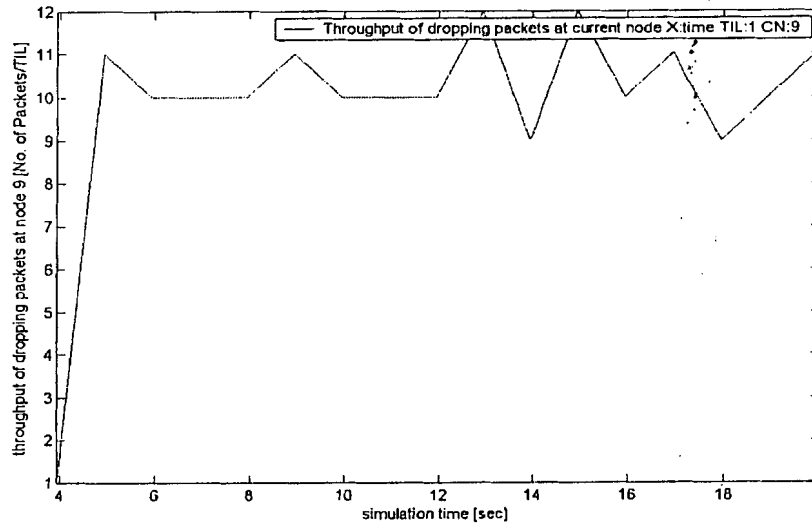


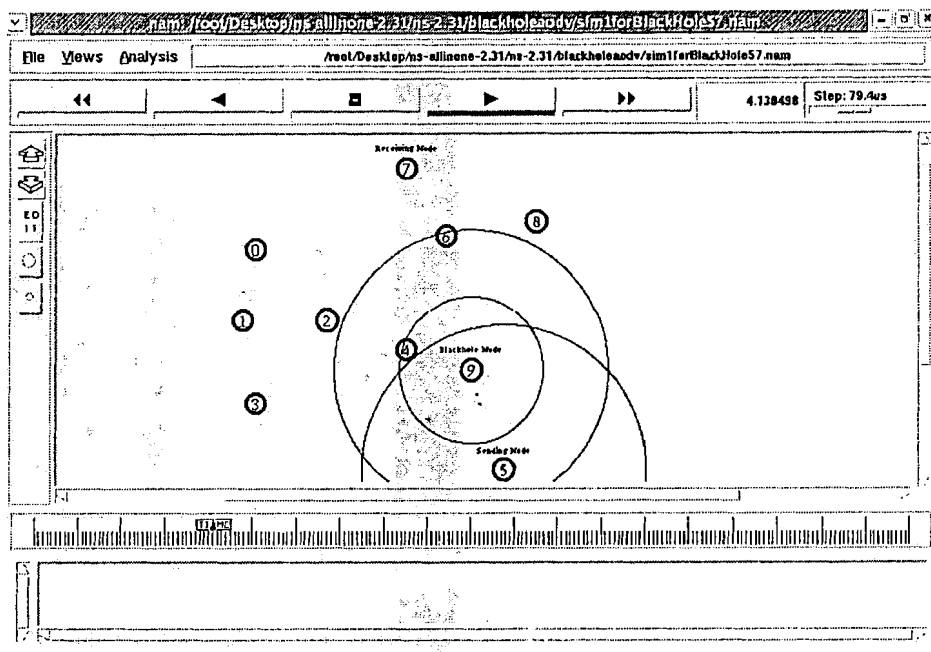
Figure 29: Number of Packets Dropped at all the Nodes



**Figure 30: Throughput of Dropping Packets at Blackhole Node (Node 9)**

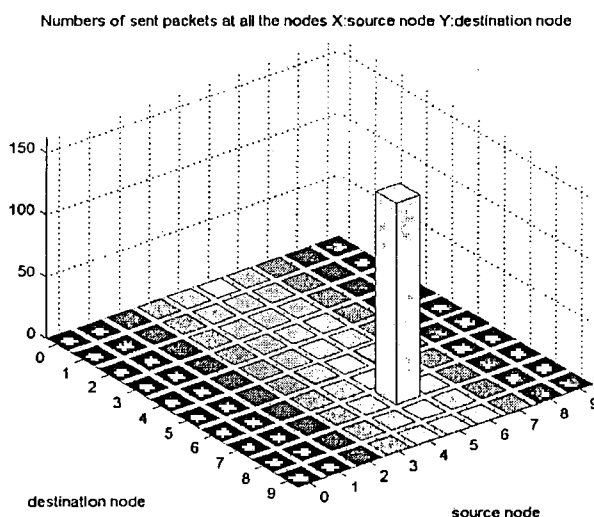
### 6.1.3 Scenario III

For a Network of 10 nodes Simulation of 20 seconds was performed. Source Node is 5, Destination Node is 7 and Blackhole Node is 9. From source to destination there are two intermediate nodes 9 and 6. The topology of the simulation scenario is shown in figure 31.

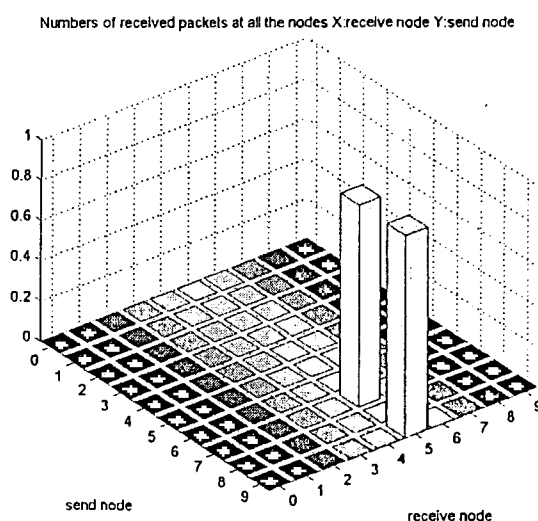


**Figure 31: Snapshot of Simulation Scenario III**

A total of 80 packets were sent by Source Node 5 to destination node 7, this is shown using figure 32. Packets received can be seen in figure 33. Node 9 acts as a active blackhole node and thus drops all the packets it received as shown in figure 34. Throughput of dropping packets at Blackhole node is shown in Figure 35.



**Figure 32: Number of Packets Sent at all the Nodes**



**Figure 33: Number of Packets Received at all the Nodes**

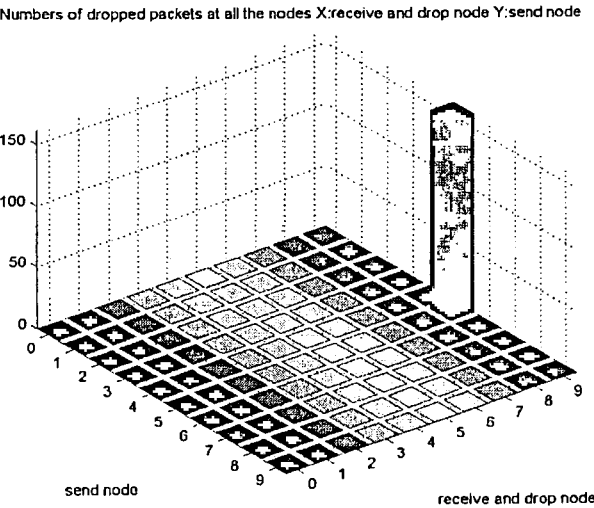


Figure 34: Number of Packets Dropped at all the Nodes

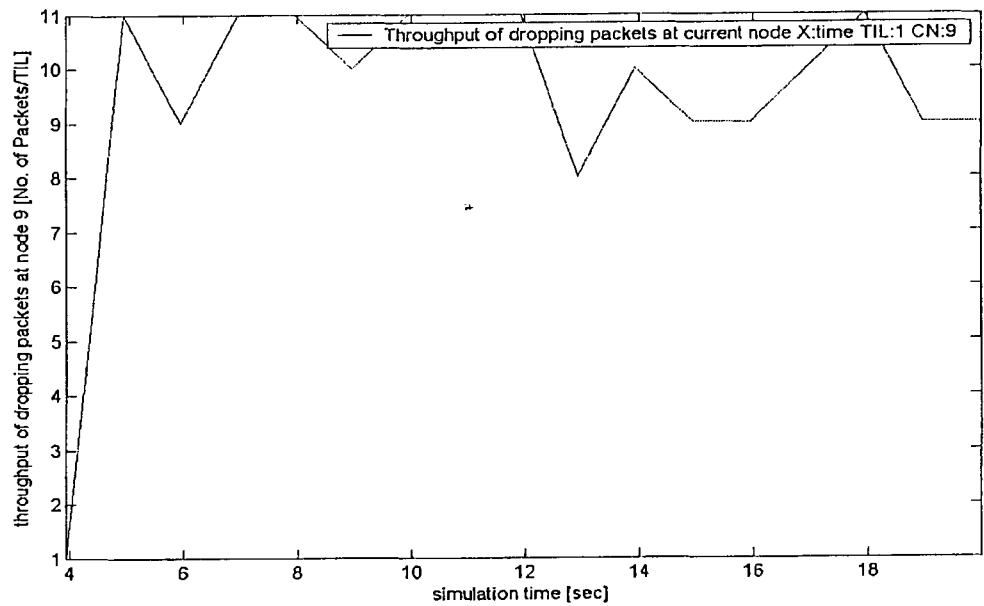


Figure 35: Throughput of Dropping Packets at Blackhole Node (Node 9)

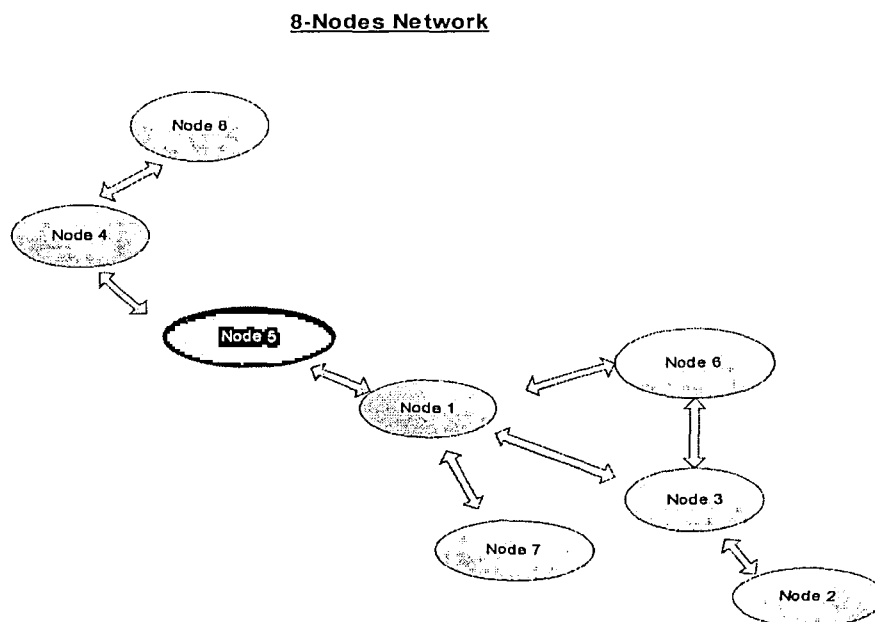


## 6.2 Simulation Scenarios in OMNET ++

In order to test system model three different simulation scenarios are performed. The three scenarios differ in simulation time and number of nodes. The simulation starts and takes some time to setup the network topology and path discovery on the basis of AODV routing protocol. Once the paths are established, the communication starts between the nodes. The sender and destination nodes are randomly chosen. So is the number of packets in one transaction. For the ease of implementation and understanding, one node at a time sends packets on the network.

### 6.2.1 Scenario I

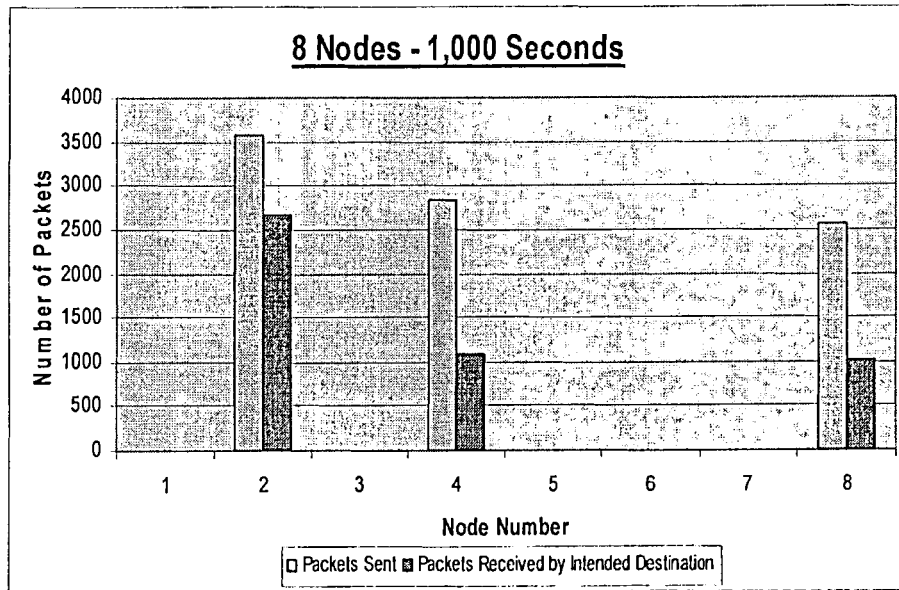
For a network of 8 nodes a simulation of 1000 seconds was performed. The blackhole is situated in our scenario in such a way that it basically divides the network into two logical networks. This can be seen in figure 36, where node 5 is the black hole and bridges two logical portions of the network to make it a single network; i.e. if node-5 is removed from the network, the network is divided into two physical networks. Such a scenario is generally catered for the worst cases among research study. However, in this study it is taken for the ease of identification of data moving to and fro between the two logical networks. The detail of the simulation being run is as follows:



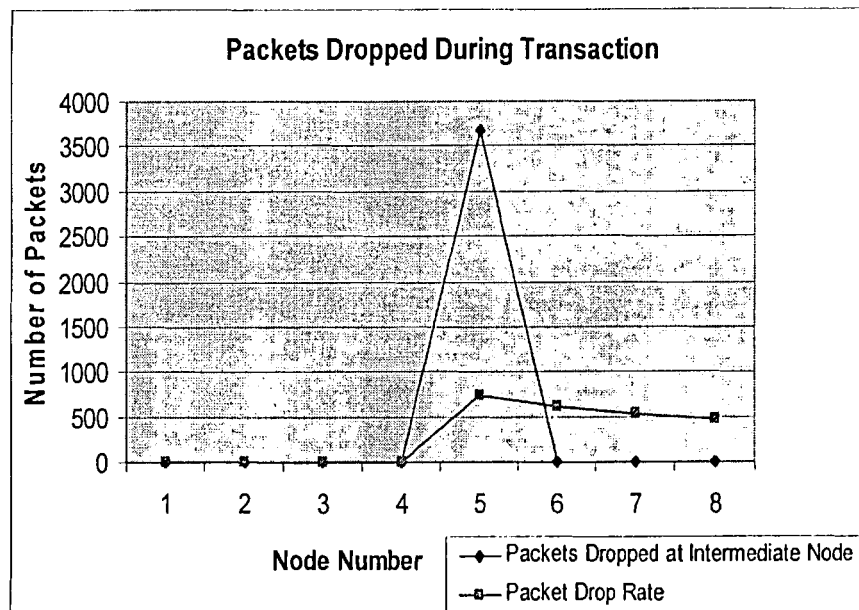
**Figure 36: Topology of Simulation Scenario I**

For the sake of simplicity, three nodes were selected as senders. The selection of these nodes along with intended destinations and number of packets is randomly chosen. Figure 37(a) shows the corresponding graph of the packets sent by the senders and received by the intended destinations. The difference is the dropped packets.

In this scenario, a total of 8,976 packets were sent, out of which approximately 5% packets were reported lost, which is common in wireless networks due to congestion and collusion, etc. However, additional 41% of the packets were not able to reach the intended destination. These packets followed the route in which the blackhole resided (node-5). For the sake of simplicity, the packets lost and packets dropped are shown in the same graph, i.e. figure 37(b). However, they have been separately catered in the simulation.



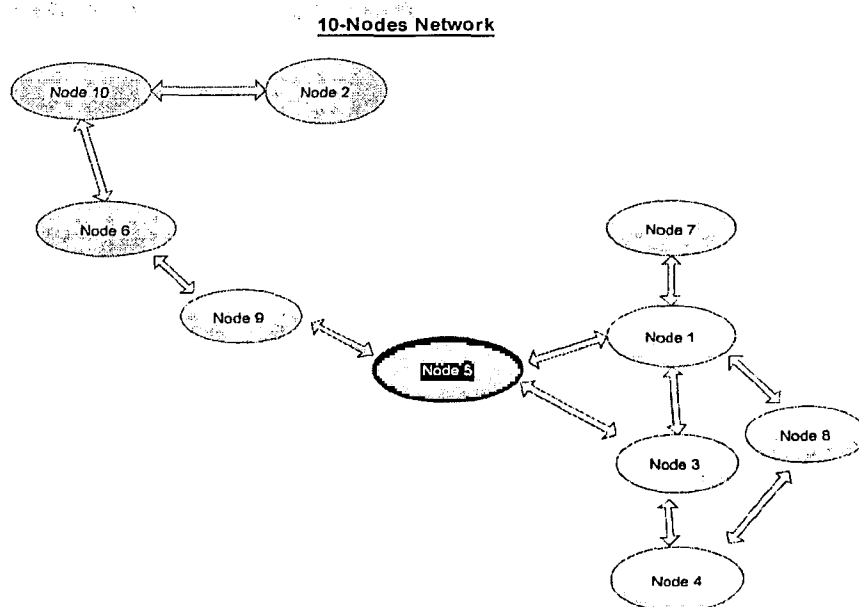
**Figure 37 (a): Packets Sent and Received at all the nodes**



**Figure 37 (b): Packets Dropped during Transaction**

### 6.2.2 Scenario II

For a network of 10 nodes a simulation of 2000 seconds was performed. The blackhole is situated in our scenario in such a way that it basically divides the network into two logical networks. This can be seen in figure 38, where node 5 is the black hole and bridges two logical portions of the network to make it a single network; i.e. if node-5 is removed from the network, the network is divided into two physical networks. Such a scenario is generally catered for the worst cases among research study. However, in this study it is taken for the ease of identification of data moving to and fro between the two logical networks. The detail of the simulation being run is as follows:



**Figure 38: Topology of Simulation Scenario II**

For the sake of simplicity, three nodes were selected as senders. The selection of these nodes along with intended destinations and number of packets is randomly chosen. Figure 39(a) shows the corresponding graph of the packets sent by the senders and received by the intended destinations. The difference is the dropped packets.

In this scenario, a total of 21,243 packets were sent, out of which approximately 5% packets were reported lost, which is common in wireless networks due to congestion and collusion, etc. However, additional 49% of the packets were not able to reach the intended destination. These packets followed the route in which the blackhole resided (node-5). For the sake of simplicity, the packets lost and packets dropped are shown in the same graph, i.e. figure 39(b). However, they have been separately catered in the simulation.

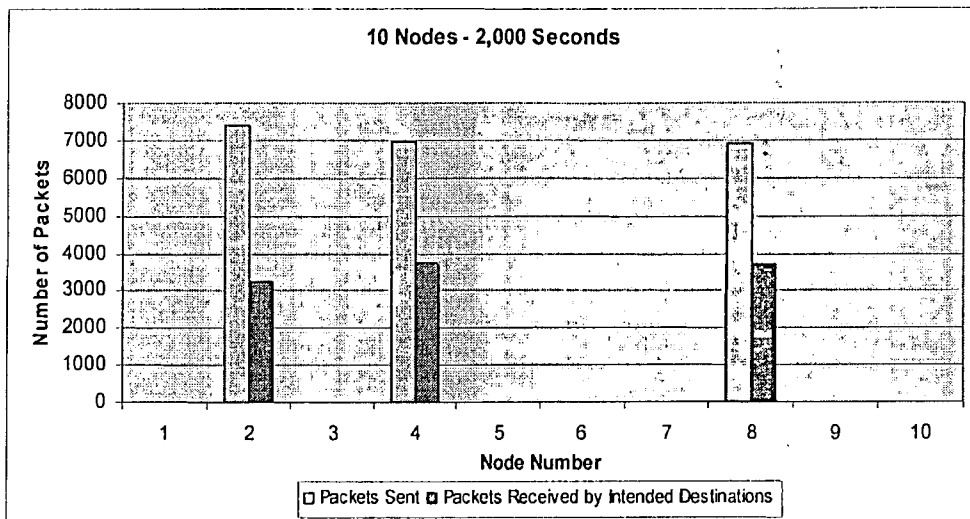


Figure 39 (a): Packets Sent and Received at all the nodes

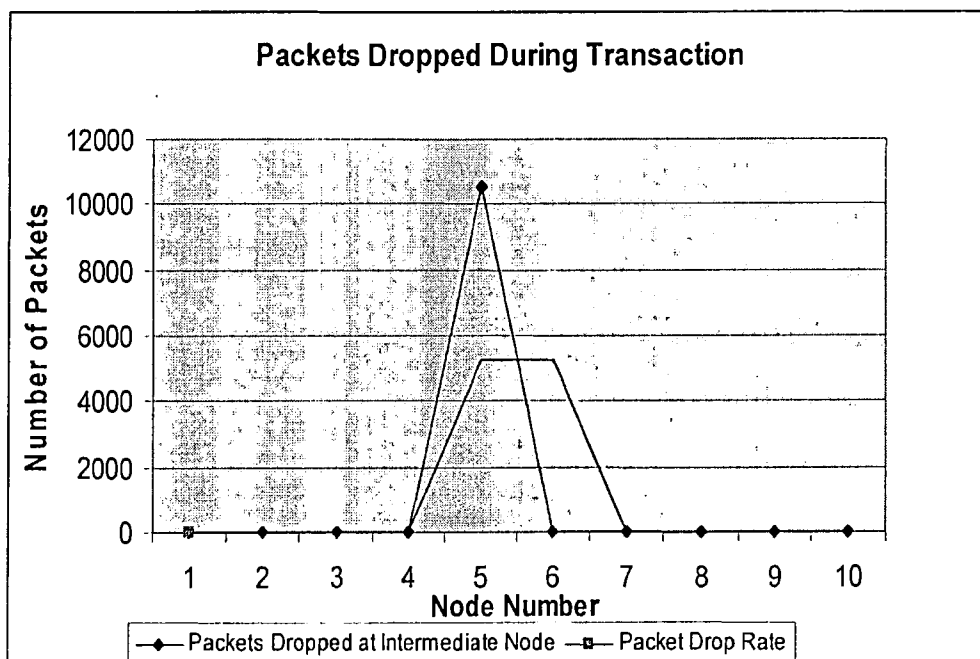
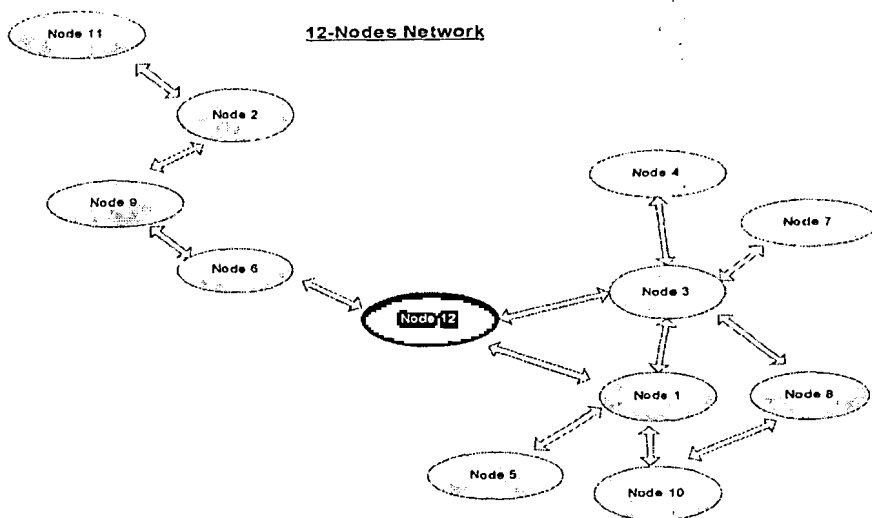


Figure 39 (b): Packets Dropped during Transaction

### 6.2.3 Scenario III

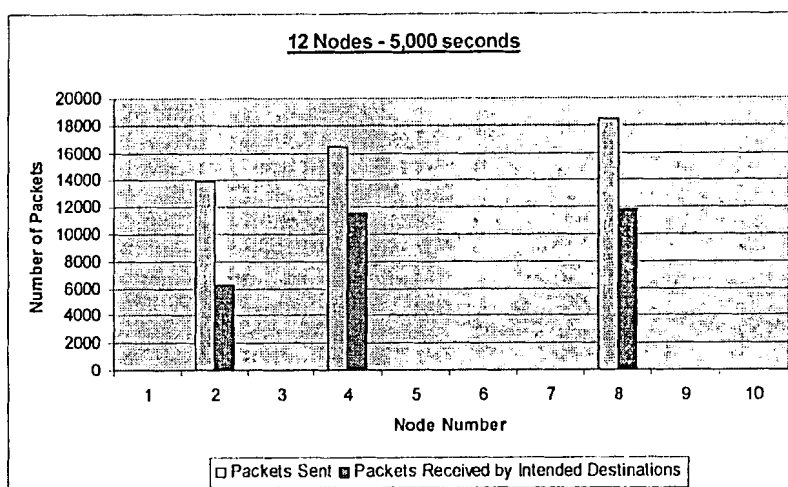
For a network of 12 nodes a simulation of 5000 seconds was performed. The blackhole is situated in our scenario in such a way that it basically divides the network into two logical networks. This can be seen in figure 40, where node 12 is the black hole and bridges two logical portions of the network to make it a single network; i.e. if node-12 is removed from the network, the network is divided into two physical networks. Such a scenario is generally catered for the worst cases among research study. However, in this study it is taken for the ease of identification of data moving to and fro between the two logical networks. The detail of the simulation being run is as follows:



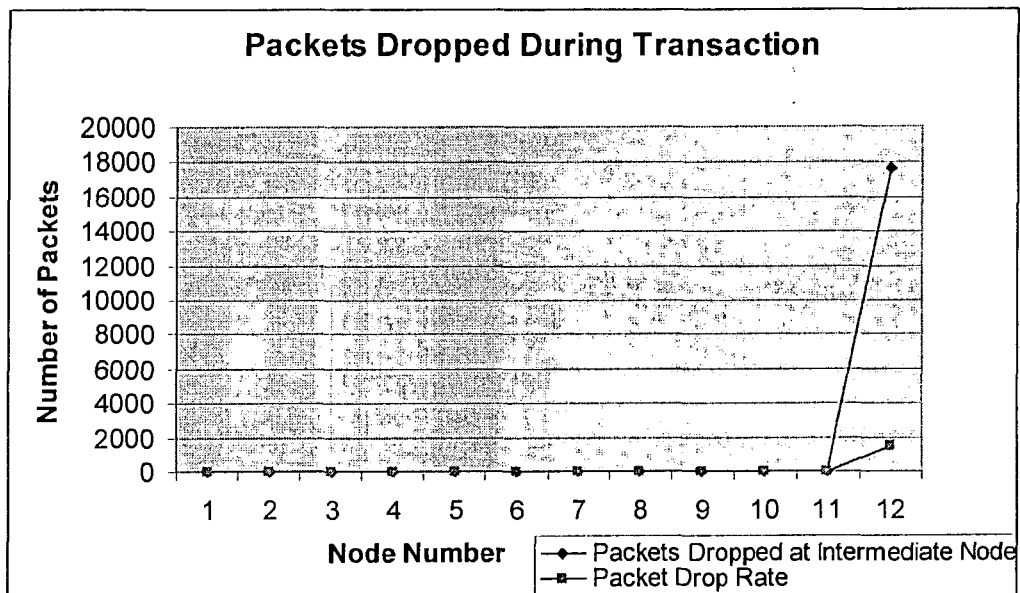
**Figure 40: Topology of Simulation Scenario III**

For the sake of simplicity, three nodes were selected as senders. The selection of these nodes along with intended destinations and number of packets is randomly chosen. Figure 41(a) shows the corresponding graph of the packets sent by the senders and received by the intended destinations. The difference is the dropped packets.

In this scenario, a total of 48,892 packets were sent, out of which approximately 5% packets were reported lost, which is common in wireless networks due to congestion and collusion, etc. However, additional 35% of the packets were not able to reach the intended destination. These packets followed the route in which the blackhole resided (node-12). For the sake of simplicity, the packets lost and packets dropped are shown in the same graph, i.e. figure 41(b). However, they have been separately catered in the simulation.



**Figure 41 (a): Packets Sent and Received at all the nodes**



**Figure 41 (b): Packets Dropped during Transaction**

### 6.3 Results

Simulation results show that active blackhole attack implemented in network simulator 2.31 is dropping all the packets by first attracting them to itself using highest sequence number attack. One has taken three types of simulation scenarios with different sender and receiver nodes. As there is a blackhole node in the network, it incorporates sequence number attack by adding highest sequence number i.e; 4294967295 in RREP message in response of every RREQ. Graphs show that active blackhole attack attracts all the packets toward itself on the basis of this highest sequence number and later on drops all the packets thus decreasing network performance. According to simulation results, a source node sends packets and is received by blackhole node either directly if it is in the transmission range of the sender node or indirectly through an intermediate node. All the packets and throughput of packets being dropped at the blackhole node can be seen in the graph which shows malicious intent of the blackhole node.

In OMNET simulations, Passive Blackhole node in order to save its battery power and to extend its existence in the network behaves in a selfish way and drops the packets coming to it randomly on the basis of some probability value. Simulation scenario shows that the blackhole node is a prominent part of the network as it is connecting two logical portions of one physical network. This node wants to save its battery power in order to remain in the network. For this it becomes selective in forwarding packets on behalf of others and hence become a selfish node. Simulation graphs for three different scenarios show that almost 50 % of the packets can be affected by this type of passive blackhole node thus decreasing network performance.

## CHAPTER 07

# CONCLUSION & FUTURE WORK

## Chapter 7: Conclusion and Future Work

### 7.1 Conclusion

A range of attacks intended for the network layer have been recognized and deeply studied in research papers. By attacking the routing protocols, attackers can absorb network traffic, inject themselves into the path between the source and destination, and thus control the network traffic flow. There are attacks that are intended for some particular routing protocols. In DSR, the attacker may modify the source route listed in the RREQ or RREP packets. It can delete a node from the list, switch the order, or append a new node into the list. In AODV, the attacker may advertise a route with a smaller distance metric than the actual distance, or advertise a routing update with a large sequence number and invalidate all routing updates from other nodes. This thesis has focused on improving the security solution for Adhoc Networks using AODV protocol, via implementation of an active and a passive black hole node in a wireless network and designed algorithms to isolate these attacks. Author presented Active Blackhole attack simulation for Adhoc Network using Network Simulator 2.31 and Passive Blackhole attack simulation using OMNET ++. The corresponding results have been presented in the relevant chapters. Author has also designed algorithms to isolate active and passive blackhole attack in Adhoc network environment working under AODV protocol.

### 7.2 Future Work

Future work might include the implementation of this solution for other type of attacks like wormhole or Byzantine attacks. The implemented mechanism does not provide protection from all possible active attacks but can be further extended to protect the ad hoc network from more type of attacks. It can be applied to other reactive protocols like DSR and OLSR.



## **REFERENCES & BIBLIOGRAPHY**

## REFERENCE AND BIBLIOGRAPHY

- [1] Sirisha R. Medidi, Muralidhar Medidi and Sireesh Gavini, "*Detecting Packet-dropping faults in Mobile ad-hoc networks*", In proceedings of IEEE ASILOMAR Conference on Signals, Systems and Computers (ASILOMAR), vol. 2, pages 1708-1712, Monterey, CA, November 2003.
- [2] Yongguang Zhang, Wenke Lee, "*Intrusion detection in wireless Adhoc networks*" Proceedings of the 6th annual international conference on Mobile computing and networking, pages 275–283, 2000.
- [3] Sanjay Ramaswamy, Huirong Fu, Manohar Sreekantaradhya, John Dixon and Kendall Nygard "*Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks*"
- [4] Patrick Albers, Olivier Camp, Jean-Marc Percher, "*Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches*"
- [5] Perkins, C. Royer, E. "*Ad hoc on-demand distance vector routing*". Request for Comments: 3561, University of Cincinnati, July 2003. [www.ietf.org/rfc/rfc3561.txt](http://www.ietf.org/rfc/rfc3561.txt)
- [6] Johnson, D. Maltz, D. "*Dynamic Source Routing in Ad Hoc Wireless Networks*". Request for Comments: 4728, Microsoft Research, February 2007. [www.ietf.org/rfc/rfc4728.txt](http://www.ietf.org/rfc/rfc4728.txt)
- [7] Hongbo Zhou, "*A Survey on Routing Protocols in MANETs*," Department of Computer Science and Engineering, Michigan State University, East Lansing, MI , Technical Report: MSU-CSE-03-08, Mar 28, 2003
- [8] Jacquet, P. et al. "*Optimized Link State Routing Protocol*", Request for Comments: 3626, Project Hipercom, INRIA, October 2003, [www.ietf.org/rfc/rfc3626.txt](http://www.ietf.org/rfc/rfc3626.txt)
- [9] *Dynamic feed-back mechanisms in Trust-Based DSR*,
- [10] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (secure-bgp). *IEEE Journal on Selected Areas in Communications*, 18(4), April 2000.
- [11] Panagiotis Papadimitratos and Zygmunt J. Haas, "*Secure Routing for Mobile Ad hoc Networks*". In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX
- [12] A. Patwardhan, J. Parker, A. Joshi, A. Karygiannis and M. Iorga. "*Secure Routing and Intrusion Detection in Ad Hoc Networks*", Third IEEE International Conference on Pervasive Computing and Communications 2005.
- [13] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. A. Olsson. "*Detecting disruptive routers: A distributed network monitoring approach*". In *Proc. of the IEEE Symposium on Security and Privacy*, pages 115–124, May 1998.

- [14] J. R. Hughes, T. Aura, and M. Bishop. “*Using conservation of flow as a security mechanism in network protocols*”. In *IEEE Symp. on Security and Privacy*, pages 132–131, 2000.
- [15] A.T. Mizrak, Y.-C. Cheng, K. Marzullo, and S. Savage, “*Fatih: Detecting and Isolating Malicious Routers*”, DSN '05: Proc. 2005 Int'l Conf. Dependable Systems and Networks (DSN'05), pp. 538-547, 2005.
- [16] Alper Mizrak, Yu-Chung Cheng, K. Marzullo and S.Savage, “*Detecting and Isolating Malicious Routers*”, IEEE transactions on dependable and secure computing, sep 2006.
- [17] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, third edition.
- [18] Samba Sesay, Zongkai Yang and Jianhua He, “*A Survey on Mobile Ad Hoc Wireless Network*” Information Technology Journal 3 (2): 168-175, 2004 ISSN 1682-6027
- [19] Mario Gerla, Kaixin Xu, Xiaoyan Hong, “*Exploiting Mobility in Large Scale Ad Hoc Wireless Networks*”, IEEE Computer Communication Workshop (CCW 2003), Oct. 2003.
- [20] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, second edition, 2002.
- [21] Khalid Hussain, Faraz Ahsan, “*Conservation of Flow in Wireless Mesh Network*”, NCICT-2007 University of Science and Technology, Bannu, NWFP, Pakistan.
- [22] Bing Wu, Jianmin Chen, Jie Wu, Mihaela Cardei, “*A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks*” *Department of Computer Science and Engineering Florida Atlantic University*
- [23] Sonja Buchegger, “*Coping with Misbehavior in Mobile Ad-hoc Networks*” February, 2004
- [24] H.Deng, W.Li, D.P.Agrawal, “*Routing Security in Wireless Ad-hoc Networks*”, IEEE communications Magazine, vol.40, no.10, October 2002
- [25] Bo Sun, Yong Guan, J. Chen, “*Detecting black-hole attack in Mobile Ad-hoc Networks*”, Personal Mobile Communications Conference, April 2003
- [26] S.Dokurer, Y.M.Erten, C.E.Acar, “*Performance Analysis of ad-hoc networks under black-hole attacks*”, IEEE, 2007

- [27] Faraz Ahsan, Khalid Hussain, Nyla Khadam, "*Identification of Lossy Channel in Wireless Mesh Network using Conservation of Flow*", *Journal of Information & Communication Technology* Vol. 1, No. 2, Karachi, Pakistan, Fall 2007.
- [28] Sebastian Roschke, "*Implementation of Feedback Mechanism into AODV based on NS2*", 2007
- [29] S.Dokurer, "Simulation of Blackhole Attack in Wireless Adhoc Networks", Atılım University, september 2006
- [30] András Varga, "*OMNeT++: Discrete Event Simulation System, Version 3.2 User Manual*"
- [31] Sihyung Lee, Tina Wong Hyong, S.Kim, "*Secure Split Assignment Trajectory Sampling: A Malicious Router Detection System*", ECE Department, Carnegie Mellon University
- [32] R.Perman, "*Network Layer Protocols with Byzantine Robustness*", PHD Thesis Massachusetts Institute of Technology, October 1998
- [33] Sergio Marti, T.J. Giuli, Kevin Lai, "*Mitigating Routing Misbehavior in Mobile Adhoc Networks*", In Proceedings of MOBICOM 2000.
- [34] Frank Kargl, Andreas Klenk, Michael Weber, "*Advanced Detection of Selfish or Malicious Nodes in Adhoc Networks*"
- [35] Sonja Buchegger and Jean Le Boudec, "*Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes-Fairness in Dynamic Adhoc Networks*", In Proceedings of IEEE/ACM Symposium on Mobile Adhoc Networking and compute(MobiHOC), Lausanne, CH, June 2002

