ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Implementing Enhanced AODV Protocol to Prevent Black hole Attack in Mobile Ad hoc Networks

By

Solomon Gebremeskel

A Project paper submitted to
The School of Graduate Studies of Addis Ababa University
in partial fulfillment of the requirements for the Degree of
Master of Science in Computer Science

November, 2011
Addis Ababa

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Implementing Enhanced AODV Protocol to Prevent Black hole Attack in Mobile Ad hoc Networks

By: Solomon Gebremeskel

Advisor: Dejene Ejigu (PhD)

Name and Signature of members of the Examining Board:

Name	Signature
1. Advisor: **Dejene Ejigu (PhD)**	
2. Chair Person:	
3. Examiner :	

Acknowledgement

This project work is done not only by my effort. It has got an input from different individuals. First of all, I present my thanks to my advisor, Dr. Dejene Ejigu, for dedicating his time from the start to the end of this project.

Thanks to my friends whose support and encouragement goes even beyond the accomplishment of this project.

Since the work is cumulative effect of the two years learning, I would like to thank all the staff members of the department of computer science involved in the process as well as my classmates for working together in different projects and assignments in harmony and understandings.

Finally, my thanks go to all my family, since it is difficult to mention their contribution to my achievements in words it is better to say my heart has recorded it forever.

Abstract

Mobile ad-hoc networks are self-configuring networks of mobile devices that can be established without a need for a network infrastructure. The fact that mobile ad-hoc networks lack central administration and use wireless link for communication makes them very susceptible to various types of adversary's malicious attacks. Black hole attack is one of the severe security threats in ad-hoc networks that can be easily performed by exploiting vulnerability of on-demand routing protocols such as AODV. We implemented a solution to prevent black hole attacks imposed by both single and multiple black hole nodes. Intrusion Detection using Anomaly Detection (IDAD), works based on a general principle of Intrusion Detection Systems (IDS). It means an IDAD system identifies anomaly activities of an adversary from normalcy activities of non-malicious mobile nodes. The identification process involves comparing communication attributes of each mobile node participating in a given ad-hoc network. The most distinguishing characteristic of IDAD is that it works in no-peer-trust principle. Unlike the existing black hole attack prevention techniques that rely on the cooperation of mobile nodes to announce a presence of intrusion, IDAD enables each mobile node to protect itself from an intruder. Implementation of the prevention mechanism has been carried out by using the Network Simulation version 2 (NS2). A Java Parser program and Tracegraph has also been used to analyze results of simulation. Post-analysis result proves the prevention method implemented maximizes network performance by effectively preventing black hole attacks against mobile ad-hoc networks as well as minimizing generation of control (routing) packets.

Key words: MANET, Black hole, IDAD, AODV, IDS

Table of Contents

List of Figures v

List of Tables vi

Abbreviations vii

CHAPTER ONE: INTRODUCTION 1

1.1. Background 1

1.2. Motivation 2

1.3. Problem Description 3

1.4. Objectives 4

1.4.1. General objective 4

1.4.2. Specific objectives 4

1.5. Methodology 5

1.6. Organization of the Paper 5

CHAPTER TWO: LITERATURE REVIEW 6

2.1. Mobile Ad-hoc Networks 6

2.2. Characteristics of MANETs 7

2.3. Architecture of MANETs 9

2.4. Application of MANETs 10

2.5. Routing in MANETs 11

2.5.1. Factors that Affect Routing in MANETs 11

2.5.2. Types of Ad-hoc Routing Protocols 12

2.5.3. Problems Associated with Ad-hoc Routing Protocols 19

2.6. Security Threats to MANETs 20

2.6.1. Ad-hoc Flooding Attack 21

2.6.2. Gray Hole Attack 22

2.6.3. Impersonation 22

2.6.4. Modification 22

2.6.5. Passive Eavesdropping 23

2.6.6. Selfish Nodes (Selective Existence Attack) 23

2.6.7. Attack against Routing Table 24

2.6.8. Black Hole Attack 24

CHAPTER THREE: RELATED WORKS 27
CHAPTER FOUR: SOLUTION DESIGN 29
4.1. The Routing Protocol 29
4.1.1 Control Messages of AODV 29
4.1.2 Communication between Nodes 30
4.2. Intrusion Detection Systems 33
4.3. Intrusion Detection using Anomaly Detection 34
4.4. IDAD algorithm 35
CHAPTER FIVE: IMPLEMENTATION 38
5.1. The network Simulator 38
5.2. Implementation of the Simulation System 39
5.2.1. Implementation of AODV in NS2 40
5.2.2. Simulation of MANET Topology with Black Hole Behavior 44
5.2.3. Analyzing Trace Files 47
5.2.4. Comparison of Results 48
5.2.5. Implementing Enhanced AODV Protocol to Prevent Black Hole Attack 52
5.2.6. Testing the IDS-AODV in NS2 53
5.2.7. Simulation of IDSAODV and Evaluation of Results 54
CHAPTER SIX: CONCLUSION AND FUTURE WORK 60
6.1. Conclusion 60
6.2. Future Work 60
References 62
Appendix A: Tcl Script Used in Simulating blackholeAODV and IDSAODV 65
Appendix B: Java Parser 68
Appendix C: Trace File Field Types 71
Appendix D: Sample Trace File 73
Appendix E: IDSAODV C++ Code 76

List of Figures

Figure 2.1: An example of ad-hoc network with various mobile devices 7
Figure 2.2: Example of MANETs application 10
Figure 2.3: Classification of ad-hoc routing protocols 12
Figure 2.4: RREQ broadcast as a route discovery process 14
Figure 2.5: Propagation of RREQ message 16
Figure 2.6: RREP unicast process and Updating sequence number entry of a routing table 17
Figure 2.7: An example of routing zone of node I 18
Figure 2.8: False RREQ message from a black hole node 25
Figure 4.1: Architecture of the system 34
Figure 4.2: IDAD algorithm to prevent black hole attack 36
Figure 5.1: Structure of NS-2 38
Figure 5.2: Screenshot of nodes in each other's transmission range communicating directly 41
Figure 5.3: Nodes communicating via an intermediate node 42
Figure 5.4: Data flow between Node 2 and Node 5 via Node 1 and Node 6 43
Figure 5.5: Data flow between Node 2 and Node 5 via Node 3 and Node 4 43
Figure 5.6: False RREP message of Black Hole Attack 45
Figure 5.7: A late RREP from actual destination node 45
Figure 5.8: A malicious node absorbing traffic 46
Figure 5.9: Mobile nodes communicating with each other using AODV without black hole attack 49
Figure 5.10: A single black hole node absorbing traffic in a MANET topology that uses AODV 50
Figure 5.11: Throughput of received packets in MANET with and without black hole attack 51
Figure 5.12: CBR packet are reached to destination node properly 54
Figure 5.13: Impact of Network size on the performance 56
Figure 5.14: Impact of mobility on the performance 57
Figure 5.15: Throughput of received packets using AODV, BlackAODV, and IDS-AODV 58

List of Tables

Table 5.1: The result of measure metrics without black hole attack 50
Table 5.2: The result of measured metrics with black hole behavior 51
Table 5.3: The result of measured metrics with IDS 55

Abbreviations

ACK	Acknowledgement Packet
AD	Audit Data
AODV	Ad-hoc On-demand Distance Vector
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
CBR	Cluster Bit Rate
DOS	Denial of Service
DRI	Data Routing Information
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
FREQ	Further Request
IARP	Intra-zone Routing Protocol
IDS	Intrusion Detection System
IDAD	Intrusion Detection using Anomaly Detection
IERP	Inter-zone Routing Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Protocol
MANETs	Mobile Ad-hoc Networks
NAM	Network Animator
NS2	Network Simulator version 2
OSPF	Open Shortest Path First
PDA	Personal Digital Assistants
RREP	Route Reply
RREQ	Route Request
RRER	Route Error
TCP	Transmission Control Protocol
TTL	Time to Live
UDP	User Datagram Protocol
ZRP	Zone Routing Protocol

CHAPTER ONE: INTRODUCTION

1.1. Background

A mobile ad hoc network (MANET) is a collection of wireless mobile nodes which have the ability to communicate with each other without having fixed network infrastructure or any central base station. Since mobile nodes are not controlled by any other controlling entity, they have unrestricted mobility and connectivity to others. Routing and network management are done cooperatively by each other nodes. Due to limited transmission power, multi hop architecture is needed for one node to communicate with another through network. In this multi hop architecture, each node works as a host and as well as a router that forwards packets for other nodes that may not be within a direct communication range. Each node participates in an ad hoc route discovery protocol which finds out multi hop routes through the mobile network between any two nodes. These infrastructure-less mobile nodes in ad hoc networks dynamically create routes among themselves to form own wireless network on the fly. Thus, mobile ad hoc networks provide an extremely flexible communication method for any place where geographical or terrestrial constraints are present and need network system without any fixed architecture, such as battlefields, and some disaster management situations [1].

Security is an essential requirement in mobile ad hoc networks to provide protected communication between mobile nodes. MANETs are vulnerable to various security attacks. Black hole is one of the possible attacks, which is a type of routing attack where a malicious node advertise itself as having the shortest path to all nodes in the environment by sending fake route reply. By doing this, the malicious node can deprive the traffic from the source node. It can be used as a denial-of-service attack where it can drop the packets later [2]. To overcome the challenges, there is a need to build a multi-fence security solution that achieves both broad protection and desirable network performance.

To address this need of secure communication, this project implemented the solution to prevent black hole attack in mobile ad-hoc networks using Ad hoc On-demand Distance Vector (AODV) protocol and anomaly detection mechanism.

1.2. Motivation

In the near future, a truly pervasive computing environment is expected with traditional home appliances attached with computing & communicating powers and small devices like mobile phones, Personal Digital Assistants & wearable computers enhancing information processing and accessing capabilities with mobility. Millions of people nowadays have portable computers and they generally want to read their e-mail and access their normal file systems wherever in the world they may be. This demand for mobility has fueled the rapid progression of computer & communication technologies from networks consisting of both stationary hosts & routers, to networks consisting of mobile hosts and stationary routers, and more recently to the other extreme case of networks having both mobile hosts and mobile routers.

The MANET technology truly supports pervasive computing because in many contexts information exchange between mobile units cannot rely on any fixed network infrastructure but on rapid configuration of a temporary wireless network [23]. Due to recent advancements of the technologies in note book computers, PDA and other mobile devices, wireless communication has grown faster than ever before. Wireless networking has many advantages over the traditional wired networking. Freedom of mobility ("Anytime-Anywhere"), easy to build and inexpensive network devices such as wireless modems are some of the advantages. On the contrary, vulnerability to a number of security threats is the major disadvantage associated with wireless networks.

Recent research on MANET shows that the MANET has larger security issues than conventional networks [30]. Security solutions for static networks would not be suitable for MANET. Zhou Haas [32] and Lundberg [33] discussed several types of attacks that can easily be performed against a MANET. In the black hole attack, malicious nodes provide false routing information to the source node whose packets they want to intercept. In denial of service attacks, malicious node floods the targeted node so that the network or the node no longer operates correctly. In route table overflow attacks, an attacker tries to create lots of routes to non existence nodes and overflows the routing tables. In impersonation attacks, malicious node may impersonate another node while sending the control packets to create an anomaly update in routing table and there are many possible security threats that can drastically degrade the performance of a mobile ad hoc

network. But this project mainly focuses on preventing the black hole attack that results in the absorption of traffic in the network by giving false routing information to neighboring nodes.

Black hole attack is a malicious node that falsely replies for route requests without having an active route to the destination. It exploits the routing protocol to advertise itself as having a good and valid path to a destination node. It tries to become an element of an active route, if there is a chance. It has bad intention of disrupting data packets being sent to the destination node or obstructing the route discovery process.

The main purpose of this project is, therefore, to implement a relevant detection and prevention method to the misleading behaviors of black hole attacks. Black hole nodes will be alleviated from pretending to have active route and reduce overcrowd trafficking.

1.3. Problem Description

Security in MANET is the most important concern for the basic functionality of the network. Availability of network services, confidentiality and integrity of the data can be achieved by assuring that security issues have been met. MANET often suffer from security attacks because of its features like open medium, dynamic change of topology, lack of central monitoring and management, cooperative algorithms and no clear defense mechanism.

A black hole is a node that always responds positively with a Route Reply (RREP) message to every Route Request (RREQ), even though it does not really have a valid route to the destination node. Since a black hole node does not have to check its routing table, it is the first to respond to the RREQ in most cases. Then the source routes data through the black hole node, which will drop all the data packets it received rather than forwarding them to the destination. Thus, the malicious node can easily misroute lot of network traffic to itself and could cause an attack to the network.

Previously the works done on Black Hole attack involved in MANET were based on reactive routing protocol like Ad Hoc on Demand Distance Vector (AODV). Black Hole attack is studied under the AODV routing protocol and its effects are elaborated by stating how these attacks

disrupt the performance of MANET. Very little attention has been given to study the impact and protect more than one Black Hole attack in MANET using the protocol and vulnerability of protocol against the attack. There is a need to evaluate and prevent these black hole attacks using an Ad-hoc On-demand Distance Vector (AODV) protocol. The contemporary routing protocols for ad-hoc networks cope well with dynamically changing topology but are not designed to accommodate defense against malicious attackers.

Researchers have proposed solutions to identify and eliminate black hole nodes. They propose a solution for individual black holes. However, they have not considered the case of multiple black hole attacks. According to the existing solution, information about the next hop to destination should be included in the RREP packet when any intermediate node replies for RREQ. The source node then sends a further request (FREQ) to next hop of replied node and asks about the replied node and route to the destination. By using this method we can identify trustworthiness of the replied node only if the next hop is trusted. However, this solution cannot prevent more than one black hole attacks on MANETs.

Having discussed about the problems associated with the existing proposed solution to combat a black hole attack against MANETs, this project work aims to implement a prevention method (algorithm) to prevent performance degradation that occurs as a consequence of both single and multiple (2 or more black hole nodes) black hole attacks.

1.4. Objectives

1.4.1. General objective

The general objective of this project is to examine the effects of multiple black hole attacks and implement a reliable solution using an AODV protocol.

1.4.2. Specific objectives

- To study black hole attacks in MANET and their consequences
- To design AODV protocol based algorithm to prevent single and multiple black hole attack on mobile ad hoc network

- To implement AODV based solution that identifies and prevents single and multiple black hole attack on mobile ad hoc network
- To test the solution and evaluate its performance

1.5. Methodology

So as to achieve the above stated objectives various methods and tools have been used. The following list will describe the methods and the tools used in this work:

- Before implementation, literature review will be conducted to study and analyze the previous efforts specific to MANETs and security issues in MANETs. It is important to understand the basic concepts and developments regarding MANETs and its security issues.
- Literature review will be followed by designing the algorithm that employs easy and better technique to prevent that performance degradation of MANET.
- Different tools like Network Simulator version 2, C++, java parser, tracegraph, and their functionality will be studied.
- During implementation stage, programming languages, simulators and other software tools will be used.
- For the overall system development life cycle, an object-oriented software development approach will be used.
- Lastly, the results will be gathered, analyzed and conclusions will be drawn on the basis of the results obtained from the implementation.

1.6. Organization of the Paper

The remaining part of this document is organized in six chapters. Literature review is dealt with in chapter two. Works related with ours are presented in the third chapter. Chapter four presents the solution design to come up with the desired solution. The results from this work are presented in chapter five together with discussion about the implementations. The final chapter concludes the whole work and states the future work.

CHAPTER TWO: LITERATURE REVIEW

2.1. Mobile Ad-hoc Networks

A computer network allows sharing of resources and information among different devices connected to the network. Computer networks can be classified according to the hardware and software technology that is used to interconnect the individual devices in the network, such as Optical-Fiber, Ethernet, Wireless Local Area Network (WLAN), MANET etc. Ethernet uses physical wiring to connect devices. Frequently deployed devices include hubs, switches, bridges and/or routers. Wireless LAN technology is designed to connect devices without wiring. These devices use radio waves or infrared signals as a transmission medium [2]. MANETs also use radio waves or infrared signals as a transmission medium but unlike Wireless LANs, MANETs have no access points. Access point is a device that allows wireless communication devices to connect to a network using Wi-Fi, Bluetooth or related standards. Lack of access points forces mobile nodes in ad-hoc network to function as both host and routing device.

MANETs are self-organizing networks with a collection of mobile nodes that are capable of communicating with each other as shown in figure 2.1 [3]. Communication in an ad-hoc network does not require existence of a central base station or a fixed network infrastructure. Each node of an ad-hoc network is both a host and a router. This means, a node that is between two nodes that are far away from each other's transmission range can serve as a router to forward data packets. This multi-hop support in ad-hoc networks, which makes communication between nodes outside direct radio range of each other possible, is probably the most distinct difference between MANETs and wireless LANs [3, 4].

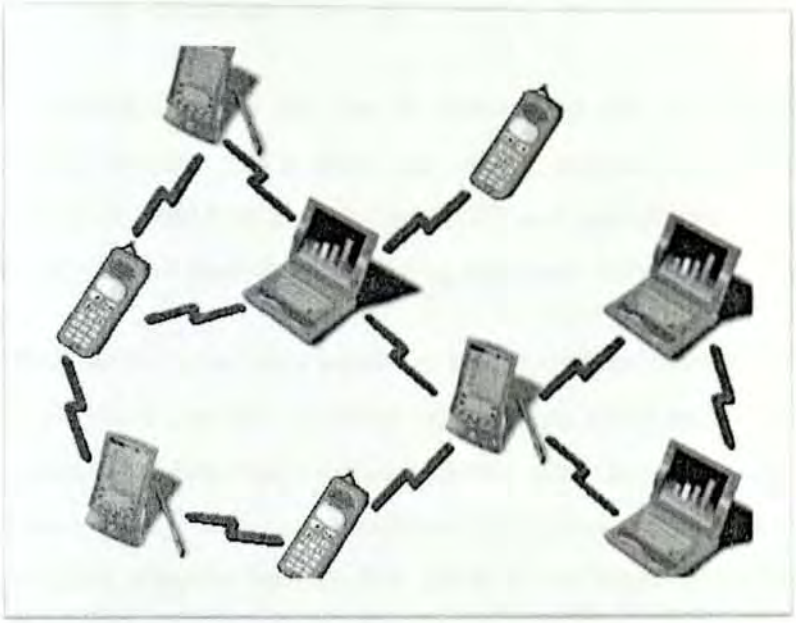

Figure 2.1: An example of ad-hoc network with various mobile devices

2.2. Characteristics of MANETs

A MANET consists of mobile platforms (e.g., a router with multiple hosts and wireless communications devices)--herein simply referred to as "nodes"--which are free to move about arbitrarily. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices, and there may be multiple hosts per router [35]. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interface with a fixed network. In the latter operational mode, it is typically envisioned to operate as a "stub" network connecting to a fixed internetwork. Stub networks carry traffic originating at and/or destined for internal nodes, but do not permit exogenous traffic to "transit" through the stub network [2].

MANET nodes are equipped with wireless transmitters and receivers using antennas which may be omnidirectional (broadcast), highly- directional (point-to-point), possibly steerable, or some combination thereof [7]. At a given point in time, depending on the nodes' positions and their transmitter and receiver coverage patterns, transmission power levels and co-channel interference levels, a wireless connectivity in the form of a random, multihop graph or "ad hoc" network exists between the nodes [10]. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters.

MANETs have several salient characteristics: [5].

- **Dynamic Topology:** Nodes are free to move randomly and organize themselves arbitrarily. They are also free to leave and join the network easily. Thus, the network topology, typically multi-hop, may change rapidly and unpredictably. That leads towards the need of an effective routing, a challenging problem to solve.

- **Low Bandwidth:** Wireless links, which can be either bidirectional or unidirectional, are characterized to have significantly lower capacity than wired mediums. In addition, the actual throughput of wireless communications– after accounting for the effects of multiple access, fading, noise, and interference conditions, etc.–is often much less than a radio's maximum transmission rate. One effect of the relatively low to moderate link capacities is that congestion is typically the norm rather than the exception, so demand from users will likely approach or exceed network capacity frequently. As the mobile network is often simply an extension of the fixed network infrastructure, mobile ad-hoc users will demand similar services. These demands will continue to increase as multimedia computing and collaborative networking applications rise.

- **Energy-Constrained Operation:** Some or all of the nodes in MANETs may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation. As a result, routing algorithms should be as simple as possible to avoid intensive computations. For example, overhearing transmissions requires a large amount of energy to receive and decode entire packets.

- **Limited Security:** Mobile wireless networks are generally more prone to security threats than wired nets. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered. Existing link security techniques are often applied within wireless networks to reduce security threats. As a benefit, the decentralized nature of network control in MANETs provides additional robustness against the single points of failure of more centralized approaches.

- **Scalability:** Network size may vary from a relatively small network with 10 nodes to a large network with 1000 nodes in mobile ad-hoc networks. Thus, routing protocols should be able to scale to this amount.

2.3. Architecture of MANETs

The nature of MANETs makes them very vulnerable to adversary's malicious attacks [22]. Using wireless links as a means of communication medium renders MANETs susceptible to attacks ranging from passive eavesdropping to active interference. Unlike wired networks where an adversary must gain physical access to network wires or pass through several lines of defense at firewalls and gateways, attacks on MANETs can come from any direction and target at any node. The damage that can occur due to the attacks can include leaking secret information, message contamination and node impersonation [5]. All these mean that a mobile ad-hoc network will not have a clear line of defense like the one that can be found in wired networks. Which in turn means every node must be prepared to encounter an adversary directly or indirectly. Mobile nodes are autonomous that are capable of roaming independently. Since tracking down a particular mobile node in a large scale ad-hoc networks can be done easily, attacks by compromised nodes from within the network are far more damaging and much harder to detect. Therefore, any node in a mobile ad-hoc network must operate in a mode that trusts no peer.

Decision making in ad-hoc networks is usually decentralized and many ad-hoc algorithms rely on the cooperative participations of all node involved in the network. The lack of centralized management means that the adversaries can exploit this vulnerability for new types of attacks designed to break the cooperative algorithms [17]. For example, the current MAC protocols for ad-hoc networks are all vulnerable. Although there are many MAC protocols, the basic working principles are similar.

In a contention-based method, nodes must compete for control of the transmission channel each time they want to transmit a message. Nodes must strictly follow a predefined procedure that avoid collusions or recover from them. In a contention free-method, each node must seek from all other nodes a unanimous promise for an exclusive use of a channel resource on a one-time recurring basis [15].

Regardless of the type of MAC protocol, if a node behaves maliciously, the MAC protocol can break down in scenario resembling a denial-of-service attack on the entire network [4]. Although such attacks are rare in wired networks because the physical networks and the MAC layer are isolated from the outside world by layer 3 gateways/firewalls, every mobile node is completely vulnerable in the wireless open medium.

Ad-hoc routing presents other vulnerability. Unlike wired networks where extra protection can be placed on gateways or firewalls, an adversary that hijacks an ad-hoc node could paralyze the entire wireless network by disseminating false routing information. False routing information could result in messages fed to the compromised node.

2.4. Application of MANETs

Once the basic understanding and characteristics of ad-hoc networks have been discussed, it is fairly appropriate to mention some of the areas that ad-hoc networks have a vital importance. Applications of MANETs range from military operation to civil rapid deployment such as emergency search and rescue missions and instantaneous class room or meeting room applications [34].

A common example could be using ad-hoc networks to communicate after the existing infrastructure has been destructed by some natural phenomenon such as earth quake or volcanic eruption. As shown on figure 2.2 [34], MANETs are extremely useful in military.

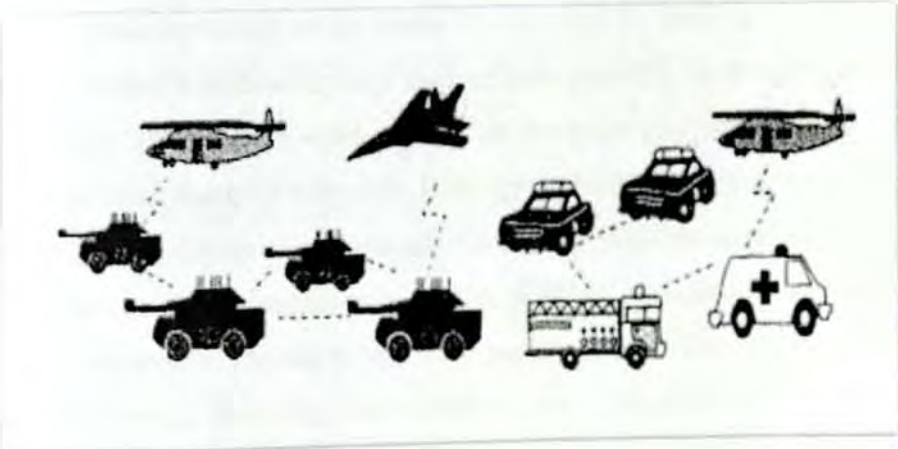

Figure 2.2: Example of MANETs application

2.5. Routing in MANETs

Routing in ad-hoc networks is completely different from routing in traditional wired networks. This is due to the different characteristics of ad-hoc networks such as no fixed infrastructures, use of radio waves as a means of communication, dynamic network topologies and many other reasons. Hence, the traditional routing paradigm cannot be deployed in ad-hoc networks.

2.5.1. Factors that Affect Routing in MANETs

Having a unique nature of mobility, use of wireless links for communication and lack of central administration makes it difficult to develop a standard ad-hoc routing protocol that is not vulnerable to various types of security threats. Below are some of the factors that affect routing process in MANETs [7, 8, 9].

- **Lack of Fixed Infrastructure**: An ad-hoc network is an infrastructure less network. Unlike traditional networks, there is no pre-established infrastructure such as centrally administered routers or strict policy for supporting end-to- end routing. In MANETs, the nodes themselves are responsible for routing packets. Each node relies on the other nodes to route packets. Mobile nodes in each other's radio transmission range can communicate directly, but nodes that are too far apart to communicate directly must depend on the intermediate nodes to forward packets.
- **Dynamic Topology**: Ad-hoc networks contain nodes that may frequently change their locations. Hence, the topology in these networks is highly dynamic. This results in frequently changing neighbors on whom a node relies for routing. Random movement of nodes also makes it difficult to keep track of node position. As a result traditional routing protocols can no longer be used in such an environment. This mandates new routing protocols that can handle the dynamic topology by facilitating fresh route discoveries.
- **Wireless Media**: As the communication is through wireless medium, it is possible for any intruder to tap the communication easily. Wireless channels offer poor protection and routing related control messages can be tampered. The wireless medium is susceptible to signal interference, jamming, eavesdropping and distortion. An intruder can easily eavesdrop to access sensitive routing information or jam the signals to prevent propagation of routing information or worse interrupt messages and distort them to manipulate routes. Routing protocols should be well adopted to handle such problems.

2.5.2. Types of Ad-hoc Routing Protocols

An ad-hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets between computing devices in a mobile ad-hoc network. In ad-hoc networks, nodes do not start out familiar with the topology of their networks; instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcement broadcasts by its neighbors. Routing protocols between any pair of nodes within an ad-hoc network can be difficult because the nodes can move randomly and can also join or leave the network. This means that an optimal route at a certain time may not work seconds later. Generally ad-hoc routing protocols can be classified into three different categories [10] as shown in figure 2.3 [10]:

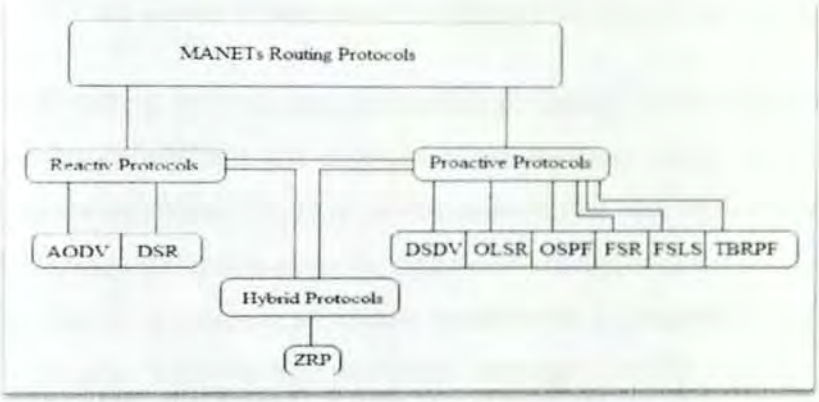

Figure 2.3: Classification of ad-hoc routing protocols

Proactive Protocols: Also known as Table-driven Protocols, work out routes in the background independent of traffic demands. Each node uses routing information to store the location information of other nodes in the network and this information is then used to move data among different nodes in the network. This type of protocol is slow to converge and may be prone to routing loops [2]. These protocols keep a constant overview of the network and this can be a disadvantage as they may react to change in the network topology even if no traffic is affected by the topology modification which could create unnecessary overhead. Even in a network with little data traffic, Table Driven Protocols will use limited resources such as power and link bandwidth therefore they might not be considered an effective routing solution for ad-hoc Networks. Destination Sequenced Distance Vector (DSDV) is an example of a Table Driven Protocol [10].

- **Reactive Protocols**: More often referred to as On-demand Routing Protocols, establish routes between nodes only when they are required to route data packets. There is no updating of every possible route in the network instead it focuses on routes that are being used or being set up. When a route is required by a source node to a destination for which it does not have route information, source node requests for one. On-demand protocols are generally considered efficient when the route discovery is less frequent than the data transfer because the network traffic caused by the route discovery step is low compared to the total communication bandwidth [10].

This makes On Demand Protocols more suited to large networks with light traffic and low mobility. Ad hoc On Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) are common examples of on-demand Routing Protocol [2].

The AODV routing protocol enables multi-hop routing between participating mobile nodes wishing to establish and maintain communication. AODV is based up on the distance-vector algorithm [30]. In an ad-hoc network that uses AODV nodes do not have complete routing information about the entire network that is stored in their routing table. To find a path to destination, all mobile nodes work in cooperation using the routing control messages. With the help of control messages, AODV routing protocol offers quick adaptation to dynamic network conditions such as low processing, memory overhead and low network bandwidth utilization with small size control messages. The most distinguishing feature of AODV compared to the other routing protocols is that it uses a destination sequence number for each route entry. The destination sequence number is generated by the destination when a connection is requested from it. Using the destination sequence number ensures loop freedom. AODV makes sure the route to the destination does not contain a loop and is the shortest path.

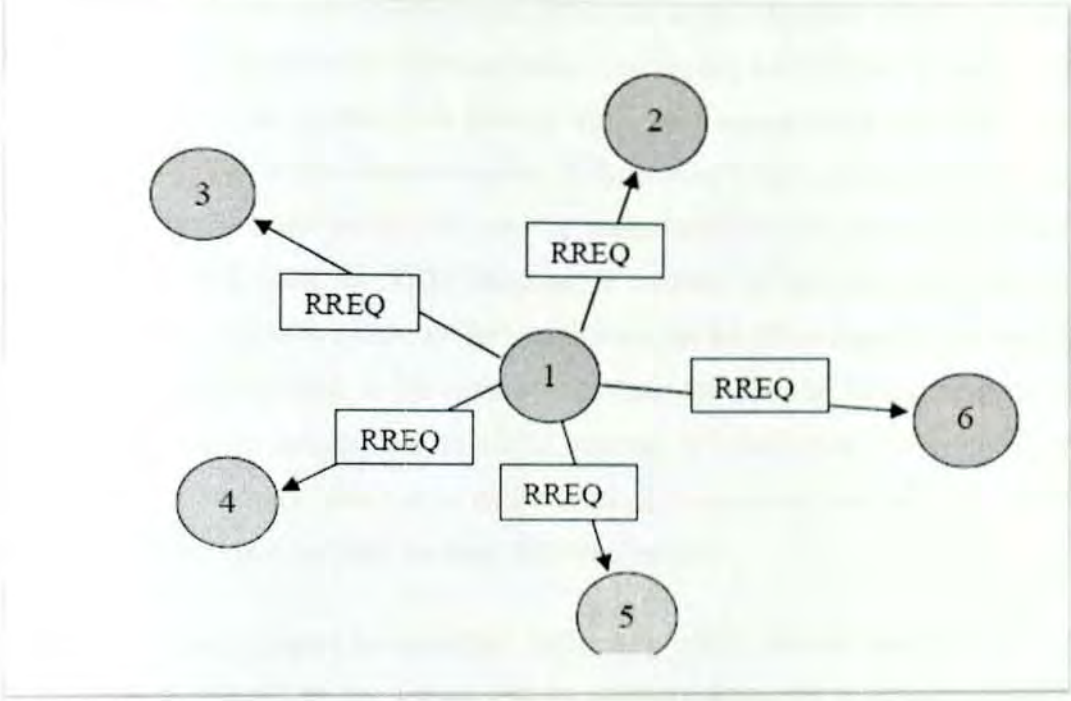

Figure 2.4: RREQ broadcast as a route discovery process

Route Request (RREQ), Route Reply (RREP), and Route Error (RERR) are control messages used for establishing a path from source to destination [30]. When a node wants to initiate transmission of packets, first it needs to broadcast a RREQ message (as shown in Figure 2.4 [20]) to the neighboring nodes inquiring a fresh enough route to a specific destination node. Neighboring nodes that receive RREQ broadcast will check their routing table and respond to the RREQ broadcasts by sending an RREP message if they have fresh enough route to the destination node. Fresh enough means that the intermediate node has a valid route to destination formed a period of time ago, lower than the threshold. To make sure the route is fresh the destination node that sends the RREP will add a destination sequence number to each reply. Otherwise, the node will re-broadcast the original RREQ message to other nodes. While the RREQ message propagates through the network, every intermediate node increases the hop count by one. If an RREQ message with the same RREQ ID is received, the node silently discards the newly received RREQ message, controlling the ID field of the RREQ message. This process will continue until the RREP message reaches the destination node or an intermediate node with a fresh enough route to the destination. Figure 2.5 [20] explains

the RREQ message propagation process. When the destination node or intermediate node that has fresh enough route to the destination receive the RREQ message they create an RREP message and update their routing tables with accumulated hop count and the sequence number of the destination node. If there is no RREP message sent back to the sending node, the destination node may be unreachable and the transmission cannot be established [30]. Once the RREQ message is received by the destination node or an intermediate node with a route to destination node, an RREP message is unicasted (refer to Figure 2.6 [20]) back to the source node along the way the RREQ message came. Here, it should be noticed that an RREQ message is broadcasted to every neighboring node to a source node where as an RREP message is unicasted back via a saved route to the source node that initiated the route discovery process.

If a link breakage occurs an immediate notification will be sent to the neighboring nodes that will be affected by the broken link. In addition, nodes use a periodic broadcast of hello messages as a local advertisement to tell neighboring nodes a specific route is valid and can still be used. If hello messages stop coming from a neighboring node, the neighbor can assume the node has moved away and mark the link to that node as broken and notify the affected set of nodes by sending a link failure notification, another special RREP message [15].

Destination sequence numbers serve as time stamps and allow nodes to compare how fresh their information on the other node is. However, when a node sends any type of routing message, RREQ, RREP, RERR etc., it increases its own sequence number as shown in Figure 2.6 [20]. Higher sequence number is more accurate information and whichever node sends the highest sequence number, its information is considered and route is established over this node by the other nodes. The sequence number is a 32-bit unsigned integer value. If the sequence number of the routes reaches the highest sequence number, then it will be reset to zero (0) [13].

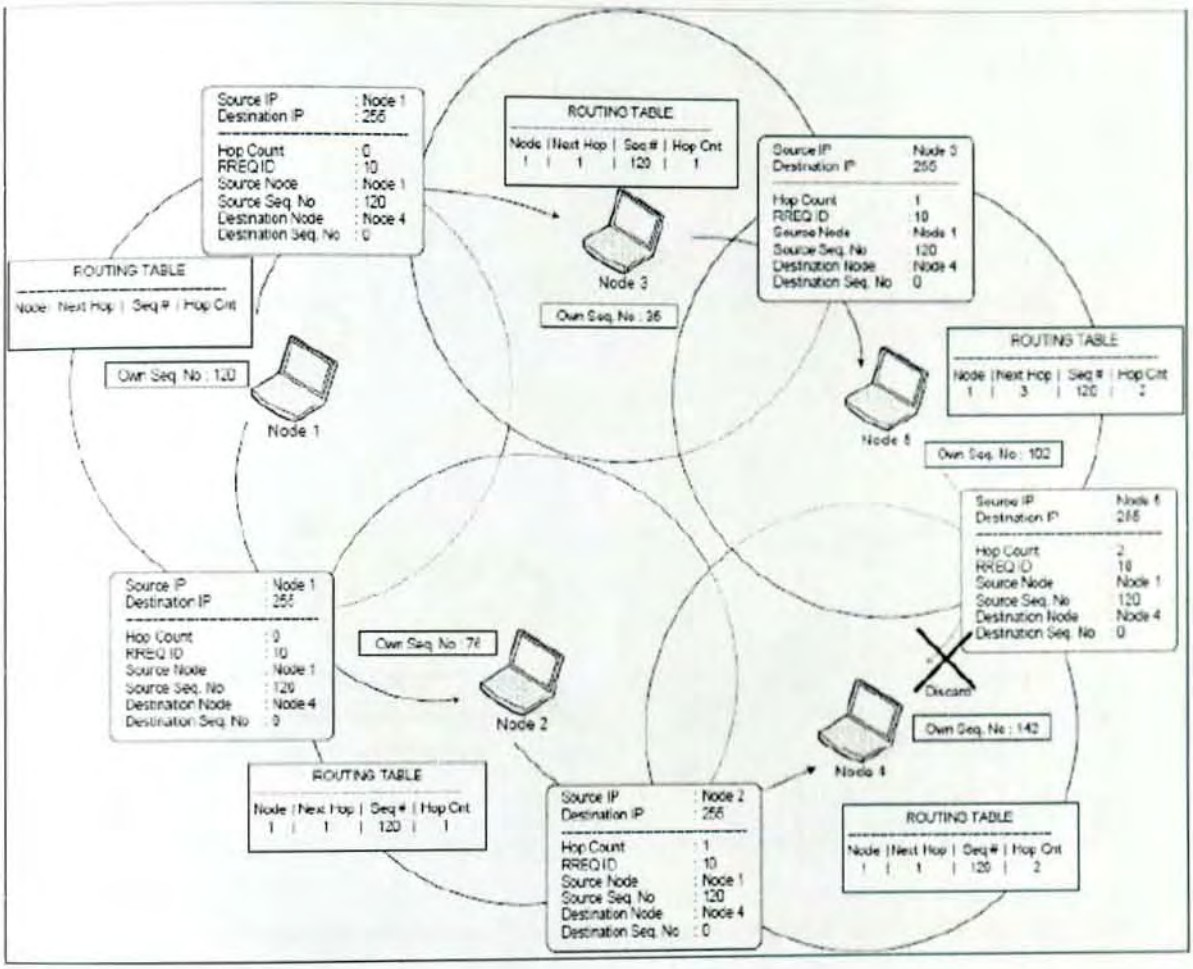

Figure 2.5: Propagation of RREQ message

In Figure 2. 6, while Node 2 forwards the RREP message coming from Node 4, it compares its own previously stored sequence number with that of Node 4. If it notices that the sequence number is newer than its own, then it changes its route table entry as necessary.

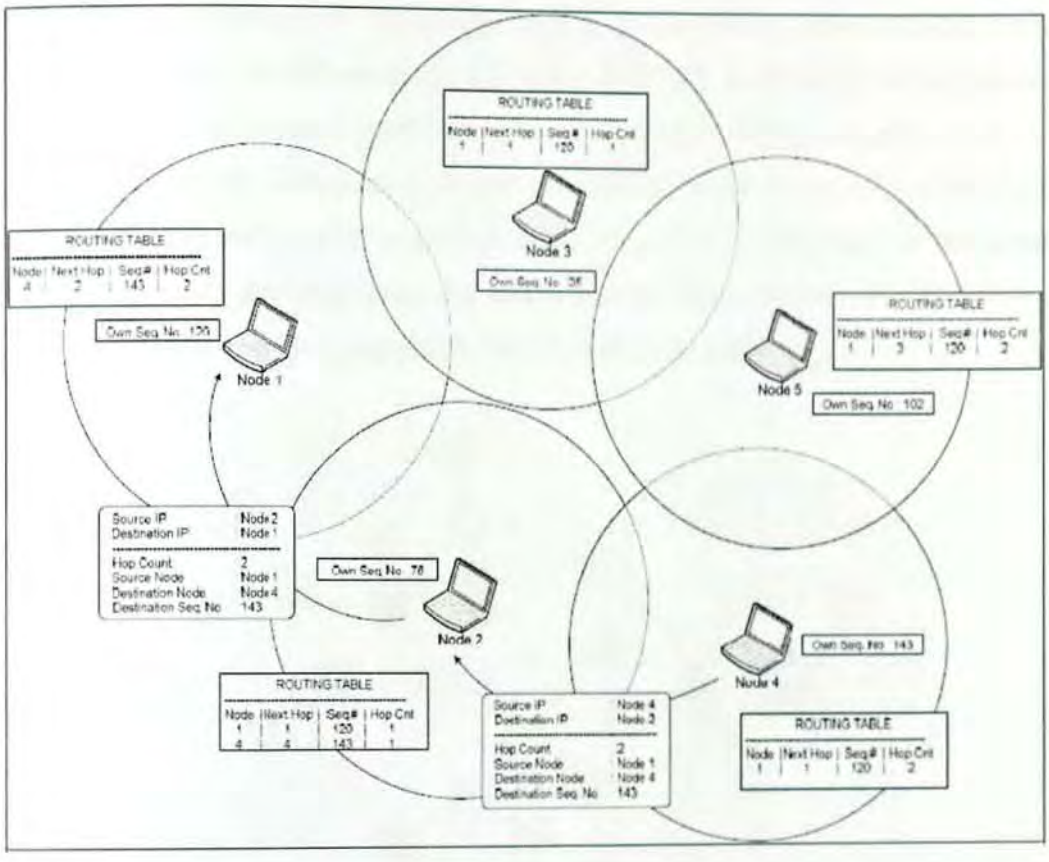

Figure 2.6: RREP unicast process and Updating sequence number entry of a routing table

- **Hybrid Routing Protocols**: As their name indicates these routing protocols combine the strength of proactive (Table-driven) and reactive (On-demand) routing protocols to make up a new protocol with better routing performance [2, 10]. They use distance-vectors for more precise metrics to establish the best paths to destination networks, and report routing information only when there is a change in the topology of the network. Each node in the network has its own routing zone, the size of which is defined by a zone radius, which is defined by a metric such as the number of hops. Each node keeps a record of routing information for its own zone. Zone Routing Protocol (ZRP) is a good example of Hybrid routing protocol. ZRP was the first hybrid routing protocol with both a proactive and a reactive routing component. ZRP was first introduced by Haas [32] in 1997. ZRP is proposed to reduce the control overhead of proactive routing protocols and decrease the latency caused by routing discover in reactive routing protocols. As in Figure 2.7 [32], ZRP defines a zone around each node consisting of its k-neighborhood (e. g. k=3). In ZRP, the distance and a node, all nodes within -hop distance from node

belongs to the routing zone of the node. ZRP is formed by two sub-protocols, a proactive routing protocol: Intra-Zone Routing Protocol (IARP) is used inside routing zones and a reactive routing protocol: Inter-Zone Routing Protocol (IERP) is used between routing zones, respectively. A route to a destination within the local zone can be established from the proactively cached routing table of the source by IARP; therefore, if the source and destination is in the same zone, the packet can be delivered immediately. Most of the existing proactive routing algorithms can be used as the IARP for ZRP.

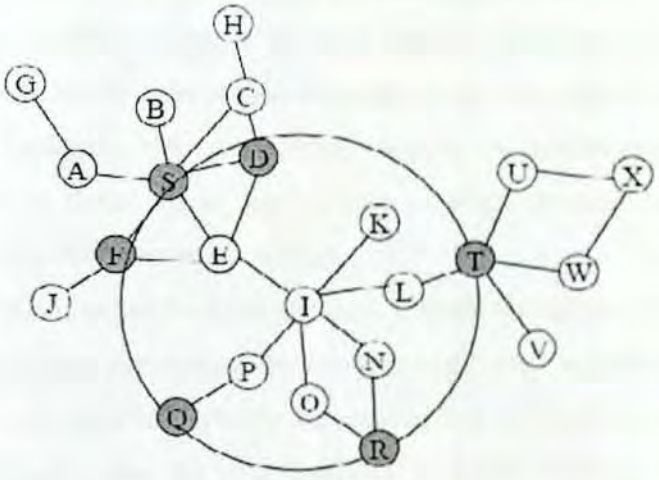

Figure 2.7: An example of routing zone of node I

For routes beyond the local zone, route discovery happens reactively. The source node sends a route requests to its border nodes, containing its own address, the destination address and a unique sequence number. Border nodes are nodes which are exactly the maximum number of hops to the defined local zone away from the source. The border nodes check their local zone for the destination. If the requested node is not a member of this local zone, the node adds its own address to the route request packet and forwards the packet to its border nodes. If the destination is a member of the local zone of the node, it sends a route reply on the reverse path back to the source. The source node uses the path saved in the route reply packet to send data packets to the destination. ZRP has some features that appear to make it somewhat less susceptible to routing attacks [33]. Its hierarchical organization hides some of the routing information within the zones. ZRP provides some form of security against disclosing network topology by dividing routing into zones, which conceal the internal organization.

2.5.3. Problems Associated with Ad-hoc Routing Protocols

Having discussed about some of the ad-hoc routing protocols developed by a special group from Internet Engineering Task Force (IETF) [2, 35] called the MANET group, some of the problems associated with them will be explained briefly.

- **Implicit Trust Relationship between Neighbors**: Current ad-hoc routing protocols inherently trust all participants. This means packet routing is accomplished based on the principle of peer trust [10, 21]. Most Ad-hoc routing protocols are cooperative by nature and depend on neighboring nodes to route packets. This naive trust model allows malicious nodes to paralyze an ad-hoc network by inserting erroneous routing updates, replaying old messages, changing routing updates or advertising incorrect routing information. While these attacks are possible in fixed network as well, the ad-hoc environment makes detection more difficult.
- **Throughput**: Ad-hoc networks maximize total network throughput by using all available nodes for routing and forwarding. However a node may misbehave by agreeing to forward packets and then failing to do so, because it is overloaded, selfish, malicious or broken. Misbehaving nodes can be a significant problem. Although the average loss in throughput due to misbehaving nodes is not too high, in the worst case it is very high. In the case where there is a malicious node that continuously drops traffic that it is supposed to forward (black hole attack), the network throughput will be decreased by a significant amount [24, 30].
- **Lack of Detecting and Isolating Malicious Nodes**: As explained earlier, misbehaving nodes can affect network throughput adversely in worst-case scenarios. The existing Ad-hoc routing protocols do not include any mechanism to identify misbehaving nodes. It is necessary to clearly define misbehaving nodes in order to prevent false positives. It may be possible that a node appears to be misbehaving when it is actually encountering temporary problem such as overload or low battery. A routing protocol should be able to identify malicious nodes and isolate them during route discovery operation [24].

- **Easily Leak Information about Network Topology**: Ad-hoc routing protocols like AODV and DSR carry routes discovery packets in clear text. These packets contain the routes to be followed by a packet. By analyzing these packets any intruder can find out the structure of the network. The attack might use information gained to know which other nodes are adjacent to the target or the physical location of a particular node. Such an attack can be done passively [23, 24]. It can reveal roles of nodes in the network and their location. Intruders can use this information to attack command and control nodes.
- **Lack of Self-stabilization Mechanism:** Routing protocols should be able to recover from an attack in finite time. An intruder should not be able to permanently disable a network by injecting a smaller number of mal-informed routing packets. E.g. AODV however, is prone to self stabilization problems as sequence numbers are used to verify route validity times, and incorrect state may remain stored in routing tables for a long time [25].

2.6. Security Threats to MANETs

Lack of security is the main problem associated with mobile ad-hoc networks [11]. The nodes in MANETs are responsible for dynamically discovering other nodes to communicate. Although the ongoing trend is to adopt ad-hoc networks for commercial uses due to their certain unique properties, the main challenge is the vulnerability to security threats. As MANETs are quickly spreading for the property of its capability in forming temporary network without the aid of any established infrastructure or centralized administration, security challenges has become a primary concern to provide secure communication. The absence of central administration and fixed infrastructure make it very difficult to install firewalls or other protection mechanisms. Use of wireless links renders an ad-hoc network susceptible to link attacks ranging from passive eavesdropping to active impersonation, message replay and message distortion. Eavesdropping might give an attacker access to secret information thus violating confidentiality. Active attacks could range from deleting messages, injecting erroneous messages; impersonate a node etc thus violating availability, integrity, authentication and non-repudiation. Nodes roaming freely in a hostile environment with relatively poor physical protection have non-negligible probability of being compromised [12]. Hence, we need to consider malicious attacks not only from outside but also from within the network from compromised nodes. Ad-hoc network is dynamic due to frequent changes in topology. Even the trust relationships among individual nodes also changes,

especially when some nodes are found to be compromised. Vulnerabilities of operating systems and upper layer applications that belong to user programs such as databases, browsers or client-server applications are not considered as a security issue for ad-hoc networks. Attacks associated with route discovery and packet routing processes are the most devastating threats to the performance of ad-hoc networks. These attacks can originate from both outside of the network or from within the network because of compromised nodes.

In this section, security threats in MANETs will be discussed for a general understanding and the black hole attack in detail. The following are some of the common security threats to mobile ad-hoc networks:

2.6.1. Ad-hoc Flooding Attack

Flooding attack is also one of the most common attacks that compromise the security of ad-hoc networks significantly. This kind of attack is classified under a general name called DOS attack. Flooding attack can result in denial of service when used against on-demand routing protocols for mobile ad-hoc networks, such as AODV, DSR. In such attacks, an intruder broadcasts mass RREQ packets to exhaust the communication bandwidth and node resource so that a valid communication cannot be kept [11, 12].

Flooding RREQ packets in the whole network will consume available network bandwidth and results in congestion. To reduce congestion in a network, AODV protocol adopts some methods. A node can not originate more than RREQ_RATELIMIT RREQ messages per second. After broadcasting an RREQ message, a node waits for an RREP message. If a route is not received within round-trip milliseconds, the node may try again to discover a route by broadcasting another RREQ message, up to a maximum of retry times at the maximum TTL value. Repeated attempts by a source node at route discovery for a single destination must utilize a binary exponential back off [13]. In Flooding Attacks, the whole network will be full of RREQ messages that the attacker sends. The communication bandwidth is exhausted by the flooded RREQ packets and the resource of nodes is also occupied at the same time. If the available bandwidth is dominated by a single malicious node, other nodes will be unable to communicate with each other because of lack of enough bandwidth. This result in a catastrophic damage in a network called Denial of Service [30].

2.6.2. Gray Hole Attack

Gray hole attack is when an adversary drops some of the packets that it has to forward to a destination node. Unlike black hole attack that sends a false RREP message to a RREQ message from a sending node, gray hole attack does not send false RREP messages. Rather, a gray hole attack sends a true RREP message containing a genuine route to a destination node as a reply to RREQ messages. But in gray hole attack, an adversary forwards data packets correctly for some time and will start dropping some of the packets that are destined to a specific destination node [17].

2.6.3. Impersonation

Impersonation attacks are also called spoofing attacks. The attacker assumes the identity of another node in the network, thus receiving messages directed to the node it fakes. Usually this would be one of the first steps to intrude a network with the aim of carrying out further attacks to disrupt operation. Depending on the access level of the impersonated node, the intruder may even be able to reconfigure the network so that other attackers can (more) easily join or he could remove security measures to allow subsequent attempts of invasion. A compromised node may also have access to encryption keys and authentication information. In many networks, a malicious node could obstruct proper routing by injecting false routing packets into the network or by modifying routing information [17].

2.6.4. Modification

Malicious nodes can cause redirection of network traffic and DoS attacks by altering control message fields or by forwarding routing messages with falsified values. Modification attacks can be performed in one of the following several ways.

- Redirection by modified route sequence numbers
- Redirection with modified hop counts
- Denial-of-service with modified source routes
- Tunneling

Tunneling - Ad hoc networks have an implicit assumption that any node can be located adjacent to any other node. A *tunneling* attack is where two or more nodes may collaborate to encapsulate

and exchange messages between them along existing data routes. One type of vulnerability is two such nodes may collaborate to falsely represent the length of available paths by encapsulating and tunneling between them legitimate routing messages generated by other nodes. In this case, tunneling prevents honest intermediate nodes from correctly incrementing the metric used to measure path lengths [34].

2.6.5. Passive Eavesdropping

Eavesdropping is the act of secretly listening to the private conversation of others without their consent. This attack discloses the information about the network but at the same time the attacker can access and read the contents of data packets transmitted over the communication channel. The attacker can mainly gain two types of information by mounting this sort of attack against an ad-hoc network, i.e. the attacker can read data transmitted in the session and also gets information about the packet's characteristics [18].

2.6.6. Selfish Nodes (Selective Existence Attack)

A selfish node is one that does not participate in the network operations but uses the network for its advantage to enhance performance and save its own resources such as power. To achieve that, selfish node puts forth its existence whenever personal cost is involved. Therefore these selfish node behaviors are known as selective existence attacks. For instance, selfish nodes do not even send any HELLO messages and drop all packets even if they are sent to itself, as long as it does not start the transmission [15]. When a selfish node wants to start a connection with another node, it performs a route discovery and then sends the necessary packets. When the node no longer needs to use the network, it returns to the "silent mode" After a while, neighboring nodes invalidate their own route entries to this node and selfish node becomes invisible on the network.

Actually, dropping packets may be divided into two categories according to the aims of the attacking node [30]. Attacker may want to drop the packets of only the other nodes that it will attack later. To do that it must look at the packet to see whether it comes from this node. If attacker looks at the content of all packets aggregating from the network, it spends CPU resource and naturally battery life. This is not desirable behavior for selfish nodes because it spends battery life. Therefore attackers are not interested in the content of the packets if its aim is not to consume its own resources. First category of dropping packets cannot be evaluated as a selfish

node behavior. Selective existence is kind of a passive attack, nodes just do not participate in the network operations and they do not change the content of packets [18].

2.6.7. Attack against Routing Table

Every node has its own routing table to find other nodes easily in the network. At the same time, this routing table draws the network topology for each node for some period of time (max. 3 seconds, duration of ACTIVE_ROUTE_TIMEOUT constant value in the case of AODV protocol) [14]. If malicious node attacks against this table, attacked nodes do not find any route to other nodes to connect. This attack is always performed by fabricating a new control message. Therefore such attacks are also referred as fabricating attacks. There are many attacks against routing tables. Each one is done by fabricating false control messages. For example, to attempt a black hole attack, malicious node first invades into the routing table of the victim, sending false RREP message. Malicious node also spreads false RRER messages to the network so that valid working links are marked as broken. Another attack type against the routing table is an attempt to create too many route entries for nonexistent nodes using RREQ messages. This results a full routing table of an attacked node preventing it from having enough memory to create new entries. This attack type is also known as routing overflow. Attacks against the routing tables also affect the integrity of the network by changing the network topology established in the routing tables of nodes. False control messages are disseminated quickly in the network due to route discovery process and influence the network integrity in a wide range. This type of attack on routing table is called network integrity attack [7].

2.6.8. Black Hole Attack

The contemporary routing protocols for ad-hoc networks cope well with dynamically changing topology but are not designed to accommodate defense against malicious attackers [11, 34]. Generally black hole attack is an attack that happens when a malicious node tries to absorb all the traffic that passes through it and drops it. It has already been discussed that ad-hoc networks that use a reactive routing protocol (AODV in this case) uses a route discovery mechanism to discover and establish a route for communication between participating nodes. In a route discovery process any node that wishes to start communication with another node on the network must send an RREQ message to neighboring nodes in order to establish a two-way route with a desired destination node. Having sent the RREQ message, the sending node must wait for a

possible RREP message either from the destination node or an intermediate node that has a fresh link to the destination node.

A black hole node, as in Figure 2.8 [20], waits for an RREQ message that is sent for a route discovery from another node (sending node) that wishes to establish a route and initiate transmission of data packets with a specific destination node. When the black hole node receives an RREQ message, it will automatically provide the sending node with a so called fresh enough route by sending a false RREP message containing the destination node's address copied from the original RREQ message. Since a malicious node never checks its routing table for the requested route the RREP message it generates is the first to reach the sending node [4].

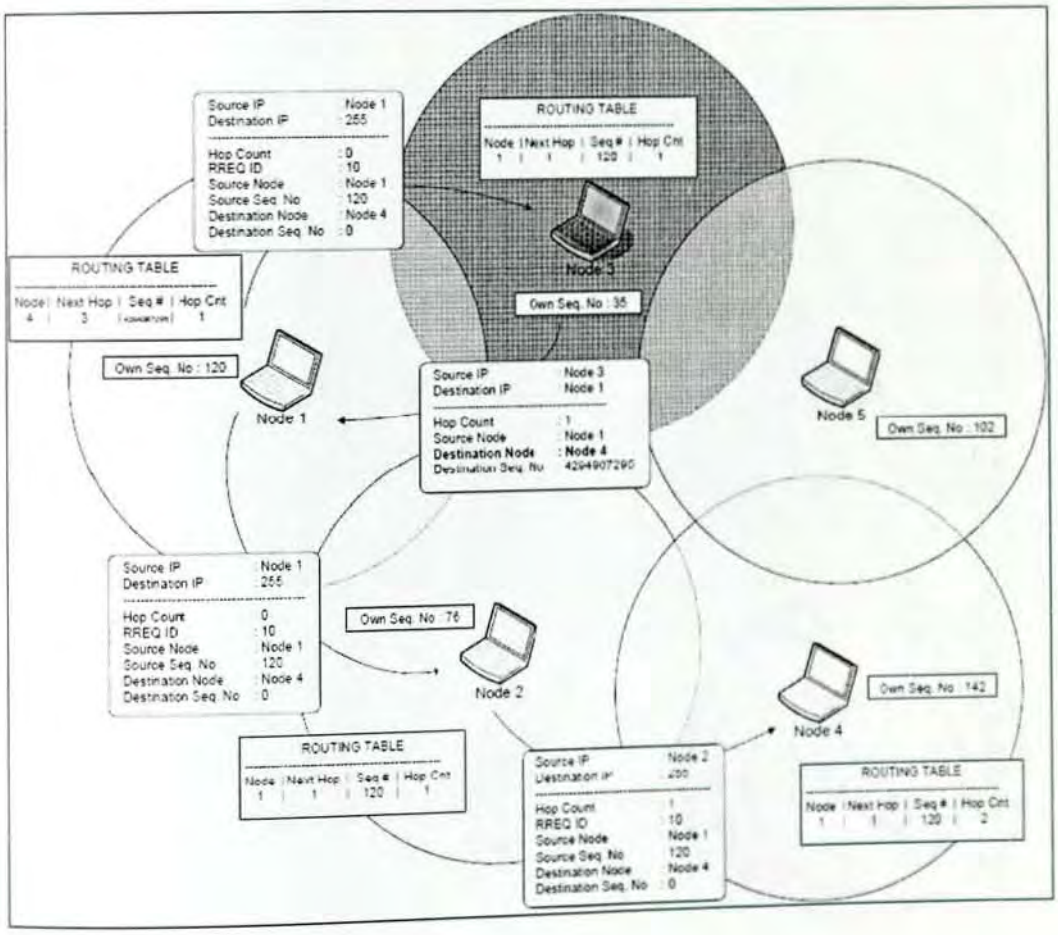

Figure 2.8: False RREQ message from a black hole node

After a while another RREP message will arrive at the sending node from a node with a genuine and fresh enough route to the specified destination node. However, a source node receives only

the first arriving RREP message and discards other RREP messages. Once RREP message is received and route is established, the source node will start sending data packets hopping they will be forwarded to destination. All data packets received by the black hole node will be dropped rather than being forwarded to the actual destination node. By doing so a black hole node greatly compromises the security of ad-hoc networks resulting in a very low over all network throughputs [15, 16]. Reactive protocols such as AODV that work on a route discovery basis are mostly vulnerable to black hole attacks.

CHAPTER THREE: RELATED WORKS

Several researches have been carried out to prevent black hole attack in MANETs. Various techniques have also been proposed to protect MANETs from black hole attack that degrades network performance dramatically by absorbing network traffic. It is a crystal truth that any kind of solution to any problem might have a drawback when it is seen from other points of view rather than the main objectives the solution is supposed to meet. To mention some, the techniques employed in the existing solutions have a low efficiency when it comes to utilization of limited resources and some are a bit complicated in their nature which will make the implementation of the solutions fairly complex.

Several works [29, 30, 31] proposed various solutions that prevent the well known security threat in mobile ad-hoc networks called black hole attack (specifically in Ad-hoc On-demand Distance Vector (AODV) protocol as it is highly vulnerable to such attack).

According to the proposed solution in [29], information about the next hop to destination should be included in the Route Reply (RREP) message when any intermediate node replies for Route Request (RREQ) broadcasts. Then the source node sends a Further Request (FREQ) message to next hop of the intermediate node (the node that sent RREP message) and asks it about the validity of a route through the particular intermediate node. By using this method it is possible to identify trustworthiness of the intermediate node only if the next hop is trusted. This method can work well in isolating the black hole node but it has problem when it comes to efficiency in addition to the problem already mentioned in [30] that the proposed solution does not consider attacks by more than one black hole node.

Extra FREQ messages can increase the network overhead significantly as they have to propagate through multiple nodes if the next hop is located far away from the source node, which will consume the precious bandwidth in wireless communication.

The solution proposed in [31] solved one of the problems or weakness of the solution that was proposed in [29]. But again the new solution does not consider the limitation of resources in wireless communication at all. The authors in [31] have tried to solve the problem of cooperative black hole attack against mobile ad-hoc networks but the mechanism employed was fairly

complicated and far from efficient. The proposed solution introduced two ideas or techniques to prevent cooperative black hole attack.

The fist is application of Data Routing Information (DRI) an extra database having information about past routing experience that will be recorded and maintained by every single node in the network. In addition to the routing table, nodes have to maintain to communicate with other nodes in the network, the introduction of DRI does not only consumes the limited available memory to store the new database but also makes nodes busy in maintaining the DRI that takes up a significant amount of expensive processing time of mobile nodes [7, 10].

The second technique is to apply cross checking of an RREP message from an intermediate node to check if the source node has used this particular intermediate node to route data. If the source node has used the intermediate node before to route data, then the intermediate node is reliable and source node starts to route data through the specified intermediate node. Otherwise, intermediate node is unreliable and the source node sends an FREQ message to next hop to check the identity of the intermediate node.

In [31], anomaly detection method was used in their technique but anomaly condition checking depends merely on destination sequence number. However, this is not always the case. A large network with many mobile nodes and numerous route discovery processes may reach maximum destination sequence number even under normal circumstances.

The other drawback of this technique is broadcasting extra ALARM packets as a procedure to announce detection of malicious node. This will significantly consume network bandwidth if the network is large.

Those various authors have given various proposals for detection and prevention of black hole attack in MANET but every proposal has some limitations and their respected solutions. Although these approaches lead to black hole node detection, they failed to address what single and multiple black hole attack can have an impact on Ad hoc mobile network. So, this project focused on implementing anomaly detection mechanism for prevention and detection of single and multiple black hole attack in ad hoc mobile networks.

CHAPTER FOUR: SOLUTION DESIGN

4.1. The Routing Protocol

The term routing [9], in a computer network, refers to the process of selecting paths to send data. This process can be split in a routing protocol, used to exchange information about topology and link weights, and a routing algorithm, that actually computes paths between nodes. An ad-hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets between computing devices in a mobile ad-hoc network. In ad-hoc networks, nodes do not start out familiar with the topology of their networks; instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcement broadcasts by its neighbors.

Ad hoc On-Demand Distance Vector (AODV) is described in RFC 3561 [35]. It is reactive routing protocol that enables multi-hop routing between participating mobile nodes wishing to establish and maintain communication. AODV is based up on the distance-vector algorithm. In an ad-hoc network that uses AODV nodes do not have complete routing information about the entire network that is stored in their routing table.

To find a path to destination, all mobile nodes work in cooperation using the routing control messages. With the help of control messages, AODV routing protocol offers quick adaptation to dynamic network conditions such as low processing, memory overhead and low network bandwidth utilization with small size control messages.

4.1.1 Control Messages of AODV

AODV protocol establishes routes between nodes only when they are required to route data packets and provide topology information for the nodes. For the purpose of our project the following control messages are required the system for route discovery and maintenance:

- **Route Request Message RREQ:** Source node that needs to communicate with another node in the network transmits RREQ message. AODV floods RREQ message, using expanding ring technique. There is a time to live (TTL) value in every RREQ message, the value of TTL states the number of hops the RREQ should be transmitted [2, 7, 10].

- **Route Reply Message RREP:** A node having a requested identity or any intermediate node that has a route to the requested node generates a route reply RREP message back to the originator node [7].

- **Route Error Message RERR:** Every node in the network keeps monitoring the link status to its neighbor nodes during active routes. When the node detects a link crack in an active route, Route error (RERR) message is generated by the node in order to notify other nodes that the link is down [7, 10].

- **False Route Reply Message:** When the black hole node receives an RREQ message, it will automatically provide the sending node with a so called fresh enough route by sending a false RREP message containing the destination node's address copied from the original RREQ message. Since a malicious node never checks its routing table for the requested route the RREP message it generates is the first to reach the sending node [2].

- **Route Maintenance Process:** When the transmission of data started, it is the responsibility of the node that is transmitting data to confirm that the next hop received the data along with source route. The node generates a route error message, if it does not receive any confirmation to the originator node. The originator node again performs new route discovery process [10].
- **Further Request Message FREQ**: When a source node receives the RREP packet from an intermediate node, it sends a Further Request to the next hop to verify that it has a route to the intermediate node who sends back the RREP packet, and that it has a route to the destination. When the next hop receives Further Request, it sends Further Reply which includes check result to source node. Based on information in Further Reply, the source node judges the validity of the route [30].

4.1.2 Communication between Nodes

Mobile ad-hoc networks are self-organized networks. Communication in ad-hoc network does not require existence of a central base station or a fixed network infrastructure. Each node of an ad-hoc network is the source and destination of some information packets while at the same time it can function as relay station for other packets to their final destination.

In this system we have three types of nodes to have full communication between nodes; source node, destination node, and intermediate node that exist in two forms normal and black hole nodes as described below:

- **Source Node:** All source nodes are mobile and can be connected dynamically in an arbitrary manner. Nodes of MANET behave as router and takes part in discovery and maintenance to establish a reliable route of each other. When the source node wants to connect to the destination node, it propagates the route request packet to its neighbors [2]. Just as neighbors of the source node receive the broadcasted request packet, they forward the packet to their neighbors and this action happens until the destination is found. Afterward, the destination node sends a replay packet to the source node in the shortest path. The route remains in the route tables of the nodes through shortest path until the route is no longer needed. The source node starts the data transmission as soon as it receives the first RREP, and then later updates its routing information of better route to the destination node. Each route table entry contains the following information [25]:
    - Destination node
    - Next hop number of hops
    - Destination sequence number
    - Active neighbors for the route
    - Expiration timer for the route table entry

  The route discovery process is reinitiated to establish a new route to the destination node, if the source node moves in an active session. If the link is broken and the node receives a notification, and Route Error (RERR) control packet is being sent to all the nodes that uses this broken link for further communication. The source node then restarts the discovery process.

- **Destination Node:** The destination sequence number specifies the freshness of a route to the destination before it can be accepted by the source node. Eventually, a RREQ will arrive to node that possesses a fresh route to the destination. If the intermediate node has a route entry for the desired destination, it determines whether the route is fresh by comparing the destination sequence number in its route table entry with the destination sequence number in the RREQ received [25]. The intermediate node can use its recorded route to respond to the

RREQ by a RREP packet, only if, the RREQ's sequence number for the destination is greater than that recorded by the intermediate node. Instead, the intermediate node rebroadcasts the RREQ packet. If a node receives more than one RREPs, it updates its routing information and propagates the RREP only if RREP contains either a greater destination sequence number than the previous RREP, or same destination sequence number with a smaller hop count. It restrains all other RREPs it receives.

- **Intermediate Node:** for this work we have two types of intermediate nodes, the normal (malicious free) node and the black hole node:

  - **Normal Node:** the intermediate normal node is used if the source node is not in the transmission range of destination node; the intermediate node is used as a routing device to establish a route between source node and destination node and to forward data packets from source to destination.

  - **Black Hole Node:** As mobile ad-hoc networks lack an infrastructure, they are exposed to a lot of attacks. One of these attacks is the black hole attack. In the black hole attack, a malicious node absorbs all data packets in itself, similar to a hole which sucks in everything. At the black hole node, all packets in the network are dropped. A malicious node dropping all the traffic in the network makes use of the vulnerabilities of the route discovery packets of the on demand protocols, such as AODV. In route discovery process of AODV protocol, intermediate nodes are responsible to find a fresh path to the destination, sending discovery packets to the neighbor nodes [16]. Malicious nodes do not use this process but instead, they immediately respond to the source node with false information as though it has fresh enough path to the destination. Therefore, source node sends its data packets via the malicious node to the destination assuming it is a true path. Black Hole attack may occur due to a malicious node which is deliberately misbehaving, as well as a damaged node interface. In any case, nodes in the network will constantly try to find a route for the destination, which makes the node consume its battery in addition to losing packets.

4.2. Intrusion Detection Systems

As network-based computer systems play vital roles in modern society especially in the business world, they have become the main targets of intruders. While stand alone computer systems are relatively safe, those that have a direct connection with a network environment are highly vulnerable to various security problems. For example a personal computer that has no internet connection has a very rare chance of getting computer viruses unless the user carries a virus with external storage devices and infect his/her computer carelessly. But for a computer that is connected to a network especially the internet, there is a very high possibility of being infected by a computer virus or attacked by some kind of intrusion [23].

Intrusion Detection will be used to protect network based systems from possible intruders. For example, once an intruder that acts as a black hole node is detected in a network, a necessary measure could be taken to protect the whole network from collapsing. This not only minimizes the damage that could occur because of the intrusion but also helps to gather evidence for prosecution and even to launch a counter-attack [25]. IDS can collect audit data for the entire network. Therefore, at anytime, the only available audit trace will be limited to communication activities taking place within the radio range, and the intrusion detection algorithms must be made to work on this partial and localized information.

Intrusion detection mechanism assumes user and program activities are observable, for example via system auditing mechanisms; and more importantly normal and intrusion activities have distinct behavior. Intrusion detection therefore involves capturing audit data that can be used to identify when an intruder is trying to attack a system [24]. Intrusion Detection System (IDS) can be classified as Network-based and Host-based based on the audit data that has been collected and the place they will be installed [23].

As stated in [23] Network-based Intrusion Detection Systems are placed on gateways or firewalls of a network and they examine network packets that pass through the network hardware interfaces. Whereas Host based IDS are placed on individual hosts to monitor and analyze activities and events generated by hosts and users of the host.

Host-based IDS agents can be provided with a collection of data that can help the system identify an anomaly from normalcy activities. This collection of information is called an Audit Data (AD) [23].

In this project, we used Host-based IDS to enable the nodes protect themselves by detecting an anomaly activity from an intruder as shown in figure 4.1. Host-based IDS agent equipped with anomaly detection is installed on every node in a mobile ad hoc network.

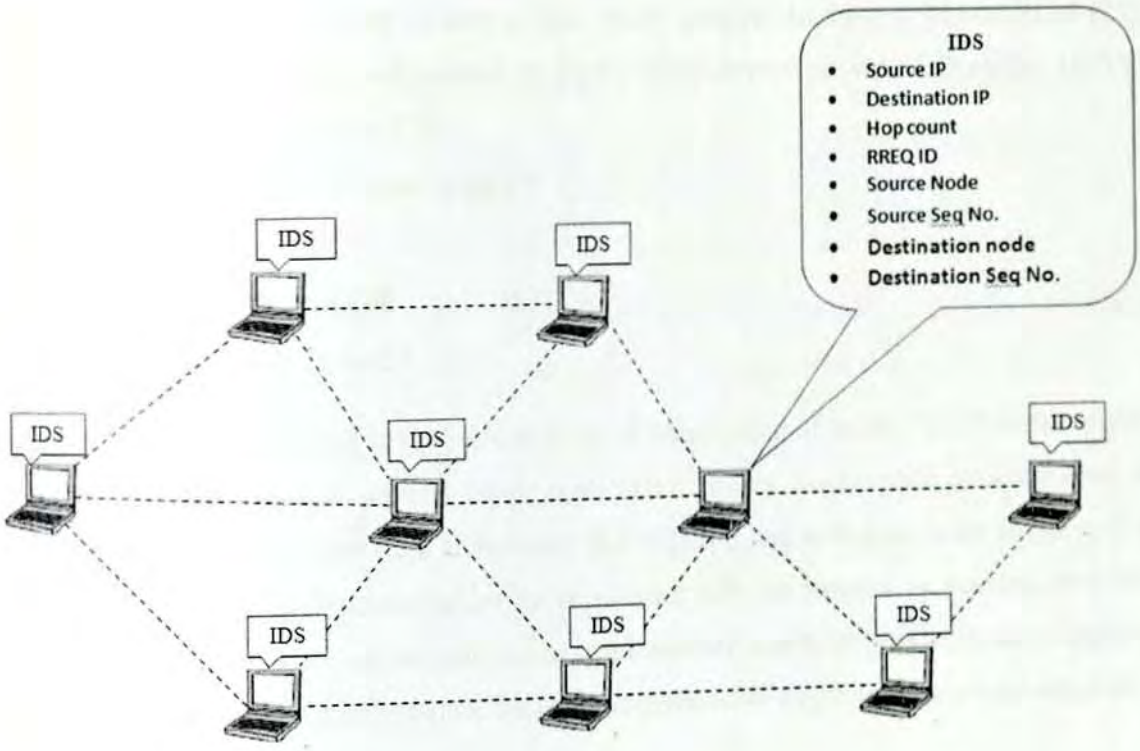

Figure 4.1: Architecture of the system

4.3. Intrusion Detection using Anomaly Detection

In Intrusion Detection using Anomaly Detection (IDAD) the IDS is provided with an audit data to help it identify anomaly and normalcy of activities performed by the nodes or users of the nodes. Anomaly detection systems flag observed activities that deviate significantly from the established normal usage profiles as anomalies, i.e. possible intrusions. For example, the normal

profile of a user may contain the averaged frequencies of some system commands used in his or her login sessions. If for a session that is being monitored, the frequencies are significantly lower or higher, then an anomaly alarm will go on.

In ad-hoc routing process, there is a set of normal activities that will be performed by mobile nodes in communicating with other nodes. Any activities or tasks done by a mobile node that do not resemble these set of normal activities are considered as anomaly.

An AD may consist of possible behaviors of anomaly activities that could happen when a malicious (black hole node) tries to send a fake RREP message to deceive source nodes [23]. Figure 4.1 shows an example configuration of AD for IDS that protects MANETs (using AODV) from black hole attack as follows [25].

- destination sequence number in RREP
- time-stamp of RREP
- number next-hop in RREP
- life time of a route in RREP

Contrary to the technique used in [30] which relies on cooperation of nodes, IDAD works totally in no-peer-trust principle to prevent black hole attack. Every single node in a network is responsible for protecting itself from an intruder. But in [30] if one node detects the presence of a malicious node, it will notify other nodes not to interact with the intruder by sending them the identity of the intruder. This will not only cause extra network overhead but also makes a gap for malicious node to perform attacks before the alarm reaches every single node in a large network.

4.4. IDAD algorithm

IDAD algorithm works in a principle that every single node in MANET's topology is responsible for protecting itself from a malicious node that performs a black hole attack. In other words, there is no co-operation of nodes in preventing black hole attack. This guarantees security of MANETs against black hole attacks no matter how many black hole nodes are participating in performing an attack. If an IDAD system is preventing a MANET from black hole attack by multiple malicious nodes, the percentage of data loss might be slightly less than it would be in a

single black hole attack. This is because the number of possible routes between nodes decreases as the number of nodes isolated by IDAD systems gets more.

To insure the perfection of a decision made by a source node that an RREP message is received from a black hole node, all entries of RREP messages can be checked against a set of predefined audit data for possible anomaly conditions. IDAD Algorithm to prevent black hole attack in MANETs is given in figure 4.2 below:

```
Source Node broadcasts RREQ
Source Node Receives RREP
IF (RREP (R1) is different from Anomaly Detected (A1, A2, A3...An))
{
        If packet destination number is >destination number
        {
        Save route to Routing Table
        WHILE (size of BUFFER is not zero)
        Send Data Packet
        }
        Else
        {
        Discard RREP
        }
}
 ELSE
{
        Discard RREP
        Goto Source Node Receives RREP
}
```

Figure 4.2: IDAD algorithm to prevent black hole attack

In implementing the above algorithm to enable existing AODV routing protocol in NS2, it is necessary to modify the ***recvReply*** function of the ***aodv.cc*** file. The RREP message entries are taken as audit data for anomaly detection. A black hole node is characterized by sending a false

RREP message to deceive source nodes that wish to communicate with other nodes. False RREP messages are generated by a malicious node without looking at the current information about the network topology in the routing table. Since these RREP messages are purposely designed to deceive source nodes that broadcast RREQ messages in the network, they contain the best routing information for the requested destination node. More importantly, false RREP messages have the maximum destination sequence number that tells the freshness of a specific route and life-time route availability.

CHAPTER FIVE: IMPLEMENTATION

This section describes about the implementation of the project in two sections. The first section explains about the development tool, and the second section describes about the system implementation.

5.1. The network Simulator

In this project we have used the Network Simulator Version 2 (NS-2) software for implementation of the system. NS 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 [21]. The NS is a part of software of the Virtual InterNetwork Testbed (VINT) project that is supported by Defense Advanced Research Projects Agency (DARPA) since 1995 [15].

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.

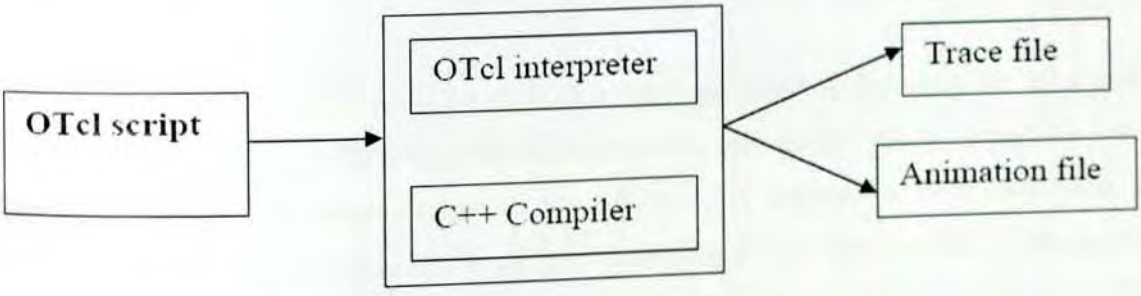

Figure 5.1: Structure of NS-2

As shown in Figure 5.1, an OTcl script written by a user is interpreted by NS. While OTcl script is being interpreted, NS creates two main analysis reports simultaneously [16]. 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. The former is ".nam" file used by NAM software that comes along with NS. The latter is a ".tr" file that includes all simulation traces in the text format.

NS project is normally distributed along with various packages (ns, nam, tcl, otcl etc.) named as "all-in-one package", but they can also be found and downloaded separately. In this project we have used network simulator version 2.34 of ns all-in-one package and installed the package in Ubuntu 10.04 operating system. Using a NS-2 We write the ".tcl" files in text editor and analyzed the results of the ".tr" file using java pareser and tracegraph tool. The implementation phase of the Black hole behavior and the prevention method to the AODV protocol is written using C++.

5.2. Implementation of the Simulation System

Simulating of ad-hoc network topology using the Network Simulator involves a number of processes. Writing a TCL script to generate a network topology that contains mobile nodes, generating a mobility scenario that guides the nodes how to move during their existence in the simulation and generating a traffic file that contains information about network traffic between nodes in the topology are some of the many pre-simulation tasks that should be done in simulating a network topology using one of the ad-hoc routing protocols that exist in NS2.Visualizing using the NAM and analyzing the trace files can be mentioned as post-simulation tasks.

To show the result of the simulation we generate a small size network that has 7, 10, 15, 17, 20, and 25 nodes and create a UDP connection between nodes, and attach CBR (Constant Bit Rate) application that generates constant packets through the UDP connection. CBR packet size is chosen to be 512 bytes long; data rate is set to 1 Mbyte. Duration of the scenarios is 60 seconds and the CBR connections start at time equals to 1.0 second and continue until the end of the simulation, in a 500 x 500 flat space. We randomly selected positions in the space for the nodes to show the data flow and also introduce a random movement to all nodes to show the changes of the data flow in the network.

We have tested our implementation of the Black Hole to see whether it is correctly working or not. To ensure the implementation is correctly working, we used the NAM (Network Animator) application of NS. To test the implementation we used two simulations. In the first scenario we

did not use any Black Hole AODV Node (the malicious node that exhibits the Black Hole Attack will be called "Black Hole Node"). In the second scenario we added a Black Hole AODV Node to the simulation. Then we compared the results of the simulations using NAM as shown in sections below.

5.2.1. Implementation of AODV in NS2

Implementation of routing protocols in NS2 is done by using C++ programming language. Under the main directory ns-2.34 there are folders named after every routing protocol that are implemented in NS2. Like all other routing protocols that are implemented in NS2, AODV has its own folder called *aodv*. In this folder, there are as many as 9 component files used in implementing the AODV routing protocol.

AODV, in NS2, was implemented to simulate a scenario that consists of mobile nodes communicating with each other using the reactive Ad-hoc On-demand Distance Vector routing protocol. In a simulation environment, a mobile node governed by AODV protocol can find a route to a destination node using a route discovery process provided that the specified destination node exists. Once the route between a source and a destination node is found and a UDP connection is established, transmission of buffered data packets is an easy task [20, 21, 22, 23, 24, 25].

In an ad-hoc network using AODV as its routing protocol to find routes from one node to another in the network, participating nodes work cooperatively to help succeed the root discovery process and maintain the routes for communication. A node that wants to transmit some data packets to a specific node in the network broadcasts an RREQ message to find a route for communicating to the desired node. If the desired destination node is within the radio transmission range of the sender node, the communication channel could be established via a direct link up on the arrival of an RREP message from the destination node.

In the simulation scenario shown in Figure 5.2, four mobile nodes namely node 0, node 1, node 2, and node 3 are participating in an ad-hoc network. In this specific MANET topology, node 1 is depicted as a sender and node 0 is named as a receiver node. Since both sender and receiver

nodes are in each other's transmission range, a direct communication link could be established between the sender and receiver node for a packet transmission as depicted in the figure below.

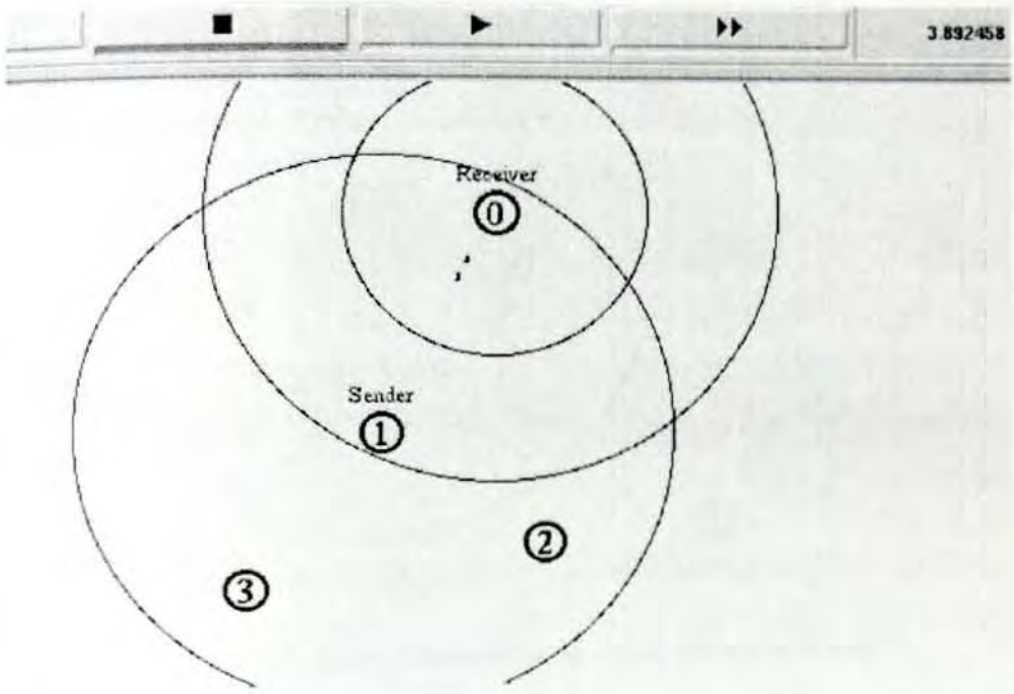

Figure 5.2: Nodes in each other's transmission range communicating directly

Lack of a fixed infrastructure that can accommodate a routing device for routing packets from a source node to a destination node is a vital reason for mobile devices to work both as a host and a routing device at the same time. This cooperative nature of mobile nodes in ad-hoc network can be clearly seen when a node wishes to communicate with another node that is not in its transmission range. When nodes are far apart in a network topology to establish a direct link with each other, intermediate nodes can be used to forward data packets acting as a routing device to keep the whole network alive.

Obviously a source node requests a route to destination by broadcasting an RREQ message and an intermediate node that has a fresh enough route to a destination node will send an RREP message containing the destination IP address, number of hop count, destination sequence informing the freshness of the route and other parameters to the source node. Once the RREP message from an intermediate node is received by the source node, a route from source to destination nodes can be established via the intermediate nodes.

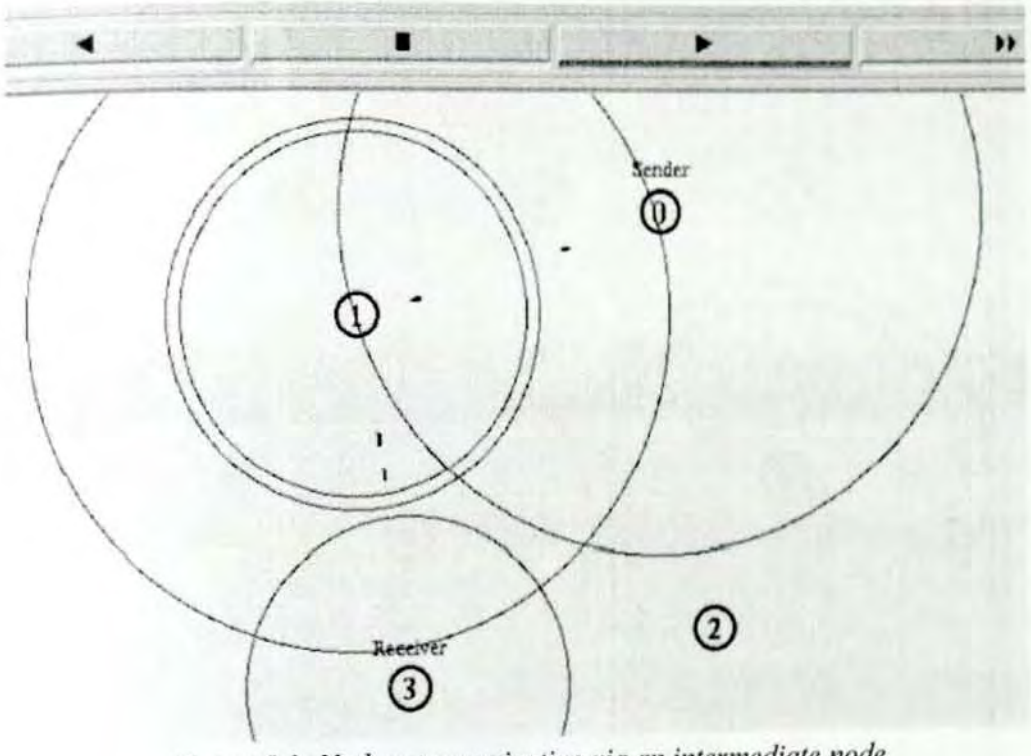

Figure 5.3: Nodes communicating via an intermediate node

In Figure 5.3 above, data packet is being transmitted from node 0 to node 3, source and destination nodes respectively. Because node 3 is not in the transmission range of node 0, the intermediate node 1 is used as a routing device to establish a route between node 0 and node 3 and to forward data packets from source to destination.

Figure 5.4 shows the data flow from Node 2 to Node 5. When Node 1 leaves the propagation range of Node 2 while moving, the new connection is established via Node 3. The new connection path is shown in Figure 5.5.

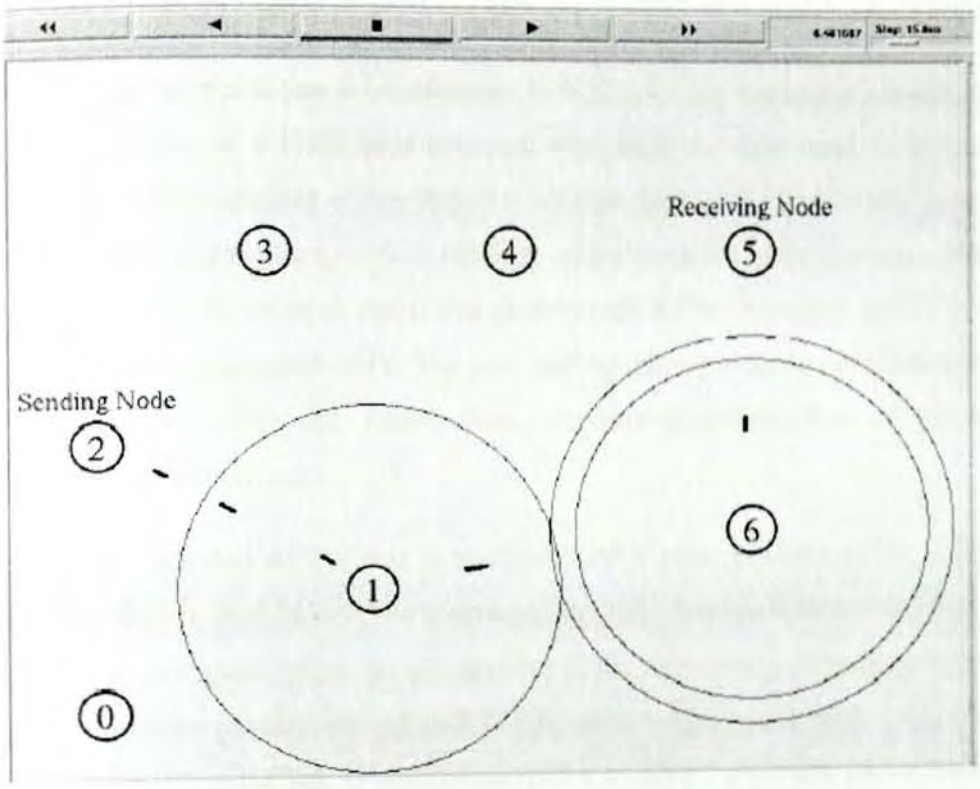

Figure 5.4: Data flow between Node 2 and Node 5 via Node 1 and Node 6

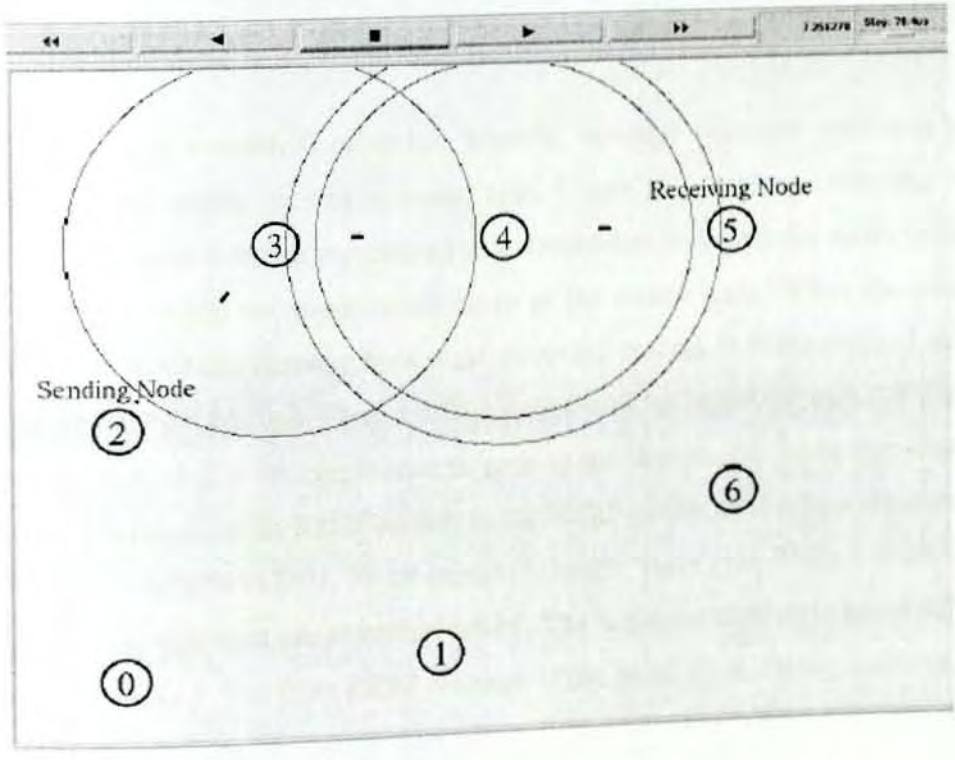

Figure 5.5: Data flow between Node 2 and Node 5 via Node 3 and Node 4

5.2.2. Simulation of MANET Topology with Black Hole Behavior

The existing AODV protocol that is implemented in NS2 does not simulate a network topology with a malicious node or a black hole behavior. Therefore, the first work in achieving the objectives stated in the beginning of this project is to implement a modified AODV protocol that is capable of simulating an ad-hoc network topology with a black hole behavior or in other words with a malicious node that drops all traffic that pass through it. This modified AODV protocol is given an abbreviation blackholeAODV. The new routing protocol has its own folder under the main folder ns-2.34. Inside this folder, there are implementation files of the modified blackholeAODV routing protocol.

In this simulation modified AODV that is blackholeAODV (that provides ad-hoc routing with black hole behavior) is used to simulate a network topology that has 6 nodes and one of them acting as a black hole node. What exactly happens is the adversary promises to forward data packets to a destination node by sending an RREP message telling source node it has fresh route to destination node. Once the link is established via the adversary node and packet transmission is started, the black hole node absorbs all the traffic that it should forward. By doing so the adversary compromises the security of the whole network (dropping data packets) and affects the network performance.

Figure 5.7 presents a simulation of ad-hoc network topology with one malicious node that absorbs the network traffic. In this scenario, node 1 acts as a sending node that wishes to communicate with node 0 that is represented as a destination node. All the nodes including the malicious node are within the transmission range of the source node. When the source node, node 1, sends out an RREQ message for a route discovery process, it is broadcasted to all of the nodes in this particular topology. When malicious node receives an RREQ packet it immediately sends RREP packet as if it has fresh enough path to the destination. Malicious node tries to deceive nodes sending such an RREP packet. In the RREP packet the highest sequence number of AODV protocol is 4294967295, 32 bit unsigned integer value [13]. Values of RREP packet that malicious node will send are described below. The sequence number is set to 4294967295 and hop count is set to 1. The false RREP message of the Black Hole Attack is shown in Figure 5.6 [13].

```
sendReply(rq->rq_src,                    // IP Destination
          1,                             // Hop Count
          index,                         // Dest IP Address
          4294967295,                    // Highest Dest Sequence Num
          MY_ROUTE_TIMEOUT,              // Lifetime
          rq->rq_timestamp);             // timestamp
```

Figure 5.6: False RREP message of Black Hole Attack

Because the malicious node does not check its routing table, the RREP sent by this black hole node is the fastest and the first to be received by the source node. Up on receiving the first RREP message from the malicious node, the source node will establish the communication link and starts transmitting data packets to the malicious node as in Figure 5.7 and 5.8.

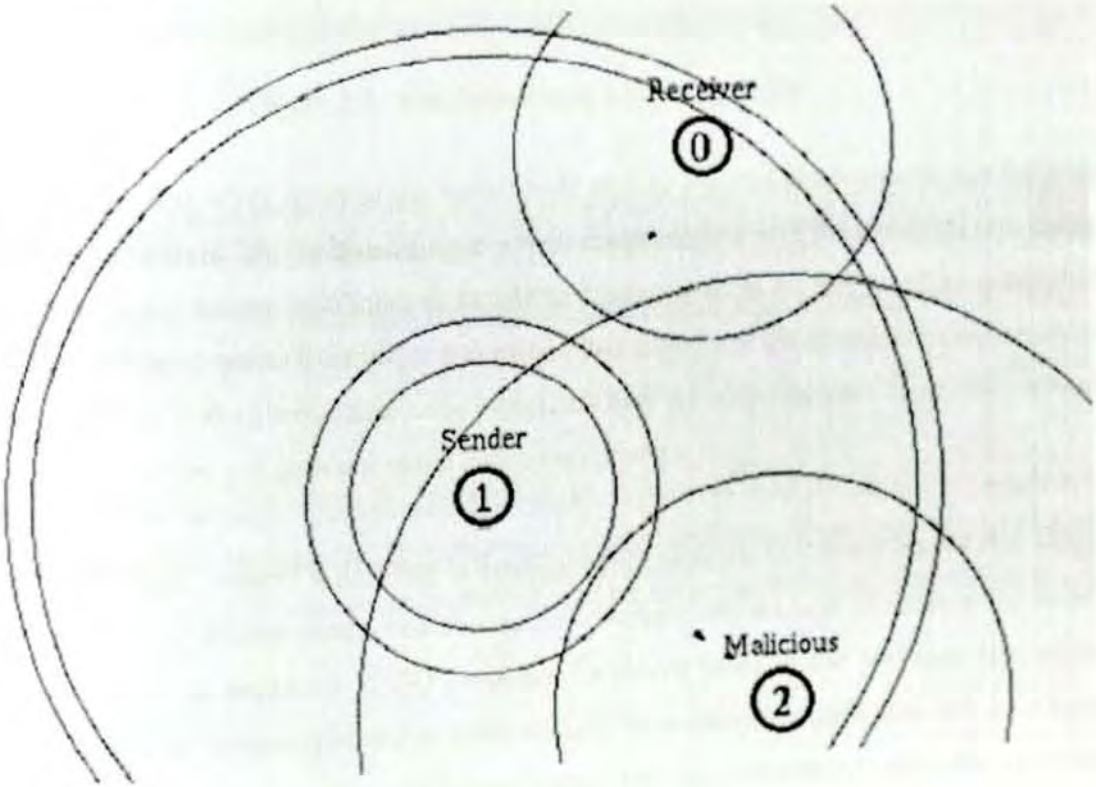

Figure 5.7: A late RREP from actual destination node

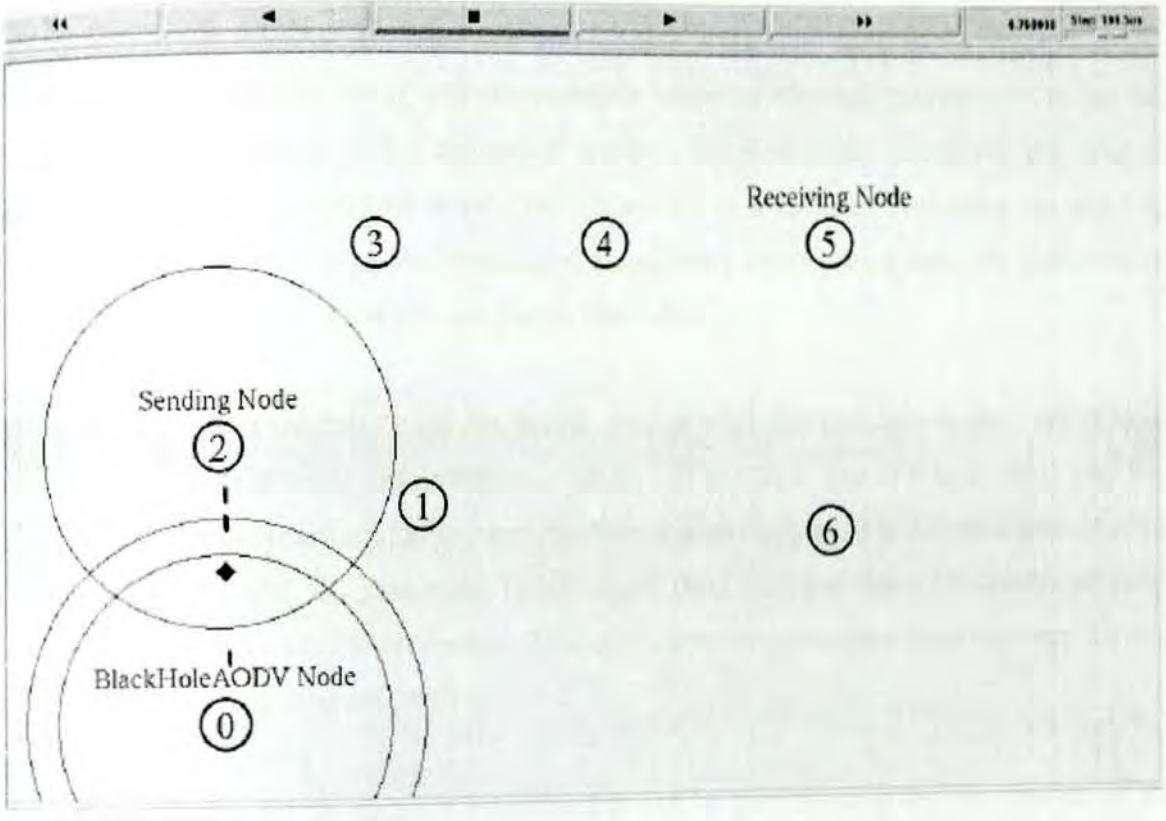

Figure 5.8: A malicious node absorbing traffic

RREP messages that arrive lately at the source node will be rejected although they are from the actual destination node. This is because once a route is established with the node that sent fastest RREP message, the source node saves that route to its routing table for the specified destination and will not change it unless a notification comes in that announces the specified route is broken. This only happens if the intermediate node associated with the specified path leaves the network or changes its position and goes out of the transmission range.

In black hole attack, even if a link that is established through the malicious node is no longer available due to the reasons mentioned above, the adversary can manage to deceive the source node when it tries to broadcast RREQ messages for a new route to the previous destination. Hence, if a network is once intruded or compromised by a malicious node that acts as a black hole node, it is almost impossible to get back to normal routing environment unless the adversary leaves the network by itself. Consequently, one black hole node in a massive network topology can result in a catastrophic damage on the network because of packet loss.

5.2.3. Analyzing Trace Files

The main purpose of simulating different scenarios using the Network Simulator is to get two important files as output of the simulation process. The first is the animation file used to visualize the scenarios using the NAM. The second file is a trace. In evaluating the effect of black hole attack in mobile ad-hoc networks, it is necessary to generate a trace file that contains the audit or record of the whole process during simulation.

Trace files include all events in the simulation such as when the packets are sent, which node generated them, which node has received, which type of packet is sent, if it is dropped why it is dropped etc. In all simulations "new-trace" file format (especially used in wireless networks and includes detailed event) has been used. To get results from the trace files a java parser program and tracegraph was used. The java parser filters the necessary parameters from the trace file that are used in evaluating measured metrics.

The following are a list of quantitative metrics that can be used to assess the performance of the systems routing protocol:

- **Number of data packets sent:** is the number of actual data packets that are sent by source node.
- **Number of data packets received:** is the number of actual data packets received by a destination node.
- **Packet delivery Ratio:** is the ratio of received packets over sent packets in percentage.
- **Normalized routing overhead**: This is the ratio of routing-related transmissions (RREQ, RREP, RERR etc) to data transmissions in a simulation. A transmission is one node either sending or forwarding a packet. Either way, the routing load per unit data successfully delivered to the destination.

- **End to End Delay**: The packet end-to-end delay is the average time in order to traverse the packet inside the network. This includes the time from generating the packet from sender up till the reception of the packet by receiver or destination and expressed in seconds. This includes the overall delay of networks including buffer queues,

transmission time and induced delay due to routing activities. Different application needs different packet delay level.

- **Network throughput**: it is the ratio of total amount of data which reaches the receiver from the sender to the time it takes for the receiver to receive the last packet. It is represented in bits per second or packets per seconds. In MANETs throughput is affected by various changes in topology, limited bandwidth and limited power. Unreliable communication is also one of the factors which adversely affect the throughput parameter.

The value of these matrices will be generated for cases when there is no black hole attack, when there is black hole attack, and when there is black hole attack and prevention method is introduced. This will be measured by varying the network load and mobility of the nodes.

Network load is the total traffic received by the entire network from higher layer which is accepted and queued for transmission. It indicates the quantity of traffic in entire network. It represents the total data traffic in bits per seconds received by the entire network from higher layer accepted and queued for transmission. It does not include any higher layer data traffic rejected without queuing due to large data packet size.

5.2.4. Comparison of Results

Simulation of MANET topology that contains different number of mobile nodes has been performed. According to the explanation in section 5.2.3, we produce animation and trace files for each scenario. In all simulations the random movement of nodes was generated by an inbuilt functionality of NS2 called ***setdest***. A random CBR traffic file was also generated by using in built functionality of NS2 called ***cbrgen.tcl***. The visualization of animation files produced from simulation of MANET topology without black hole attack and MANET topology with black hole attack is shown in Figure 5.9 and 5.10 respectively.

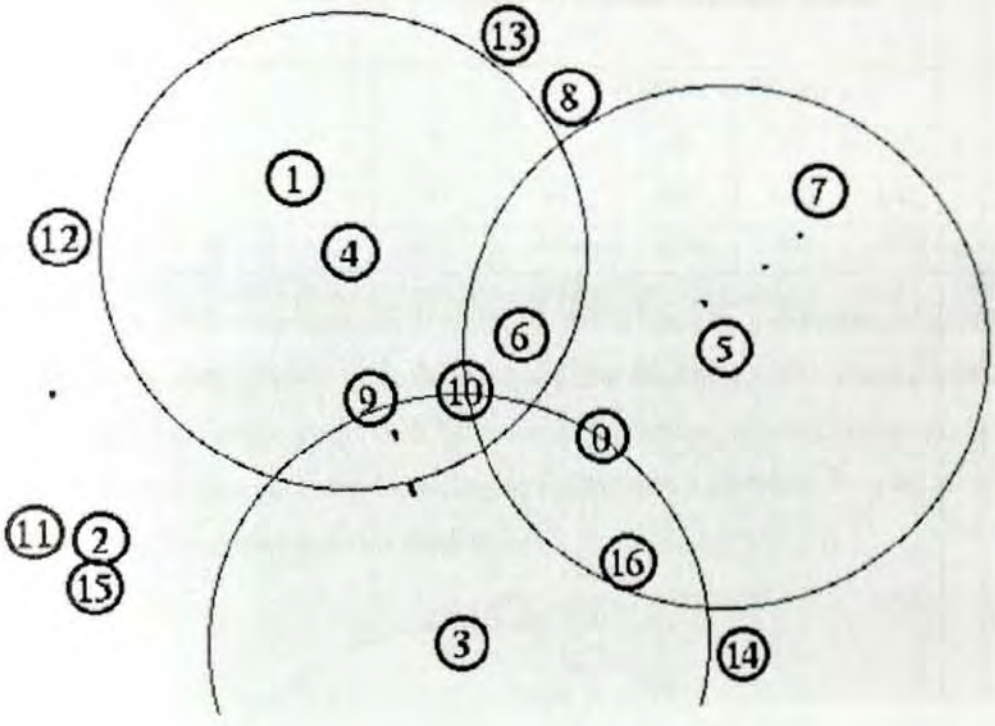

Figure 5.9: Mobile nodes communicating with each other using AODV without black hole attack

Figure 5.9 shows 17 mobile nodes communicating with each other using AODV routing protocol to discover and establish routes between source and destination nodes. In this scenario nodes have been observed cooperating with each other to establish and maintain available routes among themselves. It was also noticed a packet dropping when a node moves out of transmission range in which case a new route discovery will be triggered by the source node.

In the above scenario, there are 8 UDP connections established between 8 sources and 8 destinations. Even numbered nodes including node 0 have been represented as source nodes and odd numbered nodes as destination nodes. We analyzed the trace file (a result of simulation of MANET topology without black hole attack) and extracted the measured metrics from the trace file. The result of measured metrics extracted from the trace file using java parser is as follows in table 5.1:

Table 5.1: The result of measure metrics without black hole attack

Measured Metrics	Number of Nodes					
	7	10	15	17	20	25
Number of data packets sent	241	765	1829	2143	2473	2472
Number of data packets received	239	760	1828	2013	2470	2466

Figure 5.10 depicts a simulation scenario of MANET with a black hole node absorbing network traffic. It is almost the same process with the exception that blackholeAODV routing protocol is used for nodes that exhibits a black hole behavior in the ad-hoc network. After performing exactly the same post simulation tasks, the following results have been found from the average of different scenarios that have one and two black holes:

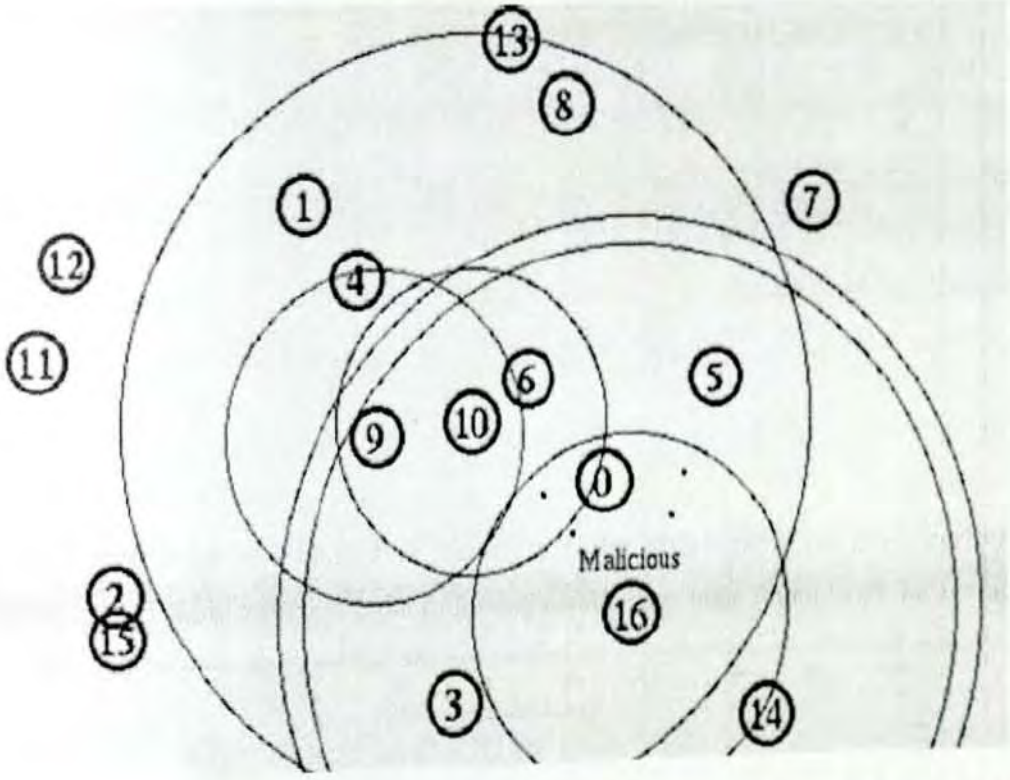

Figure 5.10: A single black hole node absorbing traffic in a MANET topology that uses AODV

Table 5.2: The result of measured metrics with black hole behavior

Measured Metrics	Number of Node					
	7	10	15	17	20	25
Number of data packets sent	243	764	1838	2136	2469	2474
Number of data packets received	72	168	9	173	374	239

Figure 5.11 present graphs generated from the trace files of both simulations for throughput of received packets in a MANET topology without black hole attack and throughput of received packets in a MANET topology under a black hole attack.

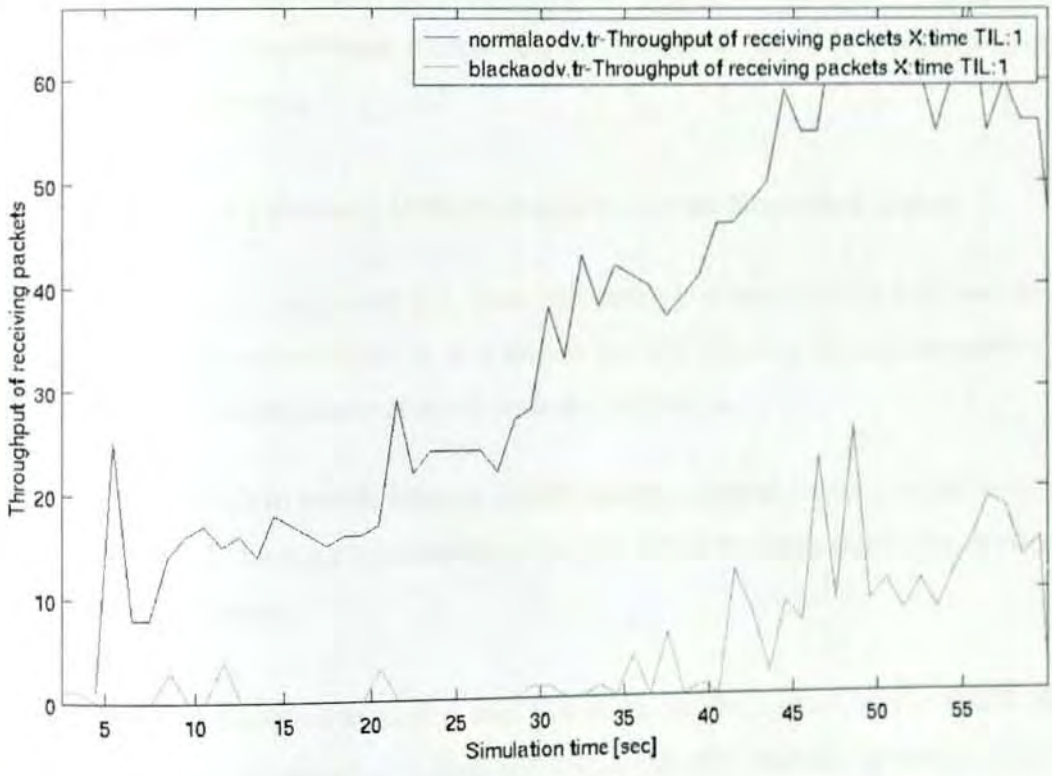

Figure 5.11: Throughput of received packets in MANET with and without black hole attack

From the information revealed by measured metrics in table 5.2 above, it is observed that the number of data packets sent by source nodes and the number of data packets received by destination nodes has a big difference in a topology that contains a malicious node in it. This means, most of the traffic generated by the nodes in the topology was dropped by a misbehaving black hole node preventing packets from reaching to actual destinations. It is also observed from

the table that from the total number of generated packets by source nodes in the network topology, only **8.099251** percent (**91.900749%** packet loss) of them has actually been received by actual destination nodes. In the case of MANET topology simulation with no malicious node, the packet delivery fraction is very high (**99.5%** packet delivery fraction and **0.5%** of packet loss) compared to the previous one.

From the results in the table it can be concluded that although the percentage of packet loss due to a misbehaving malicious node is significantly high, there is still a slight amount of packet loss in a network without any kind of malicious node. This can be due to many factors that affect performance of ad-hoc networks such as a random node movement that can lead to a destination unreachable situation, dependency on a battery power leading to sudden-power-interrupt during communication, poor transmission capacity due to susceptible nature of a wireless channel to several types of interference.

5.2.5. Implementing Enhanced AODV Protocol to Prevent Black Hole Attack

In the previous sections, we explain how black hole attack is implemented in NS2 and the results are obtained from the simulations. In this section we will describe the implementation of the prevention method and the results obtained from the simulation.

In implementing IDAD to enable existing AODV routing protocol in NS2, it was necessary to modify the ***recvReply*** function of the ***aodv.cc*** file. The RREP message entries are taken as audit data for anomaly detection.

The ***recvReply*** was modified in such a way that it has to check every single RREP message against an audit data that has already been collected to identify anomaly detection. False RREP messages from a black hole node usually contain a maximum destination sequence number. Hence, it can be taken as an entry for audit data.

To prevent such poor decision making by a source node or to avoid false positive alarms in detecting intruders, checking RREP messages with a given audit data involves multiple entries of possible anomaly detection parameters before reaching to a conclusion that an RREP message is sent from an intruder. For example, RREP messages that are sent from a black hole node are

generated exactly at the same time the RREQ message is received by the replying malicious node. This gives us timestamps of RREP messages as an additional audit data to be collected in addition to maximum destination sequence numbers.

To insure the perfection of a decision made by a source node that an RREP message is received from a black hole node, all entries of RREP messages can be checked against a set of predefined audit data for possible anomaly conditions.

5.2.6. Testing the IDS-AODV in NS2

Having implemented the IDS-AODV protocol in NS-2, we tried it in a tcl simulation. In the scenario of the simulation there are seven motionless nodes and node positions, the same as in the test simulation of the two RREP messages, shown in Figure 5.10. In this simulation IDS-AODV protocol is used instead of AODV for all nodes except the black hole node (Node 1). To change the AODV protocol to IDS-AODV we only change "*$ns node-config -adhocRouting idsAODV*". When the simulation is compiled, we saw that sending node is sending the messages to receiving node properly. Figure 5.12 shows that CBR packets are reaching the destination node as expected.

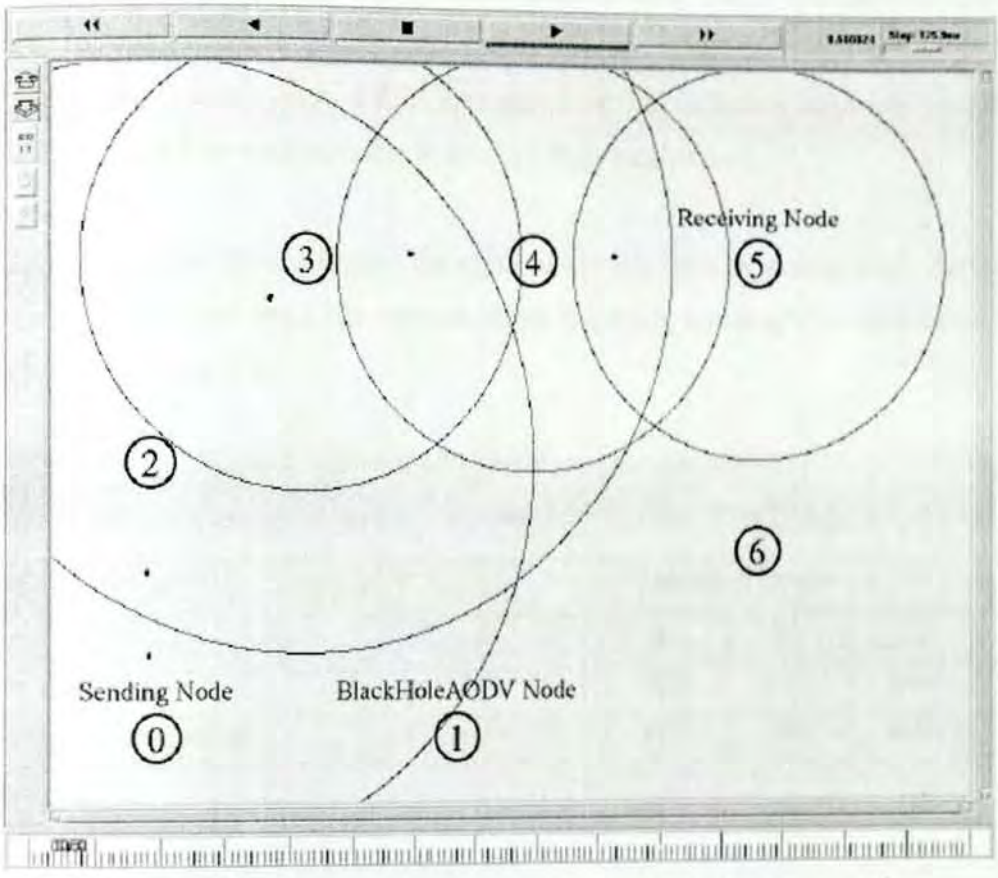

Figure 5.12: CBR packet are reached to destination node properly

In the test simulation, we ensured that the IDSAODV implementation is correctly working. Then, we performed the same simulations on the scenarios we used in section 5.2.4 to compare the performance of IDS approach.

5.2.7. Simulation of IDSAODV and Evaluation of Results

In order to evaluate the effectiveness of IDSAODV, the same MANET topology that was used in section 5.2.4 was used. Once the (IDSAODV was implemented in NS2, it can be used as any other ad-hoc routing protocols to simulate MANET topologies. To do so is as simple as defining mobile nodes to use IDSAODV as their routing protocol rather than the original AODV. This makes every single mobile node that is defined to use IDSAODV as a routing protocol will be able to protect itself from an intruder that tries to perform a black hole attack against the network.

During simulation, although a malicious node keeps sending RREP messages to fool source nodes that have broadcasted RREQ messages to discover route to a destination node, it is observed that source nodes reject RREP messages from the malicious node and accept RREP messages from other intermediate nodes to transmit their data packets.

Analyzing of trace files is done exactly the same way it was done in section 5.2.4. Extraction of desired data from a dump trace file resulted in the following meaningful measurement metrics values as shown in table 5.3.

Table 5.3: The result of measured metrics with IDS

Measured Metrics	Number of Nodes					
	7	10	15	17	20	25
Number of data packets sent	245	757	1823	2155	2456	2470
Number of data packets received	243	753	1822	1963	2451	2463

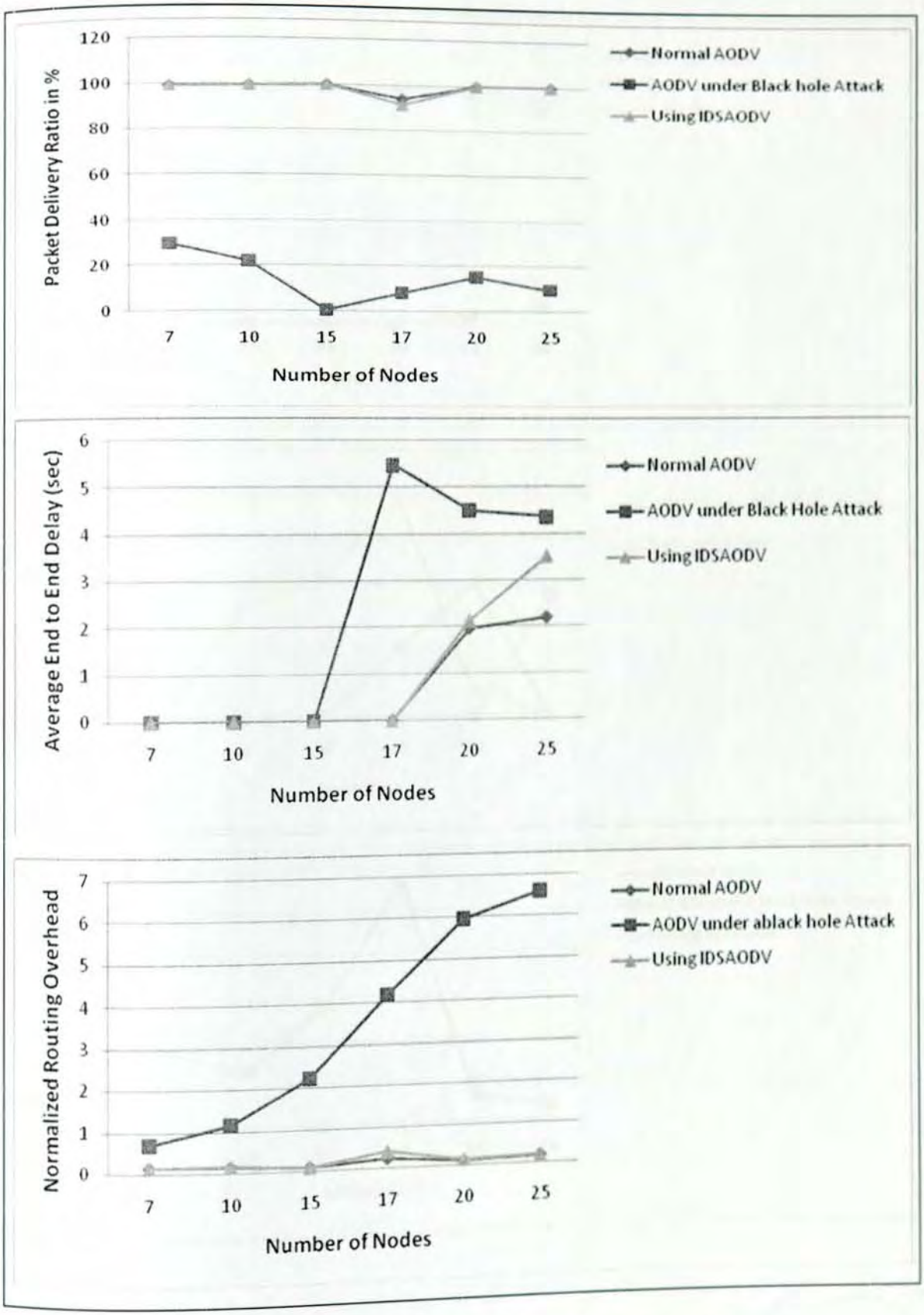

Figure 5.13: Impact of Network size on the performance

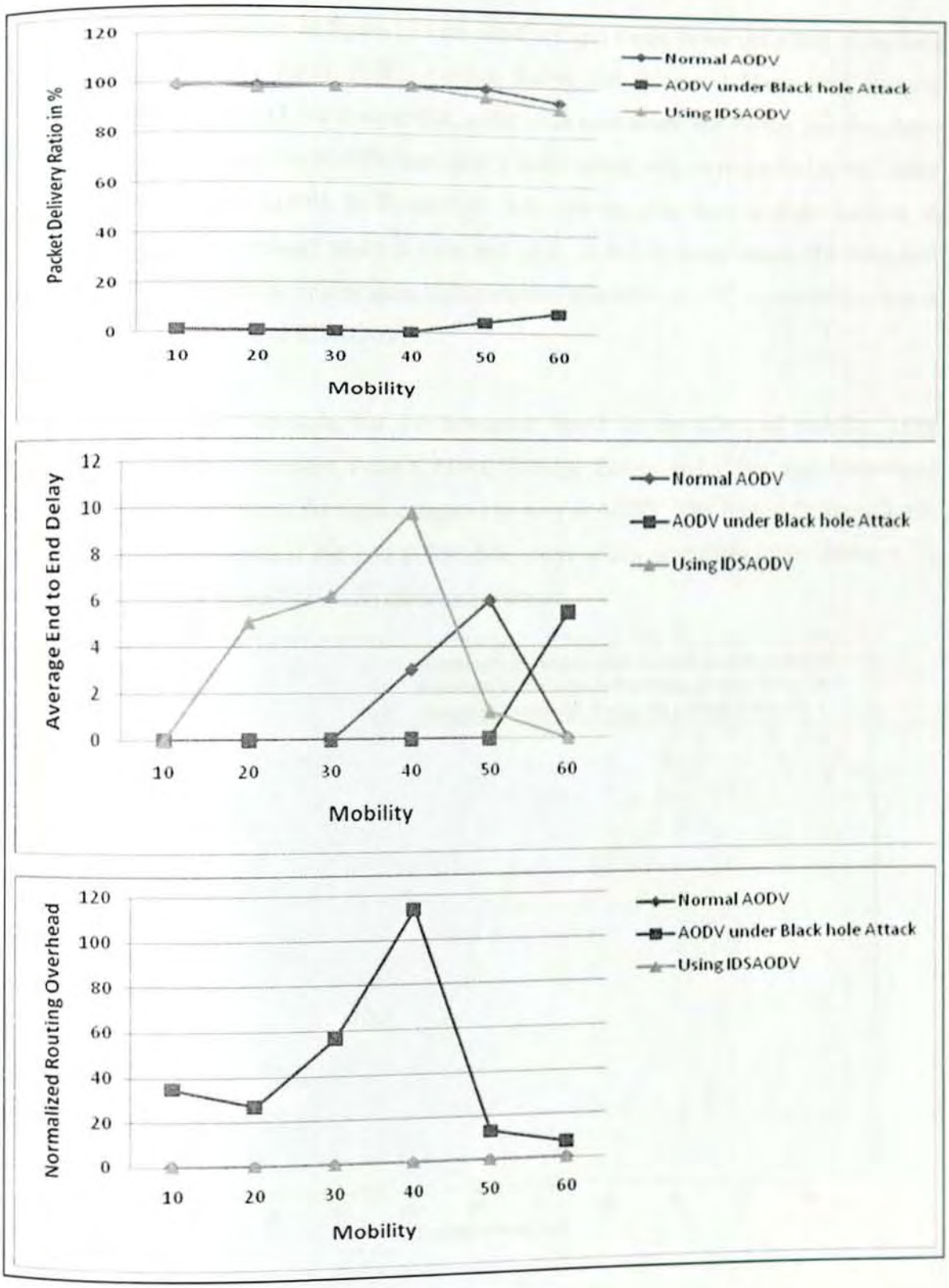

Figure 5.14: Impact of mobility on the performance

All the above three contexts in figure 5.13 are simulated and tested to see the effect of network size on Packet Delivery Ratio(PDR), Average End-to end delay and Normalized Routing Overhead. From figure 5.13, we analyze that, under black hole attack, the Packet Delivery Ratio of IDSAODV is improved by 90-95% than AODV under attack with Average-End-to-end delay almost same as normal AODV. In Figure 5.13, it is observed that there is slight increase in Normalized Routing Overhead, which is quite negligible. In AODV under attack, the delay will be less and routing overhead will be quite high compared to normal AODV, so our comparison is between normal AODV and IDSAODV.

From figure 5.14 we conclude that the simulation based on the effect of mobility using IDSAODV on Packet Delivery Ratio (PDR), Average End-to end delay and Normalized Routing Overhead is almost the same compared to normal AODV. The Packet Delivery Ratio, Normalized routing overhead and End to End delay stays within acceptable limits almost 4-5% lower than it should normally be with minimum overhead.

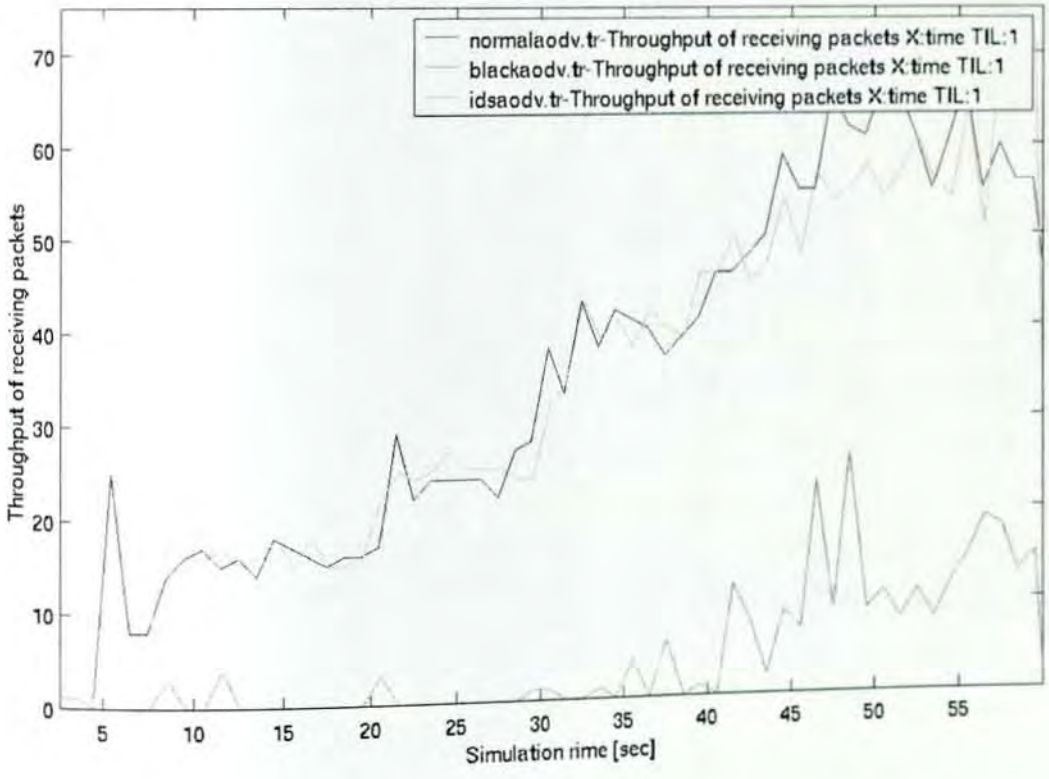

Figure 5.15: Throughput of received packets using AODV, BlackAODV, and IDS-AODV

From the information in table 5.3 and figure 5.15 above, it can be concluded that IDAD enhances the network throughput by preventing the packet loss that occurs due to the black hole attack. The prevention technique is also efficient in such a way that the number of generated routing packets (control messages) is lower which results in minimization of bandwidth consumption by routing packets.

CHAPTER SIX: CONCLUSION AND FUTURE WORK

6.1. Conclusion

In this project, it was discussed that any secure network will have vulnerability that an adversary could exploit. More specifically, on-demand ad-hoc routing protocols (such as AODV and DSR) have a vulnerability that could be exploited by malicious nodes to perform the well known black hole attack on MANETs.

We evaluated effect of the black hole attack in an ad hoc network by implementing on AODV protocol with nodes that behave as black hole in NS-2. We simulated different scenarios where each one use AODV protocol and also simulated the same scenarios after introducing one or more black hole nodes into the network. Moreover, we also implemented the prevention method (IDAD) that attempted to reduce the black hole effects in NS-2 and simulated the solution using the same scenarios.

During the simulation, we saw that the packet loss is increased in the ad-hoc network when we introduce a black hole attack. Furthermore, increase in end to end delay and routing packets, and decreases the network throughput were also evident. This shows that black hole attack affects the overall network connectivity in mobile ad hoc networks. So in this case, if the number of black hole nodes is increased then the data loss would also increase as expected.

The result shown in table 5.3 of section 5.2.7 indicates that IDAD enhances the network throughput by preventing the packet loss that occurs due to the black hole attack. The result of the simulation implementation has proved the effectiveness of the prevention method (IDAD) in preventing black hole attack against MANETs as well as maximizing performance by decreasing the number of routing packets.

6.2. Future Work

Black hole attack is one of the many security threats that exist in mobile ad-hoc networks. As discussed in different section of this project paper, there are many other security threats that affect the performance of wireless communication especially in MANETs.

Therefore, the author of this project has intention of doing similar works to implement the new technique to prevent other security threats that exist in MANETs.

References

[1] Larsson T. and Hedman N. Routing protocols in wireless ad-hoc networks - A simulation study, Master thesis, Lulea University, Stockholm, Sweden, 1998.

[2] Ullah I. and Rehman S. Analysis of Black Hole attack On MANETs Using different MANET Routing Protocols. Masters Thesis. Blekinge Institute of Technology. Sweden, 2010.

[3] Jhaveri H., Patel D., Parmar D., Shah I. MANET Routing Protocols and Wormhole Attack against AODV. IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010

[4] Akanksha Saini, Harish Kumar. Comparison between various black hole detection techniques in MANET. NCCI 2010 -National Conference on Computational Instrumentation CSIO Chandigarh, INDIA, 19-20 March 2010.

[5] Bernat A. Simulation of routing protocol for ad-hoc networks in NS-2, Master thesis, November 2006, Delft University of Technology, Delft, the Netherlands.

[6] Boursier A., Dahlen S., Fran J., Marin T. and Nethi S. Cross-layer approach in mobile ad-hoc routing, Master thesis, December 2006, Aalborg University, Aalborg, Denmark.

[7] Kim Y., Moon I. and Cho S. A comparison of improved AODV routing protocol based on IEEE 802.11 and IEEE 802.15.4, Journal of Engineering Science and Technology Vol. 4, No. 2 (2009) 132 – 141.

[8] Chakeres I. and Royer E. AODV routing protocol implementation design, Depaprtment of Electrical & Computer Engineering University of California, Santa Barbara.

[9] Pan Y. Design Routing Protocol Performance Comparison in NS2: AODV comparing to DSR as Example, Department of Computer Science SUNY Binghamton Vestal Parkway East, Vestal, NY 13850.

[10] Royer E. and C. Toh C. A Review of current routing protocols for ad-hoc mobile wireless networks, University of California, Santa Barbara.

[11] Garg N. and Mahapatra R. MANET security issues, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.8, August (2009) 241-246.

[12] Kärpijoki V. Security in ad-hoc networks, Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology.

[13] Bayya A., Gupte S., Shukla Y. and Garikapati A. Security in ad-hoc networks, Computer Science Department, University of Kentucky.

[14] Yi P., Dai Z., Zhang S. and Zhong Y. A new routing attack in mobile ad-hoc networks, Sixth Annual International Conference on Mobile Computing and Networking (MobiCom'2000), Boston, August 6-11, 2000.

[15] Deng H., Li W., and Agrawal D. Routing security in wireless ad-hoc network, IEEE Communications Magazine, vol. 40, no. 10 (2002) 70-75.

[16] Ruiz J., Friginal J., Andrés D. and Gil P. Black Hole Attack Injection in Ad hoc Networks, Fault Tolerance Systems Group (GSTF), Instituto de las TIC Avanzadas (ITACA) Universidad Politécnica de Valencia, Campus de Vera s/n, E-46022, Valencia, Spain.

[17] Tamilselvan L. and Sankaranarayanan V. Prevention of impersonation attack in wireless mobile ad-hoc networks, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.3, March (2007) 118-123.

[18] Hussain M. Passive and active attacks against wireless LAN, University of Hertfordshire, England, U.K.

[19] The NS Manual, April 28, 2008.

[20] Dokurer S. Simulation of Black Hole Attack in Wireless Ad-hoc Networks, Master thesis, September 2006, Atilim University, Turkey.

[21] Sadasivam K. Tutorial for Simulation-based Performance Analysis of MANET Routing Protocols in ns-2, Tutorial at http://www.sce.uhcl.edu/yang/teaching/. May 1, 2011

[22] Ros F. and Ruiz P. Implementing a new MANET unicast routing protocol in NS2, Dept. of Information and Communications Engineering, University of Murcia December, 2004.

[23] Zhang Y. and Lee W. Intrusion detection in wireless ad-hoc networks, Sixth Annual International Conference on Mobile Computing and Networking (MobiCom'2000), Boston, August 6-11, 2000.

[24] Caballero E. Vulnerabilities of intrusion detection systems in mobile ad-hoc networks – The routing problem, Helsinki University of Technology.

[25] Krishnan M. Intrusion detection in wireless sensor networks, Project Paper, The University of California at Berkley, 2006.

[26] Ioannis K., Dimitriou T. and Freiling F. Towards intrusion detection in wireless sensor networks, 13th European Wireless Conference, Paris, April 2007.

[27] Scarfone K. and Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS), Recommendations of the National Institute of Standards and Technology, February 2007.

[28] Chen T. Intrusion Detection for Mobile Ad-Hoc Networks, SMU, Department of Electrical Engineering, talk at Rockwell Collins, May 20, 2004

[29] Sharma S. and Gupta R. Simulation study of black hole attack in the mobile ad-hoc networks, Journal of Engineering Science and Technology, Vol. 4, No. 2 (2009) 243-250.

[30] Weerasinghe H. and Fu H. Preventing Cooperative Black Hole Attacks in Mobile Ad Hoc Networks: Simulation Implementation and Evaluation, International Journal of Software Engineering and Its Applications, Vol. 2, No. 3 (2008) 39-54.

[31] Raj P. and Swadas P. A dynamic learning system against black hole attack in AODV based MANET, IJCSI International Journal of Computer Science, Vol. 2, (2009) 54-59.

[32] Zhou L. and Haas Z. "Securing Ad Hoc Networks," IEEE Net., vol. 13, no. 6, Nov./Dec. 1999.

[33] Lundberg J. Routing Security in Ad Hoc Networks. Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, 2000.

[34] Alem, Y.F. Zhao Cheng Xuan. Preventing Black Hole Attack in Mobile Ad-hoc Networks Using Anomaly Detection. Future Computer and Communication (ICFCC), 2nd International Conference, Wuhan, 21-24 May 2010.

[35] C.Perkins, "(RFC) Request for Comments – 3561", Category: Experimental, Network, Working Group, July 2003.

Appendix A: Tcl Script Used in Simulating blackholeAODV and IDSAODV

```
# Define options
      set val(chan) Channel/WirelessChannel      ;#Channel Type
      set val(prop) Propagation/TwoRayGround  ;# radio-propagation model
      set val(netif) Phy/WirelessPhy             ;# network interface type
      set val(mac) Mac/802_11                ;# MAC type
      set val(ifq) Queue/DropTail/PriQueue       ;# interface queue type
      set val(ll) LL                         ;# link layer type
      set val(ant) Antenna/OmniAntenna       ;# antenna model
      set val(ifqlen) 150                    ;# max packet in ifq
      set val(nn) 17                         ;# total number of normal
mobilenodes
      set val(rp) AODV                       ;# routing protocol
      set val(x) 500                         ;# X dimension of topography
      set val(y) 500                         ;# Y dimension of topography
      set val(cstop) 59                      ;# time of connections end
      set val(stop) 60                       ;# time of simulation end
      set val(cp) "scenarios/scen1-n17-t60-x500-y500" ;#Connection Pattern
      set val(cc) "scenarios/cbr-1-16"                ;#CBR Connections
# Initialize Global Variables
      set ns_ [new Simulator]
      $ns_ use-newtrace
      set tracefd [open idsaodv.tr w]
      $ns_ trace-all $tracefd
      set namtrace [open idsaodv.nam w]
      $ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
# set up topography object
      set topo [new Topography]
      $topo load_flatgrid $val(x) $val(y)
# Create God
      create-god $val(nn)
# Create channel #1
      set chan_1_ [new $val(chan)]
# configure node, please note the change below.
      $ns_ node-config -adhocRouting $val(rp)
            -llType $val(ll) \
            -macType $val(mac) \
            -ifqType $val(ifq) \
            -ifqLen $val(ifqlen) \
            -antType $val(ant) \
            -propType $val(prop) \
```

```
            -phyType $val(netif) \
            -topoInstance $topo \
            -agentTrace ON \
            -routerTrace ON \
            -macTrace ON \
            -movementTrace ON \
            -channel $chan_1_
# Creating mobile nodes for simulation
      puts "Creating nodes..."
      $ns_ node-config -adhocRouting idsAODV
      for {set i 1} {$i < $val(nn)} {incr i} {
      set node_($i) [$ns_ node]
      $node_($i) random-motion 0 ;#disable random motion
}
# Creating Malicious node for simulation
$ns_ node-config -adhocRouting blackholeAODV
      set node_(0) [$ns_ node]
      $ns_ at 0.0 "$node_(0) label \"Black hole node\""
      $node_(0) random-motion 0 ;#disable random motion
# Adding connection pattern which is created using setdest, parameters shown
below
# ./setdest -n 17 -p 1.0 -M 20.0 -t 60 -x 500 -y 500 > scen1-n17-t60-x500-y500
      puts "Loading random connection pattern..."
      set god_ [God instance]
      source $val(cp)
# CBR Connections generated by ns cbrgen.tcl -type cbr -nn 16 -seed 1.0 -mc 16
-rate 8.0 > cbr-1-16
      source $val(cc)
# Define initial node position
      for {set i 0} {$i < $val(nn) } {incr i} {
      $ns_ initial_node_pos $node_($i) 30
}
# Define initial malicious node position
      #$ns_ initial_node_pos $node_(6) 30

# Tell all nodes when the simulation ends
      for {set i 0} {$i < $val(nn) } {incr i} {
      $ns_ at $val(stop).000000001 "$node_($i) reset";
}
# Tell all nodes when the simulation ends
      $ns_ at $val(stop).000000001 "$node_(0) reset";
# Ending nam and simulation
```

```
    $ns_ at $val(stop) "finish"
    $ns_ at $val(stop).0 "$ns_ trace-annotate \"Simulation has ended\""
    $ns_ at $val(stop).0 "puts \"NS EXITING...\" ; $ns_ halt"
    proc finish {} {
    global ns_ tracefd namtrace
    $ns_ flush-trace
    close $tracefd
    close $namtrace
    exec nam idsaodv.nam &
    exit 0
}
    puts "Starting Simulation..."
    $ns_ at 0.0 "$ns_ set-animation-rate 5ms"
$ns_ run
```

Appendix B: Java Parser

```
import java.util.*;
import java.lang.*;
import java.io.*;
 public class parsetrace {
 public static void main (String args[]) {
    String s, thisLine, currLine, thisLine1;
    int j=0;
    FileInputStream fin, fin1;
    FileOutputStream fout, fout1;
    final int FILES = 45;
    final int MAX_PACKETS = 1000000;
      try {
           int i=0, sends=0, receives=0;
           int drops=0, packet_id=0, highest_packet_id = 0;
           int line_count=0, current_line=0, routing_packets=0;
           int count=0;
           float pdfraction, time=0, packet_duration=0, end_to_end_delay=0;
           float avg_end_to_end_delay=0;
           float start_time[] = new float[MAX_PACKETS];
           float end_time[] = new float[MAX_PACKETS];
           float sent_packets[] = new float[MAX_PACKETS];
           float received_packets[] = new float[MAX_PACKETS];
           String tokens[] = new String[100];// initialize the start time
           for (i=0; i<MAX_PACKETS ; i++)
           start_time[i] = 0;
           fout = new FileOutputStream ("traceoutput");
           DataOutputStream op = new DataOutputStream(fout);
                for (int h=0;h<1;h++) // for (int h=0;h<FILES+1;h++)
                 {
                  j=0;
                  sends=0; receives=0; routing_packets=0;
                highest_packet_id = 0;
                end_to_end_delay=0;
                for (i=0; i<MAX_PACKETS ; i++)
                {
                start_time[i] = 0; end_time[i]=0;
                }
                fin = new FileInputStream ("normalaodv.tr");
                DataInputStream br = new DataInputStream(fin);
                while ((thisLine = br.readLine()) != null )
                {
```

```
                // scan it line by line
                java.util.StringTokenizer st = new
java.util.StringTokenizer(thisLine, " ");
                i=0;
                while(st.hasMoreElements())
                tokens[i++]= st.nextToken();
                if (tokens[0].equals("s") || tokens[0].equals("r") ||
tokens[0].equals("f"))
                {
                // parse the time
                if (tokens[1].equals("-t"))
                time = Float.valueOf(tokens[2]).floatValue();
                // parse the packet id
                if(tokens[39].equals("-Ii"))
                packet_id = Integer.valueOf(tokens[40]).intValue();
                // calculate the sent packets

     if(tokens[0].equals("s")&&tokens[18].equals("AGT")&&tokens[34].equals("c
br"))
          sends++;
          // find the number of packets in the simulation
          if (packet_id >highest_packet_id)
          highest_packet_id = packet_id;
          // set the start time, only if it' s not already set
          if (start_time[packet_id] == 0)
          start_time[packet_id] = time;
          // calculate the receives and end-end delay
          if (tokens[0].equals("r") && tokens[18].equals("AGT") &&
tokens[34].equals("cbr"))
          {
          receives++;
          end_time[packet_id] = time;
          }
          else end_time[packet_id] = -1;
          // calculate the routing packets
      if ((tokens[0].equals("s") || tokens[0].equals("f")) &&
tokens[18].equals("RTR") && (tokens[34].equals("AODV") ||
tokens[34].equals("message") ))
                routing_packets++;
                     }
                }
          }
```

```
// calculate the packet duration for all the packets
 for (packet_id = 0; packet_id <= highest_packet_id ; packet_id++)
      {
packet_duration = end_time[packet_id] - start_time[packet_id];
if (packet_duration >0)
end_to_end_delay += packet_duration;
      }
// calculate the average end-end packet delay
avg_end_to_end_delay = end_to_end_delay / (receives );
// calculate the packet delivery ratio
dfraction = ((float)receives/(float)(sends))*100;
System.out.print(" "+sends);
System.out.print(" "+receives);
System.out.print(" "+ routing_packets);
System.out.print(" "+(float)routing_packets/(float)receives);
System.out.print(" "+pdfraction);
System.out.print(" "+avg_end_to_end_delay);
System.out.println("");
op.writeBytes(" "+sends);
op.writeBytes(" "+receives);
op.writeBytes(" "+ routing_packets);
op.writeBytes(" "+(float)routing_packets/(float)receives);
op.writeBytes(" "+pdfraction);
op.writeBytes(" "+avg_end_to_end_delay);
op.writeChar('\n');
}
catch (Exception e) {
e.printStackTrace();
}
    }
}
```

Appendix C: Trace File Field Types

Field 0: event type

s: send r: receive d: drop f: forward

Filed 1: General tag

-t: time

Field 2: Next hop info

-Hs: id for this node

-Hd: id for next hop towards the destination

Field 3: Node property type tag

-Ni: node id

-Nx –Ny -Nz: node's x/y/z coordinate

-Ne: node energy level

-Nl: trace level, such as AGT, RTR, MAC

-Nw: reason for the event

Field 4: packet info at MAC level

-Ma: duration

-Md: dest's ethernet address

-Ms: src's ethernet address

-Mt: ethernet type

Field 5: Packet information at IP level

-Is: source address. Source port number

-Id: dest address.dest port number

-It: packet type

-Il: packet size

-If: flow id

-Ii: unique id

-Iv: ttl value

Field 6: Packet info at "Application level" ARP, TCP, CBR, the type of ad-hoc routing protocol like

DSDV, DSR, AODV etc. The field consists of a leading –P and the list of tags for different applications. Values for AODV and CBR are described below;

For AODV :

-Pt : Control message type,

-Ph: Hop-count,

-Pb: Broadcast-id,

-Pd: Destination,

-Pds: Dest Seqno,

-Ps: Source,

-Pss: Source Seqno

-Pl: Lifetime.

-Pc: Pkt Type, REPLY/ERROR

For CBR :

-Pn: This denotes the application of "CBR"

-Pi: sequence number

-Pf: how many times this pkt was forwarded

-Po: optimal number of forwards

Appendix D: Sample Trace File

M 1.00000 16 (254.12, 193.83, 0.00), (119.79, 436.41), 18.05

M 1.00000 0 (284.34, 126.54, 0.00), (440.90, 399.25), 1.37

M 1.00000 1 (374.38, 423.53, 0.00), (463.03, 188.25), 13.08

M 1.00000 2 (73.10, 394.72, 0.00), (397.03, 287.55), 7.07

M 1.00000 3 (302.83, 364.35, 0.00), (278.48, 199.89), 16.50

M 1.00000 4 (141.48, 439.30, 0.00), (394.45, 83.70), 13.30

M 1.00000 5 (271.14, 461.19, 0.00), (249.87, 335.60), 3.24

M 1.00000 6 (272.47, 64.14, 0.00), (1.48, 348.31), 18.96

M 1.00000 7 (486.10, 72.55, 0.00), (68.50, 54.16), 7.91

M 1.00000 8 (283.94, 55.91, 0.00), (346.41, 156.80), 1.28

M 1.00000 9 (149.45, 394.31, 0.00), (111.14, 114.49), 3.90

M 1.00000 10 (465.34, 478.64, 0.00), (263.14, 62.07), 14.26

M 1.00000 11 (316.72, 86.30, 0.00), (42.86, 325.63), 5.49

M 1.00000 12 (380.93, 315.07, 0.00), (191.80, 214.24), 17.26

M 1.00000 13 (342.88, 203.29, 0.00), (376.93, 177.17), 6.80

M 1.00000 14 (152.02, 88.26, 0.00), (219.87, 344.61), 15.26

M 1.00000 15 (128.24, 13.85, 0.00), (12.54, 477.97), 9.95

s -t 2.556838879 -Hs 0 -Hd -2 -Ni 0 -Nx 285.40 -Ny 128.39 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 1.0 -It cbr -Il 512 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

r -t 2.556838879 -Hs 0 -Hd -2 -Ni 0 -Nx 285.40 -Ny 128.39 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 1.0 -It cbr -Il 512 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

s -t 2.556838879 -Hs 0 -Hd -2 -Ni 0 -Nx 285.40 -Ny 128.39 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -

Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

s -t 2.556953879 -Hs 0 -Hd -2 -Ni 0 -Nx 285.40 -Ny 128.40 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 106 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802026 -Hs 11 -Hd -2 -Ni 11 -Nx 310.28 -Ny 91.93 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802060 -Hs 6 -Hd -2 -Ni 6 -Nx 252.08 -Ny 85.52 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802115 -Hs 8 -Hd -2 -Ni 8 -Nx 284.99 -Ny 57.61 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802195 -Hs 13 -Hd -2 -Ni 13 -Nx 351.28 -Ny 196.84 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802214 -Hs 16 -Hd -2 -Ni 16 -Nx 240.50 -Ny 218.43 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802307 -Hs 14 -Hd -2 -Ni 14 -Nx 158.10 -Ny 111.24 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802506 -Hs 12 -Hd -2 -Ni 12 -Nx 357.21 -Ny 302.42 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802509 -Hs 15 -Hd -2 -Ni 15 -Nx 124.49 -Ny 28.89 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802534 -Hs 7 -Hd -2 -Ni 7 -Nx 473.78 -Ny 72.00 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557802582 -Hs 3 -Hd -2 -Ni 3 -Nx 299.06 -Ny 338.93 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827026 -Hs 11 -Hd -2 -Ni 11 -Nx 310.28 -Ny 91.93 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P

aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827060 -Hs 6 -Hd -2 -Ni 6 -Nx 252.08 -Ny 85.52 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827115 -Hs 8 -Hd -2 -Ni 8 -Nx 284.99 -Ny 57.61 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827195 -Hs 13 -Hd -2 -Ni 13 -Nx 351.28 -Ny 196.84 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827214 -Hs 16 -Hd -2 -Ni 16 -Nx 240.50 -Ny 218.43 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

s -t 2.557827214 -Hs 16 -Hd 0 -Ni 16 -Nx 240.50 -Ny 218.43 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 16.255 -Id 0.255 -It AODV -Il 44 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x4 -Ph 1 -Pd 1 -Pds -1 -Pl 10.000000 -Pc REPLY

r -t 2.557827307 -Hs 14 -Hd -2 -Ni 14 -Nx 158.10 -Ny 111.24 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827506 -Hs 12 -Hd -2 -Ni 12 -Nx 357.21 -Ny 302.42 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827509 -Hs 15 -Hd -2 -Ni 15 -Nx 124.49 -Ny 28.89 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827534 -Hs 7 -Hd -2 -Ni 7 -Nx 473.78 -Ny 72.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 2.557827582 -Hs 3 -Hd -2 -Ni 3 -Nx 299.06 -Ny 338.93 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -Is 0.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 1 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

s -t 2.557922214 -Hs 16 -Hd -2 -Ni 16 -Nx 240.50 -Ny 218.43 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 10 -Mt 806 -P arp -Po REQUEST -Pms 16 -Ps 16 -Pmd 0 -Pd 0

r -t 2.558610549 -Hs 0 -Hd -2 -Ni 0 -Nx 285.40 -Ny 128.40 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 10 -Mt 806 -P arp -Po REQUEST -Pms 16 -Ps 16 -Pmd 0 -Pd 0

Appendix E: IDSAODV C++ Code

```
void
idsAODV::recvRequest(Packet *p) {
      struct hdr_ip *ih = HDR_IP(p);
      struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
      idsaodv_rt_entry *rt;
           if(rq->rq_src == index) {
#ifdef DEBUG
            fprintf(stderr, "%s: got my own REQUEST\n", __FUNCTION__);
#endif // DEBUG
            Packet::free(p);
            return;
      }
      if (id_lookup(rq->rq_src, rq->rq_bcast_id)) {
#ifdef DEBUG
            fprintf(stderr, "%s: discarding request\n", __FUNCTION__);
#endif // DEBUG
            Packet::free(p);
            return;
      }
      id_insert(rq->rq_src, rq->rq_bcast_id);
      idsaodv_rt_entry *rt0; // rt0 is the reverse route
      rt0 = rtable.rt_lookup(rq->rq_src);
      if(rt0 == 0) { /* if not in the route table */
            // create an entry for the reverse route.
            rt0 = rtable.rt_add(rq->rq_src);
      }
      rt0->rt_expire = max(rt0->rt_expire, (CURRENT_TIME + REV_ROUTE_LIFE));
      if ( (rq->rq_src_seqno > rt0->rt_seqno ) ||
           ((rq->rq_src_seqno == rt0->rt_seqno) &&
            (rq->rq_hop_count < rt0->rt_hops)) ) {
            // If we have a fresher seq no. or lesser #hops for the
            // same seq no., update the rt entry.
            rt_update(rt0, rq->rq_src_seqno, rq->rq_hop_count, ih->saddr(),
                  max(rt0->rt_expire, (CURRENT_TIME + REV_ROUTE_LIFE)) );
            if (rt0->rt_req_timeout > 0.0) {
                  // Reset the soft state and
                  // Set expiry time to CURRENT_TIME + ACTIVE_ROUTE_TIMEOUT
                  // This is because route is used in the forward direction,
                  // but only sources get benefited by this change
                  rt0->rt_req_cnt = 0;
```

```
                rt0->rt_req_timeout = 0.0;
                rt0->rt_req_last_ttl = rq->rq_hop_count;
                rt0->rt_expire = CURRENT_TIME + ACTIVE_ROUTE_TIMEOUT;
            }
            assert (rt0->rt_flags == RTF_UP);
            Packet *buffered_pkt;
            while ((buffered_pkt = rqueue.deque(rt0->rt_dst))) {
                if (rt0 && (rt0->rt_flags == RTF_UP)) {
                    assert(rt0->rt_hops != INFINITY2);

                    forward(rt0, buffered_pkt, NO_DELAY);
                }
            }
        }
        // End for putting reverse route in rt table
        rt = rtable.rt_lookup(rq->rq_dst);
        // First check if I am the destination ..
        if(rq->rq_dst == index) {
#ifdef DEBUG
            fprintf(stderr, "%d - %s: destination sending reply\n",
                    index, __FUNCTION__);
#endif // DEBUG

            // Just to be safe, I use the max. Somebody may have
            // incremented the dst seqno.
            seqno = max(seqno, rq->rq_dst_seqno)+1;
            if (seqno%2) seqno++;
            sendReply(rq->rq_src,           // IP Destination
                      1,                    // Hop Count
                      index,                // Dest IP Address
                      seqno,                // Dest Sequence Num
                      MY_ROUTE_TIMEOUT,     // Lifetime
                      rq->rq_timestamp);    // timestamp
            Packet::free(p);
        }
        // I am not the destination, but I may have a fresh enough route.
        else if (rt && (rt->rt_hops != INFINITY2) &&
                 (rt->rt_seqno >= rq->rq_dst_seqno) ) {
            //assert (rt->rt_flags == RTF_UP);
            assert(rq->rq_dst == rt->rt_dst);
            //assert ((rt->rt_seqno%2) == 0);   // is the seqno even?
            sendReply(rq->rq_src,
```

```
                    rt->rt_hops + 1,
                    rq->rq_dst,
                    rt->rt_seqno,
                    (u_int32_t) (rt->rt_expire - CURRENT_TIME),
                    //            rt->rt_expire - CURRENT_TIME,
                    rq->rq_timestamp);
            // Insert nexthops to RREQ source and RREQ destination in the
            // precursor lists of destination and source respectively
            rt->pc_insert(rt0->rt_nexthop); // nexthop to RREQ source
            rt0->pc_insert(rt->rt_nexthop); // nexthop to RREQ destination
#ifdef RREQ_GRAT_RREP
            sendReply(rq->rq_dst,
                    rq->rq_hop_count,
                    rq->rq_src,
                    rq->rq_src_seqno,
                    (u_int32_t) (rt->rt_expire - CURRENT_TIME),

                    //            rt->rt_expire - CURRENT_TIME,
                rq->rq_timestamp);
#endif
            Packet::free(p);
        }
        /*
         * Can't reply. So forward the  Route Request
         */
        else {
            ih->saddr() = index;
            ih->daddr() = IP_BROADCAST;
            rq->rq_hop_count += 1;
            // Maximum sequence number seen en route
            if (rt) rq->rq_dst_seqno = max(rt->rt_seqno, rq->rq_dst_seqno);
            forward((idsaodv_rt_entry*) 0, p, DELAY);
        }
}
void
idsAODV::recvReply(Packet *p) {
//struct hdr_cmn *ch = HDR_CMN(p);
        struct hdr_ip *ih = HDR_IP(p);
        struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);
        idsaodv_rt_entry *rt;
        char suppress_reply = 0;
        double delay = 0.0;
```

```
        int count;
        idsBroadcastRREP *r = rrep_lookup(rp->rp_dst);
#ifdef DEBUG
        fprintf(stderr, "%d - %s: received a REPLY\n", index, __FUNCTION__);
#endif // DEBUG
#if 0
        if (ih->daddr() == index) {
                if (r == NULL) {
                        rrep_insert(rp->rp_dst);
                        Packet::free(p);
                        return;
                } else
                        rrep_remove(rp->rp_dst);
        }
#endif
        if (r == NULL) {
                count = 0;
                rrep_insert(rp->rp_dst);
        } else {
                r->count++;
                count = r->count;
        }


        // Note that rp_dst is the dest of the data packets, not the
        // the dest of the reply, which is the src of the data packets.
        rt = rtable.rt_lookup(rp->rp_dst);
        /*
         *  If I don't have a rt entry to this host... adding
         */
        if(rt == 0) {
                rt = rtable.rt_add(rp->rp_dst);
        }
        if ( count > 1 ||
             (rt->rt_seqno < rp->rp_dst_seqno) ||   // newer route
             ((rt->rt_seqno == rp->rp_dst_seqno) &&
              (rt->rt_hops > rp->rp_hop_count)) ) { // shorter or better route
              // Update the rt entry
              rt_update(rt, rp->rp_dst_seqno, rp->rp_hop_count,
                        rp->rp_src, CURRENT_TIME + rp->rp_lifetime);
              rt->rt_req_cnt = 0;
              rt->rt_req_timeout = 0.0;
```

```
            rt->rt_req_last_ttl = rp->rp_hop_count;
            if (ih->daddr() == index) { // If I am the original source
                  // Update the route discovery latency statistics
                  // rp->rp_timestamp is the time of request origination
      rt->rt_disc_latency[(unsigned char)rt->hist_indx] = (CURRENT_TIME - rp-
>rp_timestamp)
                        / (double) rp->rp_hop_count;
                  // increment indx for next time
                  rt->hist_indx = (rt->hist_indx + 1) % MAX_HISTORY;
            }
            /*
             * Send all packets queued in the sendbuffer destined for
             * this destination.
             * XXX - observe the "second" use of p.
             */
            Packet *buf_pkt;
            while((buf_pkt = rqueue.deque(rt->rt_dst))) {
                  if(rt->rt_hops != INFINITY2) {
                        assert (rt->rt_flags == RTF_UP);
                        // Delay them a little to help ARP. Otherwise ARP
                        // may drop packets.
                        forward(rt, buf_pkt, delay);
                        delay += ARP_DELAY;
                  }
            }
      } else {
            suppress_reply = 1;
      }
      /*
       * If reply is for me, discard it.
       */
      if(ih->daddr() == index || suppress_reply) {
            Packet::free(p);
            return;
      }
      /*
       * Otherwise, forward the Route Reply.
       */
      // Find the rt entry
      idsaodv_rt_entry *rt0 = rtable.rt_lookup(ih->daddr());
      // If the rt is up, forward
      if(rt0 && (rt0->rt_hops != INFINITY2)) {
```

```
            assert (rt0->rt_flags == RTF_UP);
            rp->rp_hop_count += 1;
            rp->rp_src = index;
            forward(rt0, p, NO_DELAY);
            // Insert the nexthop towards the RREQ source to
            // the precursor list of the RREQ destination
            rt->pc_insert(rt0->rt_nexthop); // nexthop to RREQ source
        } else {
            // I don't know how to forward .. drop the reply.
#ifdef DEBUG
            fprintf(stderr, "%s: dropping Route Reply\n", __FUNCTION__);
#endif // DEBUG
            drop(p, DROP_RTR_NO_ROUTE);
        }
}
```

Declaration

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

10/11/2011

SOLOMON GEBREMESKEL ADANE

This project has been submitted for examination with my approval as an advisor.

DEJENE EJIGU (PhD)

Addis Ababa, Ethiopia

November, 2011