SIMULATION OF BLACK HOLE ATTACK IN WIRELESS AD-HOC NETWORKS

A MASTER'S THESIS

in

Computer Engineering

Atılım University

by

SEMİH DOKURER

SEPTEMBER 2006

SIMULATION OF BLACK HOLE ATTACK IN WIRELESS AD-HOC NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY
BY

SEMİH DOKURER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COMPUTER ENGINEERING
SEPTEMBER 2006

Approval of the Graduate School of Natural and Applied Science

Prof. Dr. Selçuk SOYUPAK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İbrahim AKMAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____	_____
Asst. Prof. Dr. Murat ERTEN	Asst. Prof. Dr. Çiğdem TURHAN
Co-Supervisor	Supervisor

Examining Committee Members :

Asst. Prof. Dr. Çiğdem TURHAN	_____
Asst. Prof. Dr. Murat ERTEN	_____
Asst. Prof. Dr. Hakan TORA	_____
Dr. Can Erkin ACAR	_____
Instructor Kasım ÖZTOPRAK	_____

ABSTRACT

SIMULATION OF BLACK HOLE ATTACK IN WIRELESS AD-HOC NETWORKS

Dokurer, Semih

M.S., Computer Engineering Department

Supervisor : Asst. Prof. Dr. Çiğdem TURHAN

Co-Supervisor : Prof.Dr. Murat ERTEN

September 2006, 66 pages

A Wireless ad-hoc network is a temporary network set up by wireless mobile computers (or nodes) moving arbitrary in the places that have no network infrastructure. Since the nodes communicate with each other, they cooperate by forwarding data packets to other nodes in the network. Thus the nodes find a path to the destination node using routing protocols. However, due to security vulnerabilities of the routing protocols, wireless ad-hoc networks are unprotected to attacks of the malicious nodes.

One of these attacks is the Black Hole Attack against network integrity absorbing all data packets in the network. Since the data packets do not reach the destination node on account of this attack, data loss will occur.

There are lots of detection and defense mechanisms to eliminate the intruder that carry out the black hole attack. In this thesis, we simulated the black hole attack in various wireless ad-hoc network scenarios and have tried to find a response system in simulations.

Keywords: Wireless Ad-hoc Network, Black Hole Attack, Simulation, Security, Intrusion Detection Systems.

ÖZ

KABLOSUZ ANLIK AĞLARDA
KARA DELİK SALDIRISI SİMÜLASYONU

Dokurer, Semih

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Çiğdem TURHAN

Ortak Tez Yöneticisi: Yrd. Doç. Dr. Murat ERTEN

Eylül 2006, 66 sayfa

Kablosuz Anlık Ağ; herhangi bir ağ alt yapısının olmadığı yerlerde, sürekli ve düzensiz bir şekilde hareket eden bilgisayarların (veya terminallerin) kendi aralarında oluşturdukları geçici bir ağdır. Terminallerin birbirleri ile iletişim halinde olabilmeleri için veri paketlerini ağdaki diğer bir terminale yönlendirerek iş birliği yapmaları gerekir. Bu sebeple terminaller yönlendirme protokollerini kullanarak hedef terminale bir yol bulurlar. Ancak kullanılan yönlendirme protokollerindeki güvenlik zafiyetlerinden dolayı kablosuz anlık ağlar kötü niyetli terminallerin ataklarına açıktır.

Bu saldırılardan bir tanesinde ağdaki bütün veri paketlerini içine çekerek ağ bütünlüğüne karşı yapılan Kara Delik Saldırısıdır. Bu saldırı sonucu veri paketleri hedef terminale ulaşamayacağı için ağda veri kaybı oluşacaktır. Kara Delik Saldırısını gerçekleştiren saldırganı saf dışı bırakmak için bir çok tespit ve savunma yöntemleri vardır.

Kara Delik Saldırısını gerçekleştiren saldırganın etkilerini yok etmek için bir çok tespit ve savunma mekanizmaları bulunmaktadır. Bu tezde, Kara Delik Saldırısı çeşitli kablosuz ağ senaryolarında simüle edilip simülasyonda bir savunma sistemi bulunmaya çalışılmıştır.

Anahtar Kelimeler: Kablosuz Anlık Ağ, Kara Delik Saldırısı, Simülasyon, Güvenlik, Saldırı Tespit Sistemleri

ACKNOWLEDGEMENT

I would like to express my gratitude to my advisors Asst. Prof. Dr. Çiğdem TURHAN and Asst. Prof. Dr. Murat ERTEN for their invaluable support and guidance.

I offer sincere appreciation to Dr. Can Erkin ACAR and instructor Kasım ÖZTOPRAK for their help and patience.

I would also like thank to my friends Erdal BERKTAŞ and Evren BİLGİÇ for their guidance, and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PDA : Personal Digital Assistant
WLAN : Wireless Local Area Network
ISM : Industry, Scientific, Medical
PAN : Personal Area Networks
WPAN : Wireless Personal Area Networks
LAN : Local Area Networks
WAN : Wide Area Networks
WWAN : Wireless Wide Area Networks
GSM : Global System for Mobile Communications
IEEE : Institute of Electrical and Electronics Engineers
PHY : Physical
CSMA/CD : Carrier Sense Multiple Access Network with Collusion Detection
ISO : International Organization for Standardization
Wi-Fi : Wireless Fidelity
WECA : Wireless Ethernet Compatibility Alliance
SII : Standard Indicator Icon
STA : Station
AP : Access Point
WDS : Wireless Distribution System
BSS : Basic Service Set
IBSS : Independent Basic Service Set
EBSS : Extended Basic Service Set
MANET : Mobile Ad-Hoc Network
DSDV : Destination-Sequenced Distance Vector Routing Protocol
WRP : The Wireless Routing Protocol
GSR : Global State Routing
FSR : Fisheye State Routing
HSR : Hierarchical State Routing
ZHLS : Zone-based Hierarchical Link State Routing Protocol
CGSR : Clusterhead Gateway Switch Routing Protocol

CBRP : Cluster Based Routing Protocols

AODV : Ad-Hoc On-demand Distance Vector Routing

DSRP : Dynamic Source Routing Protocol

TORA : Temporally Ordered Routing Algorithm

ABR : Associativity Based Routing

SSR : Signal Stability Routing

RREQ : Route Request

RREP : Route Replay

RERR : Route Error

DoS : Denial of Services

RFC : Request for Comment

NS : Network Simulator

NAM : Network Animator

MAC : Media Access Control

IP : Internet Protocol

ARP : Address Resolution Protocol

UDP : User Datagram Protocol

TCP : Transmission Control Protocol

ACK : Acknowledgement

CBR : Constant Bit Rate

TCL : Tool Command Language

OTCL : Object Oriented Tool Command Language

AGT : Agent

RTR : Router

IDS : Intrusion Detection System

1. INTRODUCTION

Wireless ad-hoc networks are composed of autonomous nodes that are self- managed without any infrastructure. In this way, ad-hoc networks have a dynamic topology such that nodes can easily join or leave the network at any time. They have many potential applications, especially, in military and rescue areas such as connecting soldiers on the battlefield or establishing a new network in place of a network which collapsed after a disaster like an earthquake. Ad-hoc networks are suitable for areas where it is not possible to set up a fixed infrastructure. Since the nodes communicate with each other without an infrastructure, they provide the connectivity by forwarding packets over themselves. To support this connectivity, nodes use some routing protocols such as AODV (Ad-hoc On-Demand Distance Vector), DSR (Dynamic Source Routing) and DSDV (Destination-Sequenced Distance-Vector). Besides acting as a host, each node also acts as a router to discover a path and forward packets to the correct node in the network.

As wireless 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 in. In this way, 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. Malicious nodes do not use this process and 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.

In our study, we simulated the Black Hole attack in wireless ad-hoc networks and evaluated its damage in the network. We made our simulations using NS-2 (Network Simulator version 2) simulation program that consists of the collection of all network protocols to simulate many of the existing network topologies. Even though NS-2 contains wireless ad-hoc routing protocols, it does not have any modules to simulate malicious protocols. Thus, to simulate Black Hole attacks, we first added a new Black Hole protocol into the NS-2. We started our study by writing a new AODV protocol using C++, to simulate the Black Hole attack.

Having implemented a new routing protocol which simulates the black hole we performed tests on different topologies to compare the network performance with and without black holes in the network. As expected, the throughput in the network was deteriorated considerably in the presence of a black hole.

Afterwards, we proposed an IDS solution to eliminate the Black Hole effects in the AODV network. We implemented the solution into the NS-2. and evaluated the results as we did in Black Hole implementation. As a result, our solution is eliminated the Black Hole effect with %24,38 success.

The rest of the thesis is organized as follows: In chapter 2 we presented wireless ad-hoc networks and their security vulnerabilities, including black hole attacks. In Chapter 3 we described the AODV protocol and how Black Hole Attack causes the protocol to misbehave. Chapter 4 presents NS (Network Simulator) and our contribution to this software. Chapter 5 describes the results of the network behavior due to the black hole attacks and Chapter 6 explains our solution to minimize the Black Hole effect followed by the conclusion in Chapter 7.

2. WIRELESS NETWORKS

Wireless communication is used to transfer data among users without a wired infrastructure. Using electromagnetic waves, mobile users transmit and receive data over the air. Wireless communication spreads from home RF to satellites, from cellular phones to walkie-talkies. Its mobility, simplicity and cost saving installation advantages make the wireless communication more popular, especially in recent decades Increasing user mobility needs and developments in the use of laptop computers and PDA's is one of the main reasons of the popularity of wireless networks.

2.1. Convenience Offered by Wireless Networks

Mobility

This is one of the obvious advantages of the wireless networks. Mobile users can connect to the existing networks while roaming freely and enjoying independence.

Simplicity

We can translate simplicity into rapid development. It is easy to install a wireless infrastructure, compared to a wired network.

Flexibility

Wireless network coverage area can reach where wire cannot go. It is very useful for moving vehicles or for the places where running cable is not possible like historical buildings.

2.2. Types of Networks

According to coverage area, three type of wireless interconnection have been defined. Personal Area Networks (PANs), Local Area Networks (LANs) and Wide Area Networks (WANs).

2.2.1. Personal Area Networks (PAN)

PAN is a computer network used for communication among computer devices (including telephones, PDAs, etc.) close to one person. [1] Typical PAN networks are Bluetooth, Sensor networks and zigbees. The Standards Board of the IEEE approved the standard 802.15, as MAC and PHY Specifications for Wireless PANs (WPANs).

2.2.2. Local Area Networks (LAN)

In this type of network, devices are communicating with each other in a local coverage area that can be a building or a campus. Wireless LANs (WLANs) are alternatives of conventional wired LANs. In a wired network nodes are communicating over physical environments such as cables. On the other hand, in a WLAN nodes use air as the medium. WLANs are standardized by Institute of Electrical and Electronics Engineers (IEEE).

2.2.3. Wide Area Networks (WAN)

WANs spread a relatively larger geographical area. Typically a WAN includes more than one LANs. 2G and 3G Mobile Cellular Networks, Satellite Systems and Paging Networks are examples of Wireless WANs (WWANs)

Figure 1 shows the ways in which different types of wireless networks and hardware may be used together to provide the best performance and mobility. [2]

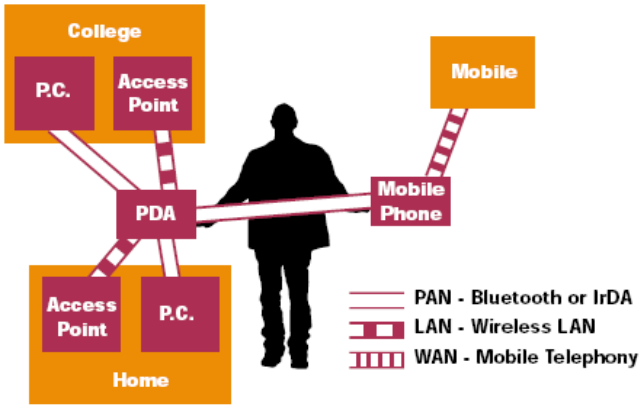

Figure 1 - Wireless uses in differing environments

Figure 1 shows that anybody who uses a PDA (equally a laptop) can access to the PCs in a wireless infrastructure using Bluetooth or WLAN technology while connecting with mobile phone over GSM. Actually Figure 1 indicates how wireless networks support mobility, simplicity and flexibility.

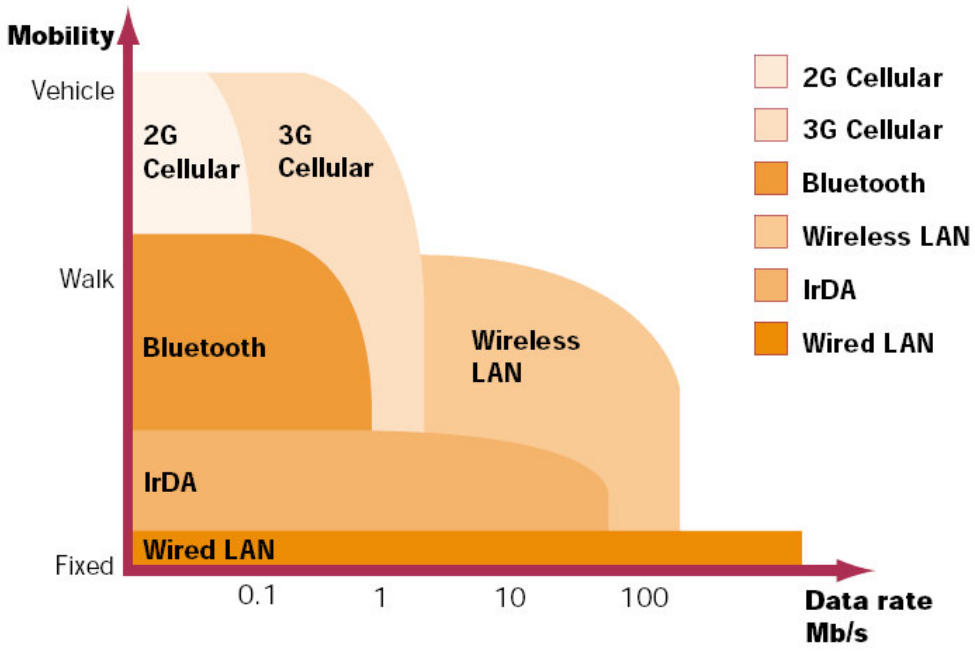

Figure 2 - Data rates and mobility for communication types

Figure 2 shows various types of wireless communication and their data rates and mobility. From this it can be seen that there is a balance to be struck between performance and mobility. [2]

2.3. Wireless Local Area Networks (WLAN)

WLANs are alternative of conventional LANs that connect nodes in wired environments. WLANs transmit information over wireless medium instead of wire. A Wireless Local Area Networks (WLAN) is a shared medium communication network that broadcast information over wireless links to be received by all stations (e.g. computing devices). [3]

WLANs are used mainly to connect to the Internet. Wireless internet access points are known as "*hot spots*" and are already available in coffeehouse and other public places such as airports, stations and hotels.

Thanks to these benefits, WLANs have gained significant popularity among mobile users to access real-time information. Actually WLANs are implemented in mobile devices such as laptops, PDAs etc. to communicate with each other without using wired Ethernet (IEEE 802.3). In a WLAN, instead of wired Ethernet protocol, IEEE 802.3, wireless Ethernet protocol, IEEE 802.11 is used.

2.4. IEEE 802.11 Standards, Specifications And Technologies

IEEE 802.11 is a member of the IEEE 802 protocol family, which defines specifications of Local Area Network (LAN) technologies. IEEE 802 specifications are focused on two lowest layers of the OSI model, the MAC and the physical (PHY) component that incorporate each other. In the IEEE 802 series, individual specifications are determined after the point. 802.3, for example design Carrier Sense Multiple Access network with Collision Detection (CSMA/CD) and 802.5 is the Token-Ring specification. Figure 3 shows the various components of the 802 family and their relation with the ISO models.

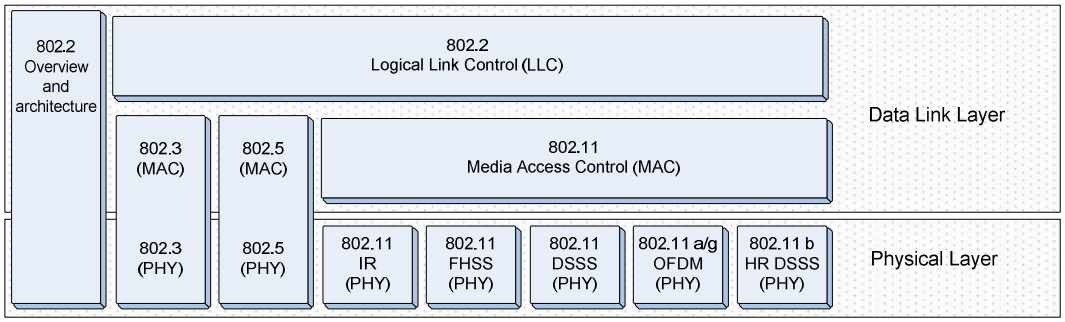

Figure 3 – IEEE 802 family and relation with the ISO models

In Figure 3, there are four modulation techniques in the physical layer and four specifications in 802.11 family. Table 1 compares the 802.11 family standards.

IEEE Standards	Speed	Frequency	Interface
802.11	Up to 2 Mbps	2.4 GHz	IR / FHSS / DSSS
802.11a	Up to 54 Mbps	5 GHz	OFDM
802.11b	Up to 11 Mbps	2.4 GHz	HR-DSSS
802.11g	Up to 54 Mbps	2.4 GHz	OFDM

Table 1 - Comparison of 802.11 standards

IEEE 802.11 standards / specifications / technologies referred to as Wi-Fi (Wireless Fidelity) that is also a trademark of the Wi-Fi Alliance, a nonprofit organization originally formed as WECA (Wireless Ethernet Compatibility Alliance). [3]

In 1999, WECA has published its Wi-Fi certification [4] program. Any 802.11 vendor can have their products tested for interoperability using Wi-Fi certification. Tested products are awarded a Wi-Fi Certified logo with colored Standard Indicator Icon (SII). Figure 4 shows examples of the Wi-Fi certified logo with SII.

Figure 4 - Wi-Fi certified logos with SII

Glossary of 802.11 Wireless Term

Station (STA) : Station is defined as an 802.11 compliant device that could be a computer with wireless network ethernet card.

Access Point (AP) : AP is a bridging function device that performs data transfer between station and/or wired network.

Wireless Distribution System (WDS) : WDS is the backbone system used to relay frames between access points.

Basic Service Set (BSS) : BSS is the basic building block of an 802.11 network, which is simply a group of stations that communicate with each other.

Basic Service Area (BSA) : BSA is a fuzzy area that is defined by the propagation characteristic of wireless medium.

2.5. WLAN Modes

If minimum two stations in a BSA communicate with each other, they are members of the BSS. The 802.11 standard has two BSS modes. These are ad-hoc and infrastructure networks. These two networks are illustrated in Figure 5 and Figure 6.

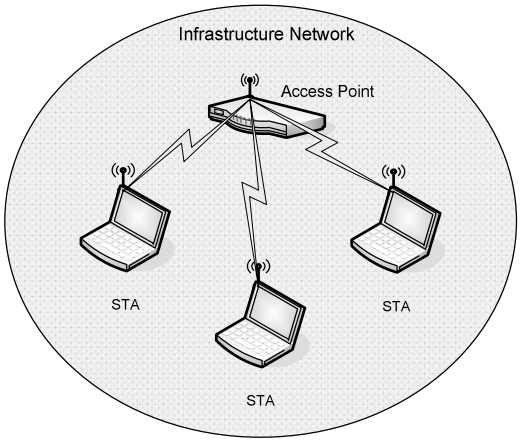

Figure 5 - Infrastructure Network

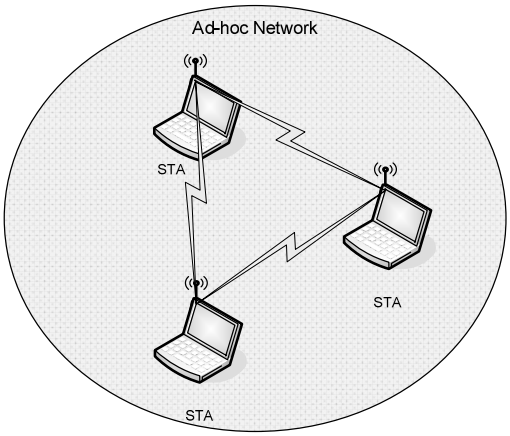

Figure 6 - Ad-hoc Network

2.5.1. Infrastructure Network:

This network is called as *Infrastructure Basic Service Set* (BSS never called IBSS). Stations in a same BSA communicate with each other over access points. Thus, a station communicates with another at two hops. First, frames are sent to the access point, then access point forwards them to the destination station. To be a member of a BSS, stations must *associate* themselves to the access point. Association function of the infrastructure networks is similar to plugging cable to the wired networks.

BSSs generally cover a small area, such as offices and home. 802.11 allows the stations to be mobile in a larger coverage area than BSSs. More than two access points can create *Extended Service Set* (ESS) by chaining each other with a backbone network. Figure 7 illustrates an ESS by which is made up of three BSSs.

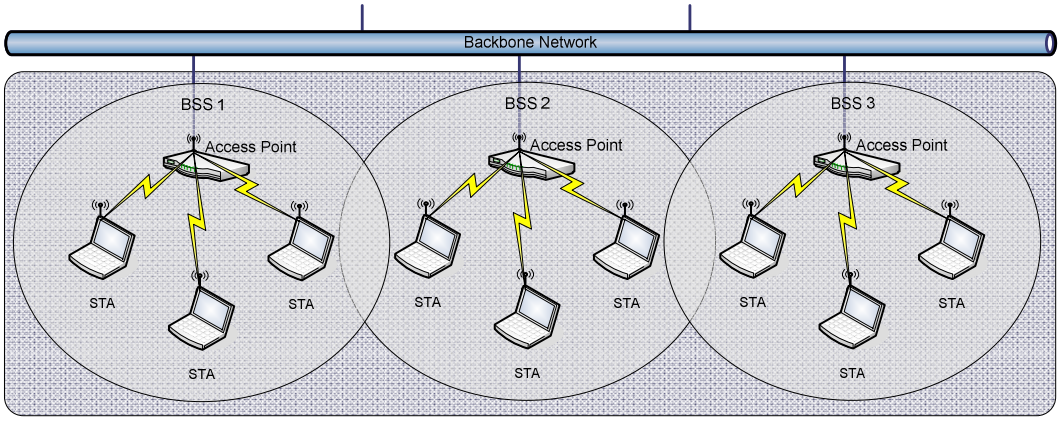

Figure 7 - Illustration of ESS

One of the benefits of the 802.11 standards is the mobility. Within the same ESS, stations can roam freely regardless of which BSA they are in. Wireless medium acts like a single layer 2 connection. Access point follows stations using association disassociation function of the BSS. While a station is roaming in an ESS, access points can follow where it is.

An access point in a BSS or an ESS must know which station is associated to itself. Station may want to transmit information to another station that exists in the same or different BSS. If access points take the address of the stations, they can bridge the

information to the right BSS. This function is supported with Wireless Distribution System of 802.11. Another function of the WDS is to connect two link layers, 802.11 and 802.3. WDS consist of bridging function of access point and ethernet backbone. Figure 8 shows the composition of these two functions in a WDS.

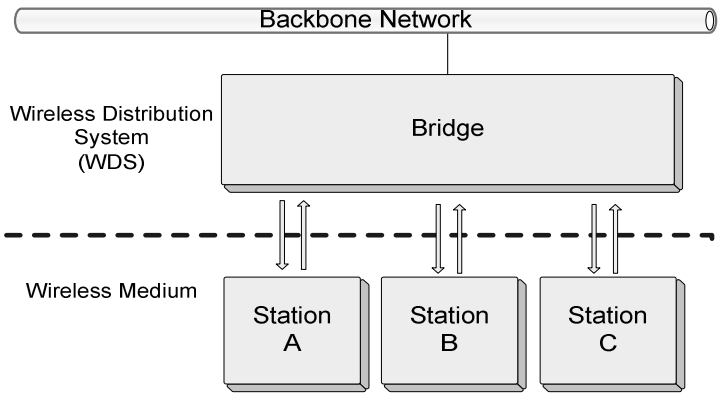

Figure 8 - Wireless Distribution System

2.5.2. Ad-Hoc Network :

This network is called *Independent Basic Service Set (IBSS)* Stations in a IBSS communicate directly with each other and do not use an access point. Because of the mobility associated with ad-hoc networks, they are commonly called MANET (Mobile Ad-hoc NETwork). MANETs are self organized networks whose nodes are free to move randomly while being able to communicate with each other without the help of an existing network infrastructure. MANETs are suitable for use in situations where any wired or wireless infrastructure is inaccessible, overloaded, damaged or destroyed such as emergency or rescue missions, disaster relief efforts and tactical battlefields, as well as civilian MANET situations, such as conferences and classrooms or in the research area like sensor networks. MANETs eliminate this dependence on a fixed network infrastructure where each station acts as an intermediate switch.

2.5.3. Comparison of Infrastructure and Ad-hoc Networks

In an infrastructure network, stations are required to be in the coverage area of access point. Therefore, mobility is limited with the distance between the access point and the station. But, in an Ad-hoc network, a station can transmit data to another one as long as there is a third station that can cover both of them. Data is forwarded via intermediate station/s using one of the ad-hoc network routing protocols. This approach supports a larger working area but physical layer complexity increases. Stations may search for the target station that is out of range by flooding the network with broadcasts that are forwarded by each station.

In an infrastructure network, access points can handle battery optimization for its stations. While a station is in the power saving mode, access point can buffer frames for it. But in an ad-hoc network, power consumption is higher, since stations transmit frames that do not concern themselves.

When transmitting packets over mobile nodes, hop count is changing in MANETs although in infrastructure WLANs, communication must have two hops. It is possible to include rapidly changing, random, multihop topologies in the routing function of MANET.

2.6. Routing In MANETs

MANETs have special limitation and properties such as limited bandwidth and power, highly dynamic topology, high error rates etc., explained in the preceding sections. Moreover, compared to infrastructure based networks, in a MANET, all nodes are mobile and can be connected dynamically in an arbitrary manner. Nodes of MANET behave as router and take part in discovery and maintenance to establish a reliable route of each other. Therefore, routing protocols for wired networks cannot be directly used in wireless networks and numerous protocols have been developed for MANETs. These routing protocols are divided into two categories based on management of routing tables. These categories are Table Driven Routing Protocols and On-Demand Routing Protocols, shown in the Table 2 and they are explained below:

MANET ROUTING PROTOCOLS	
Table Driven Routing Protocols	**On-Demand Routing Protocols**
Destination-Sequenced Distance Vector Routing Protocol (DSDV)	Ad-Hoc On-Demand Distance Vector Routing (AODV)
Wireless Routing Protocol (WRP)	Cluster based Routing Protocols (CBRP)
Global State Routing (GSR)	Dynamic Source Routing Protocol (DSRP)
Fisheye State Routing (FSR)	Temporally Ordered Routing Algorithm (TORA)
Hierarchical State Routing (HSR)	Associativity Based Routing (ABR)
Zone-based Hierarchical Link State Routing Protocol (ZHLS)	Signal Stability Routing (SSR)
Clusterhead Gateway Switch Routing Protocol (CGSR)	

Table 2 – Classification of MANET routing protocols

2.6.1. Table Driven Routing Protocols

In Table Driven Routing Protocols, each node has to keep up-to-date routing tables. To maintain reliable routing tables, every node propagates the update messages to the network when the network topology changes. Because every node has information about network topology, Table Driven Routing Protocols present several problems.

- Periodically updating the network topology increases bandwidth overhead,
- Periodically updating route tables keeps the nodes awake and quickly exhaust their batteries,
- Many redundant route entries to the specific destination needlessly take place in the routing tables.

Destination-Sequenced Distance Vector Routing Protocol (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR), Fisheye State Routing (FSR), Hierarchical State Routing (HSR), Zone-based Hierarchical Link State Routing Protocol (ZHLS) and Clusterhead Gateway Switch Routing Protocol (CGSR) are Table Driven Routing Protocols.

2.6.2. On-Demand Routing Protocols

These protocols take a lazy approach to routing. [5] Compared to Table Driven Routing Protocols; On-Demand Routing Protocols are not maintained periodically, route tables are created when required. When the source node wants to connect to the destination node, it propagates the route request packet to its neighbors. Just as

neighbors of the source node receive the broadcasted request packet, they forward the packet to their neighbors and this action is happen until the destination is found. Afterward, the destination node sends a replay packet 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.

Cluster based Routing Protocols (CBRP), Ad-Hoc On-Demand Distance Vector Routing (AODV), Dynamic Source Routing Protocol (DSRP), Temporally Ordered Routing Algorithm (TORA), Associativity Based Routing (ABR), Signal Stability Routing (SSR) are On-Demand Routing protocols.

In our work, we have used Ad-Hoc On-Demand Distance Vector Routing (AODV) and implemented Black Hole attack to this protocol. AODV protocol and Black Hole Attack are detailed in next chapter.

2.7. Security Issues for MANETs

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. General attack types are the threats against the routing layer of the ad-hoc networks; such as physical, MAC and network layer which is the most important function of wireless ad-hoc network for the routing mechanism, orienting the packets after a route discovery process. Other vulnerabilities are application security, network security, database security which are studied in different works which are not explained in detail here.

Attacks to the wireless ad-hoc network in the networking layer usually have two purposes: not forwarding packets or adding and changing some parameters of routing messages; such as sequence number and IP addresses. These will be detailed in the subsequent sections.

Using one of the key mechanisms such as cryptography or authentication, or both in a network, serves as a preventive approach and can be employed against '*attackers*'. However, these mechanisms protect the network against attacks that come from

outside, malicious '*insiders*' which use one of the critical keys can also threaten the security. For instance, in a battle field where ad-hoc networks are used, even if keys are protected by temper proof hardware that are used in the vehicles in the network, it is difficult to say that these vehicles exhibit the same behavior if the enemy captures them.

On the other hand, a node may undeliberately misbehave as if it is damaged. A node with a failed battery which is unable to perform network operations may be perceived as an attack. Another malicious behavior of the nodes is selfishness. Selfish nodes refrain from consuming its resources; such as battery, by not participating innetwork operations. Therefore; failed and selfish nodes also affect the network performance as they do not correctly process network packets, such as in routing mechanism.

We should, therefore ensure that everything is correctly working in the network to support overall security and know how an insider is able to attack the wireless ad-hoc network. Wireless ad-hoc networks should be protected with an intrusion detection system that can understand the possible actions of attackers and can produce a solution against these attacks.

2.7.1. Attack Types

2.7.1.1. Passive Eavesdropping

An attacker can listen to any wireless network to know what is going on in the network. It first listens to control messages to infer the network topology to understand how nodes are located or are communicating with another. Therefore, it can gather intelligent information about the network before attacking. It may also listen to the information that is transmitted using encryption although it should be confidential belonging to upper layer applications.

Eavesdropping is also a threat to location privacy [6]. An unauthorized node can notice a wireless network that exists within a geographical area, just by detecting radio signals. To combat this, traffic engineering techniques have been developed.

2.7.1.2. Selective Existence (Selfish Nodes)

This malicious node which is also known as *selfish node* and which is not participating in the network operations, use 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.* [7]. 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. 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. 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. Thus selectively dropping messages is not a selfish node behavior mentioned in [8]. 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.

2.7.1.3. Gray Hole Attack (Routing Misbehavior)

Gray hole attacks is an active attack type, which lead to dropping of messages. Attacking node first agrees to forward packets and then fails to do so. Initially the node behaves correctly and replays true RREP messages to nodes that initiate RREQ message. This way, it takes over the sending packets. Afterwards, the node just drops the packets to launch a (DoS) denial of service attack.

If neighboring nodes that try to send packets over attacking nodes lose the connection to destination then they may want to discover a route again, broadcasting RREQ messages. Attacking node establishes a route, sending RREP messages. This process goes on until malicious node succeeds its aim (e.g. network resource consumption, battery consumption). This attack is known as routing misbehavior. [9]

Dropping packets is also one of the behaviors of failed or overloading nodes [6]. One should not evaluate every dropping packet action as a selective existence, gray or black hole attack. Actually most routing protocols have no mechanism to detect whether data packets have been forwarded, DSR being the only exception. [10]

2.7.1.4. Black Hole Attack

The difference of Black Hole Attacks [11] compared to Gray Hole Attacks is that malicious nodes never send true control messages initially. To carry out a black hole attack, malicious node waits for neighboring nodes to send RREQ messages. When the malicious node receives an RREQ message, without checking its routing table, immediately sends a false RREP message giving a route to destination over itself, assigning a high sequence number to settle in the routing table of the victim node, before other nodes send a true one. Therefore requesting nodes assume that route discovery process is completed and ignore other RREP messages and begin to send packets over malicious node.

Malicious node attacks all RREQ messages this way and takes over all routes. Therefore all packets are sent to a point when they are not forwarding anywhere. This is called a black hole akin to real meaning which swallows all objects and matter. To succeed a black hole attack, malicious node should be positioned at the center of the wireless network.

If malicious node masquerades false RREP message as if it comes from another victim node instead of itself, all messages will be forwarded to the victim node. By doing this, victim node will have to process all incoming messages and is subjected to a sleep deprivation attack.

Gray hole attacks against one or two nodes in the network to isolate them, where as black hole attack affects the whole network. Moreover, the malicious node that attempts gray hole attacks cannot be perceived easily since it does not send false messages. Behavior of failed or overloaded nodes may seem like selfish nodes attacks or gray hole attacks due to dropping of messages. But, since failed nodes cannot fabricate a new control message, they cannot form a black hole attack although they will drop the message later.

This attack type and how the AODV Routing Protocol is misused will be illustrated in Chapter 3.

2.7.1.5. Impersonation

Due to lack of authentication in ad-hoc networks, only MAC or IP addresses uniquely identify hosts. These addresses are not adequate to authenticate the sender node. Therefore non-repudiation is not provided for ad-hoc network protocols. MAC and IP spoofing are the simplest methods to pretend as another node or hide in the network.

Malicious nodes achieve impersonation only by changing the source IP address in the control message. Another reason for impersonation is to persuade nodes to change their routing tables pretending to be a friendly node, such as attacks against routing table.

One of the interesting impersonations is Man-in-the-middle attack [7]. Malicious node performs this attack by combining spoofing and dropping attacks. Physically, it must be placed as the only node within the range for destination, in the middle of the route or victim node must be prevented from receiving any other route information to the destination. Malicious node may also change the routing tables of the victim node to redirect its packets, using attacks against the routing table. At this point, malicious node waits for an RREQ message to the destination node from source node. When source node sends an RREQ message, malicious node drops the RREQ and replays a spoofed RREP message to source node as if it is coming from the destination node. At the same time, malicious node sends a RREQ message to the destination node and

drops the RREP message from the destination node. By doing this; malicious node manages to establish a route both to the source and the destination node and attacker controls the communication between the source and destination. If the communication is encrypted or entails an authentication as to MAC or IP address, malicious node can easily get the up layer communication.

2.7.1.6. Modification Attack

Control massages are used to establish the shortest and true path between two nodes. But malicious nodes want to route packets to the direction that they want, modifying content of the control messages (e.g. RREQ, RREP and RERR). Modification means that the message does not carry out its normal functions.

Route information such as hop count, sequence number, life time etc. are carried along with control messages. This information has a big role in establishing a true route. Modifying these fields in the control messages, malicious node can perform its own attacks. Impersonation is not one of these kinds of attacks; impersonation is only performed by modifying source address to pretend as another node in the network. But changing route information in control messages is performed to mislead the victim or intermediate node and this modification is generally against the replay messages.

For example; by changing hop count or sequence number in the RREP messages, malicious node wants to change route information of victim node. In this attack type; malicious node decreases its sequence number in the RREP message, first capturing it, and finally sending it to the claimed node. When victim node receives this false message it chooses the costly route in the network. Malicious node intends to perform this attack to affect the network performance, or its intension may be selfish, it does not want to route the packet. This attack can be performed by adding a number of virtual nodes and decreasing hop count field of the RREP messages. This attack is also known as detour attack. [7]

Another attack is performed by changing destination IP field in any control message. Thus, messages are not forwarded to relevant node and the communication is broken.

At the same time the malicious node may send all messages to the victim node to perform denial of service (DoS) attack or to another malicious node to collect the aggregated network dump. To perform the latter one; more than one malicious node should be located in the network and one of them should be located in the middle of the network to collect messages. This way; collaborative malicious nodes can obtain all information about the network.

2.7.1.7. Attack Against The Routing Tables

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 a period (max. 3 seconds, duration of ACTIVE_ROUTE_TIMEOUT constant value of AODV protocol). If malicious node attacks against this table, attacked nodes do not find any route to other nodes whom it wants to connect. This attack is always performed by fabricating a new control message. Therefore it is also named fabricating attack.

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 RERR messages to the network so that valid working links are marked as broken [6]. Another attack type against the routing table is to attempt to create lots of route entries for non-existent nodes, using RREQ messages. As a result, routing table of the attacked node is full and does not have enough entry to create a new one. This attack type is known as routing table overflow. [11]

Attacks against the routing tables also affect the network integrity, changing the network topology established in the routing tables. Incorrect control messages are disseminated quickly in the network due to route discovery process and influence the network integrity in a wide area. Therefore attacks against the routing table are known as Network Integrity Attacks. [6]

2.7.1.8. Sleep Deprivation Torture Attack (Battery Exhaustion)

Many techniques are used to maximize the battery life and mobile nodes prefer to stay at the sleep mode, when they are not used. Sleep Deprivation Torture [12] is one of the serious types of Denial of Service Attacks, which affects only nodes, especially handheld devices that have limited resources.

In a period time, attacker can propagate some control messages through the network, in which other nodes are interested. Other nodes pass to the operation mode from the sleep mode and start processing these unnecessary packets until their batteries completely run out.

3. BLACK HOLE ATTACK IN AODV ROUTING PROTOCOL

Initially, we should take into account Ad-hoc On-Demand Distance Vector (AODV) Routing Protocol and then we shall explain Black Hole Attack.

3.1. Ad-hoc On-Demand Distance Vector (AODV) Routing Protocol

Ad-hoc On-Demand Distance Vector (AODV) [13] Routing Protocol is used for finding a path to the destination in an ad-hoc network. To find the path to the destination all mobile nodes work in cooperation using the routing control messages. Thanks to these control messages, AODV Routing Protocol offers quick adaptation to dynamic network conditions, low processing and memory overhead, 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.

Route Requests (RREQs), Route Replay (RREPs), Route Errors (RERRs) are control messages used for establishing a path to the destination, sent using UDP/IP protocols. Header information of these control messages are explained in [13]. When the source node wants to make a connection with the destination node, it broadcasts an RREQ message. This RREQ message is propagated from the source, received by neighbors (intermediate nodes) of the source node. The intermediate nodes broadcast the RREQ message to their neighbors. This process goes on until the packet is received by destination node or an intermediate node that has a fresh enough route entry for the destination. Figure 9 shows how the RREQ message is propagated in an ad-hoc network.

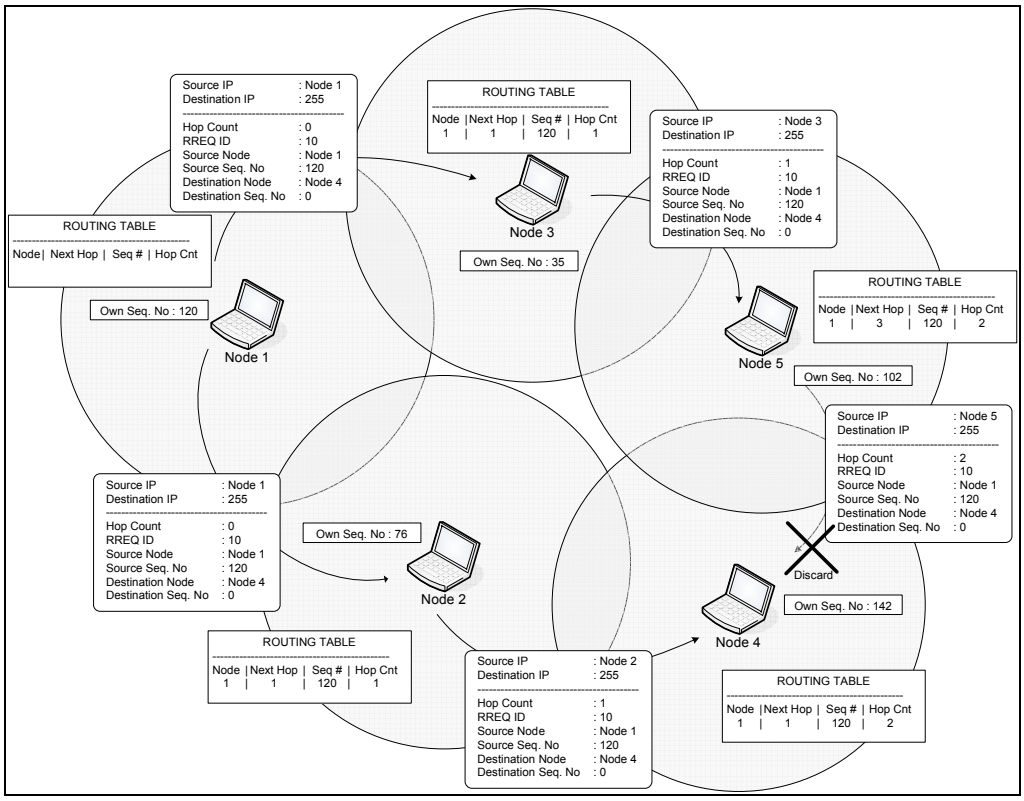

Figure 9 – Propagation of the RREQ message

Fresh enough means that the intermediate node has a valid route to destination formed a period of time ago, lower than the threshold. While the RREQ packet travels 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 RREQs, controlling the ID field of the RREQ message. 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. Afterwards the RREP message is unicasted to the source node. The difference between the broadcasting an RREQ and unicasting RREP can be seen from Figures 9 and 10. While the RREQ and the RREP messages are forwarded by intermediate nodes, intermediate nodes update their routing tables and save this route entry for 3 seconds, which is the ACTIVE_ROUTE_TIMEOUT constant value of AODV protocol. The default constant values of the AODV protocol are listed in appendix of RFC – 3561 [13]. Thus the node knows over which neighbor to reach at the

destination. In terminology, the neighbor list for destination is labeled as “Precursor List”. Figure 10shows how the RREP message is unicasted and how the route entries in the intermediate nodes are updated.

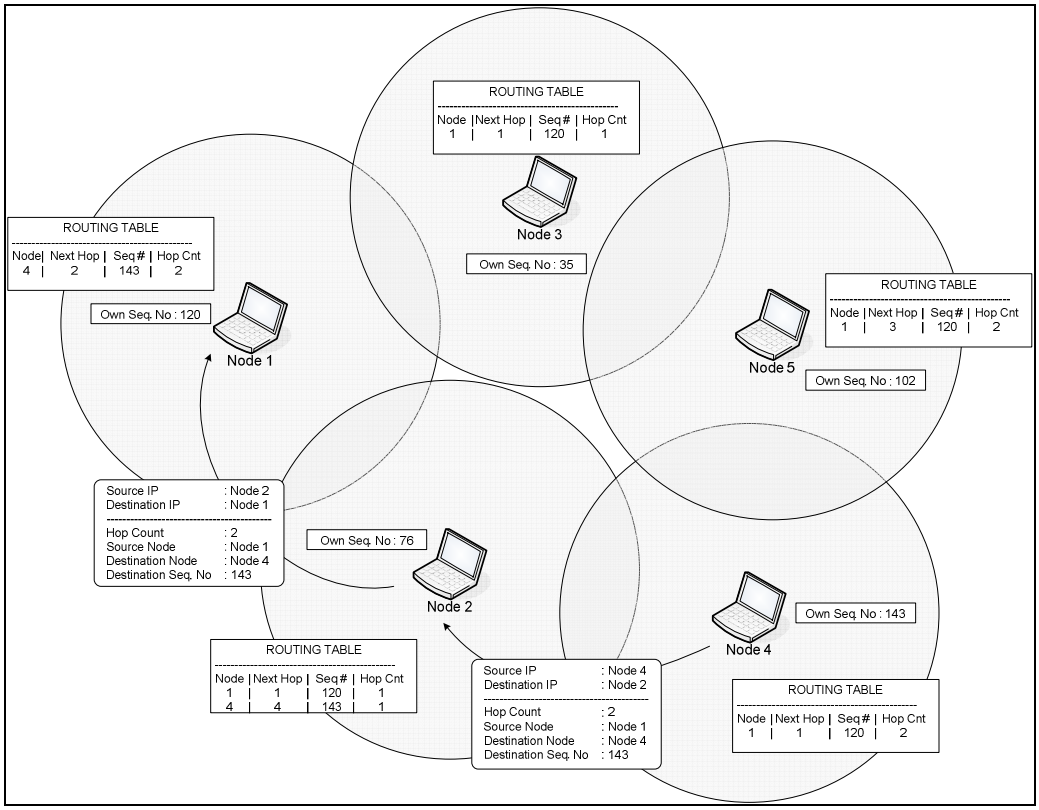

Figure 10 – Unicasting the RREP message

3.2. Sequence Numbers

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 control message, RREQ, RREP, RERR etc., it increases its own sequence number. 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 (i.e., 4294967295). If the sequence number of the node reaches the possible highest sequence number,

4294967295, then it will be reset to zero (0). If the results of subtraction of the currently stored sequence number in a node and the sequence number of incoming AODV route control message is less than zero, the stored sequence number is changed with the sequence number of the incoming control message.

In Figure 11, while Node 2 forwards the RREP message coming from Node 3, it compares its own previously stored sequence number with that of Node 3. If it notices that the sequence number is newer than its own, then it changes its route table entry as necessary.

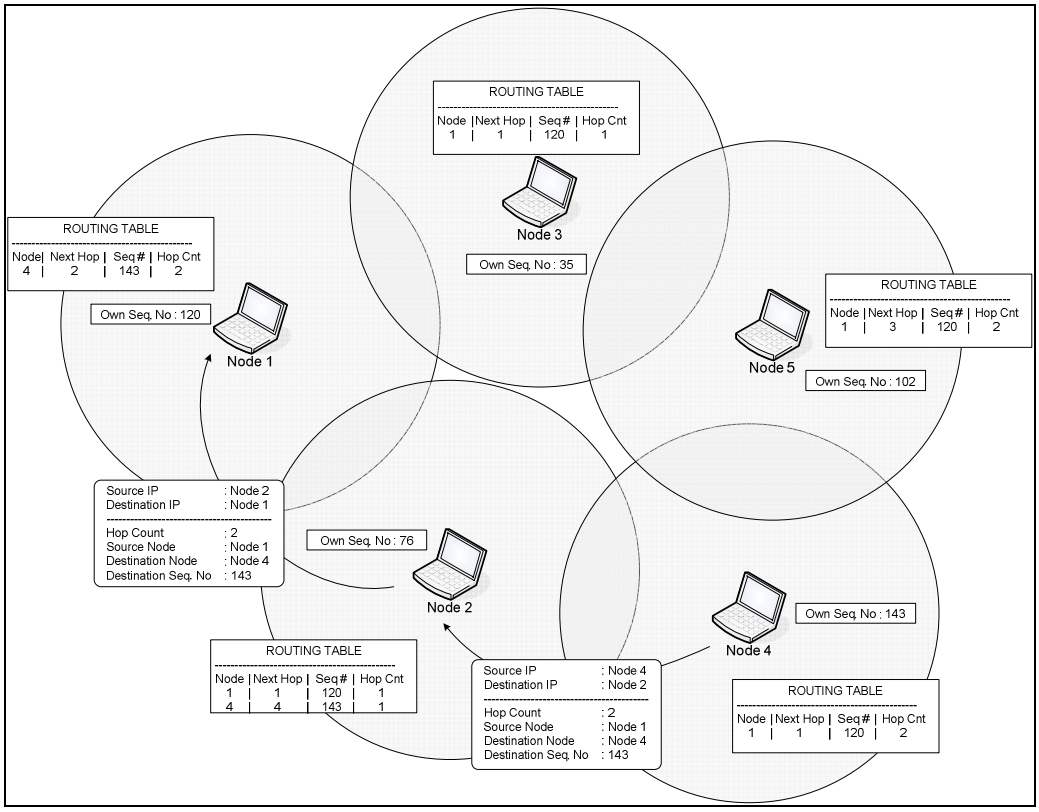

Figure 11 – Updating the Sequence Number with fresh one

3.3. Black Hole Attack

Black Hole Attack is briefly explained in the previous Chapter. In this Chapter we shall explain it in more detail as we have already explained the AODV protocol.

In an ad-hoc network that uses the AODV protocol, a Black Hole node absorbs the network traffic and drops all packets. To explain the Black Hole Attack we added a malicious node that exhibits Black Hole behavior in the scenario of the figures of the previous section.

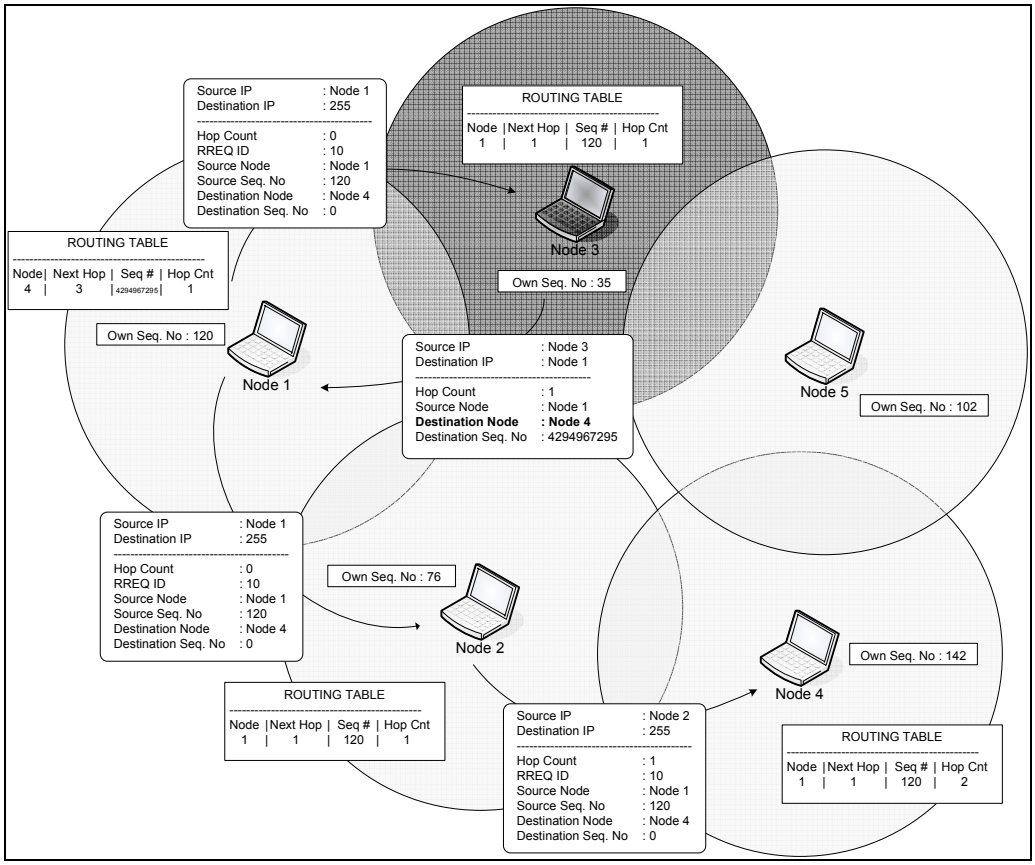

Figure 12 – Illustration of Black Hole Attack

In this scenario shown in Figure 12, we assume that Node 3 is the malicious node. When Node 1 broadcasts the RREQ message for Node 4, Node 3 immediately responds to Node 1 with an RREP message that includes the highest sequence number of Node 4, as if it is coming from Node 4. Node 1 assumes that Node 4 is behind Node 3 with 1 hop and discards the newly received RREP packet come from Node 2. Afterwards Node 1 starts to send out its data packet to the node 3 trusting that these packets will reach Node 4 but Node 3 will drop all data packets.

In a Black Hole Attack, after a while, the sending node understands that there is a link error because the receiving node does not send TCP ACK packets. If it sends out new TCP data packets and discovers a new route for the destination, the malicious node still manages to deceive the sending node. If the sending node sends out UDP data packets the problem is not detected because the UDP data connections do not wait for the ACK packets.

In our scenarios we use UDP data packets and we will explain our scenarios and their results in Chapter 5. Before Chapter 5 we will describe how Black Hole behavior is implemented in the simulator program, NS (Network Simulator).

4. NETWORK SIMULATOR (NS) AND OUR CONTRIBUTION

In this work, we have tried to evaluate the effects of the Black Hole attacks in the wireless Ad-hoc Networks. To achieve this we have simulated the wireless ad-hoc network scenarios which includes Black Hole node using NS Network Simulator [14] program. To simulate the Black Hole node in a wireless ad-hoc network we have implemented a new protocol that drops data packets after attracting them to itself. In this chapter we present NS and our contribution to this software.

4.1. NS Network Simulator

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. The NS is a part of software of the VINT project [15] that is supported by DARPA since 1995.

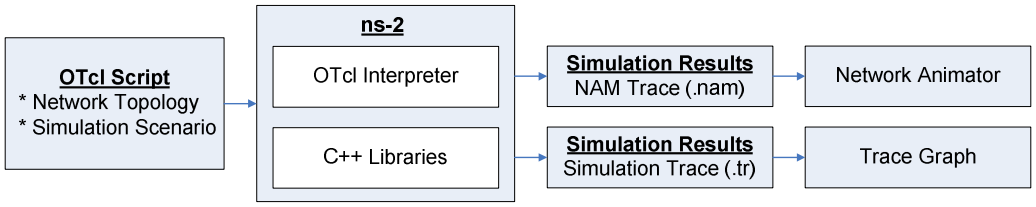

Figure 13 - NS-2 schema

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. In Chapter 5 the usage of the Tcl Language will be explained in detail.

As shown in Figure 13 [16], an OTcl script written by a user is interpreted by NS. While OTcl script is being interpreted, NS creates two main analysis reports simultaneously. One of them is NAM (Network Animator) object that shows the visual animation of the simulation. The other is the trace object that consists of the behavior of all objects in the simulation. Both of them are created as a file by NS. Former is .nam file used by NAM software that comes along with NS. Latter is a ".tr" file that includes all simulation traces in the text format.

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 study we have used version 2.29 of ns all-in-one package and installed the package in the Windows environment using Cygwin. After version 2, NS is commonly using a NS-2 and in our thesis we shell refer to it as NS-2. We have written the ".tcl" files in text editor and analyzed the results of the ".tr" file using "**cat**", "**awk**", "**wc**" and "**grep**" commands of Unix Operating System. The implementation phase of the Black hole behavior to the AODV protocol is written using C++.

4.2. Implementing A New Routing Protocol in NS To Simulate Black Hole Behavior

In [17] Implementation of a New Manet Unicast Routing Protocol in NS-2 is described. To implement our contribution we have used the details explained in this paper. In our work, we have used the nodes that exhibit black hole behavior in wireless ad-hoc network that use AODV protocol. Since the nodes behave as a Black Hole they have to use a new routing protocol that can participate in the AODV messaging. Implementation of this new routing protocol is explained below in detail:

All routing protocols in NS are installed in the directory of "**ns-2.29**". We start the work by duplicating AODV protocol in this directory and change the name of directory as "**blackholeaodv**".

Names of all files that are labeled as "aodv" in the directory are changed to "**blackholeaodv**" such as *blackholeaodv.cc*, *blackholeaodv.h*, *blackholeaodv.tcl*, *blackholeaodv_rqueue.cc*, *blackholeaodv_rqueue.h* etc. in this new directory except for "*aodv_packet.h*". The key point in our work is that AODV and Black Hole AODV protocol will send each other the same AODV packets. Therefore, we did not copy "*aodv_packet.h*" file into the blackholeaodv directory.

We have changed all classes, functions, structs, variables and constants names in all the files in the directory except struct names that belong to AODV packet.h code. We have designed aodv and blackholeoadv protocols to send each other aodv packets. These two protocols are actually the same.

After the above changes, we have changed two common files that are used in NS-2 globally to integrate new blackholeaodv protocol to the simulator. In [17] more files are changed to add new routing protocol and this new protocol uses its own packets. But in our implementation we do not need to add a new packet. Therefore we have changed only two files. The changes are explained below.

```
blackholeAODV {
set ragent [$self create-blackholeaodv-agent $node]
       }
Simulator instproc create-blackholeaodv-agent { node } {
       set ragent [new Agent/blackholeAODV [$node node-addr]]
       $self at 0.0 "$ragent start"            # start BEACON/HELLO Messages
       $node set ragent_ $ragent
       return $ragent
}
```

Figure 14 – "*blackholeaodv*" protocol agent is added in "*\tcl\lib\ ns-lib.tcl*"

The First file modified is "*\tcl\lib\ ns-lib.tcl*" where protocol agents are coded as a procedure. When the nodes use blackholeaodv protocol, this agent is scheduled at the beginning of the simulation and it is assigned to the nodes that will use

blachholeadodv protocol. The agent procedure for blackholeaodv is shown in Figure 14.

Second file which is adapted is "*makefile*" in the root directory of the "**ns-2.29**". After all implementations are ready, we have to compile NS-2 again to create object files. We have added the below lines in Figure 15 to the "*makefile*".

```
blackholeaodv/blackholeaodv_logs.o blackholeaodv/blackholeaodv.o \
blackholeaodv/blackholeaodv_rtable.o blackholeaodv/blackholeaodv_rqueue.o \
```

Figure 15 – Addition to the "*makefile*"

So far, we have implemented a new routing protocol which is labeled as blackholeaodv. But Black Hole behaviors have not yet been implemented in this new routing protocol. To add Black Hole behavior into the new AODV protocol we made same changes in blackholeaodv/blackholeaodv.cc C++ file. We will describe these changes we made in blackholeaodv/blackholeaodv.cc file explaining working mechanism of the AODV and Black Hole AODV protocols below.

When a packet is received by the "*recv*" function of the "*aodv/aodv.cc*", it processes the packets based on its type. If packet type is any of the many AODV route management packets, it sends the packet to the "*recvAODV*" function that we will explain below. If the received packet is a data packet, normally AODV protocol sends it to the destination address, but behaving as a Black Hole it drops all data packets as long as the packet does not come to itself. In the code below, the first "*if*" condition provides the node to receive data packets if it is the destination. The "*else*" condition drops all remaining packets. If statement is shown in Figure 16.

```
if ( (u_int32_t)ih->saddr() == index)
       forward((blackholeaodv_rt_entry*) 0, p, NO_DELAY);
else
       drop(p, DROP_RTR_ROUTE_LOOP);
```

Figure 16 – "If" statement for dropping or accepting the packets

If the packet is an AODV management packet, "*recv*" function sends it to "*recvblackholeAODV*" function. "*recvblackholeAODV*" function checks the type of the AODV management packet and based on the packet type it sends them to appropriate function with a "*case*" statement. For instance; RREQ packets are sent to the "*recvRequest*" function, RREP packets to "recvReply" function etc. case statements of "recvblackholeAODV" function is shown in Figure 17.

```
case AODVTYPE_RREQ:
   recvRequest(p);
   break;
case AODVTYPE_RREP:
   recvReply(p);
   break;
case AODVTYPE_RERR:
   recvError(p);
   break;
case AODVTYPE_HELLO:
   recvHello(p);
   break;

default:
   fprintf(stderr, "Invalid blackholeAODV type (%x)\n", ah>ah_type);
   exit(1);
```

Figure 17 – Case statement for choosing the AODV control message types

In our case we will consider the RREQ function because Black Hole behavior is carried out as the malicious node receives an RREQ packet. 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. 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 18.

```
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 18 – False RREP message of Black Hole Attack

After all changes are finished we have recompiled all NS-2 files to create object files. Having finished compilation, we have a new test bed to simulate Black Hole Attack in AODV protocol. In the next chapter we will describe the simulations and simulation results.

5. SIMULATION OF BLACK HOLE ATTACK AND ITS EFFECTS

In Chapter 3 we explained Black Hole Attack in AODV Routing Protocol and in Chapter 4 we described how this attack is implemented into the NS. In this Chapter, first, we will briefly explain the Tcl Language to understand the simulation scenarios. Having shown how we tested the Black Hole implementation, we will present the simulations of Black Hole Attack to demonstrate its effects. Then we will evaluate the effects of Black Hole Attack in an Ad-Hoc Networks.

5.1. Tcl Language in NS

Short for Tool Command Language, TCL is a powerful interpreted programming language developed by John Ousterhout at the University of California, Berkeley. [18] TCL is a very powerful and dynamic programming language. It has a wide range of usage, including web and desktop applications, networking, administration, testing etc. Tcl is a truly cross platform, easily deployed and highly extensible.

The most significant advantage of Tcl language is that it is fully compatible with the C programming language and Tcl libraries can be interoperated directly into C programs.

We shall describe the Tcl code we have designed to implement the black hole attacks in the next section.

5.2. Testing the Black Hole AODV

We have tested our implementation of the Black Hole to see whether it is correctly working or not. To be 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.

5.2.1. Simulation Parameters and Measured Metrics

To take accurate results from the simulations, we used UDP protocol. The source node keeps on sending out UDP packets, even if the malicious node drops them, while the node finishes the connection if it uses TCP protocol. Therefore, we could observe the connection flow between sending node and receiving node during the simulation. Furthermore we were able to count separately the sent and received packets since the UDP connection is not lost during the simulation. If we had used TCP protocol in our scenarios we could not count the sent or received packets since the node that starts the TCP connection will finish the connection after a while if it has not received the TCP ACK packet.

We generate a small size network that has 7 nodes and create a UDP connection between Node 2 and Node 5, 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 20 seconds and the CBR connections started at time equals to 1.0 seconds and continue until the end of the simulation, in a 79 x 659 meter flat space. We manually defined appropriate positions of the nodes to show the data flow and also introduce a movement only to Node 1 to show the changes of the data flow in the network. The Tcl script contains a Black Hole AODV for the first simulation as shown in Appendix A.

5.2.2. Evaluation of The Simulation

In the first scenario where there is not a Black Hole AODV Node, connection between Node 5 and Node 4 is correctly flawed when we look at the animation of the simulation, using NAM. Figure 19 shows the data flow from Node 2 to Node 5. When the Node 1 leaves the propagation range of the Node 2 while moving, the new

connection is established via Node 3. The new connection path is shown in Figure 20.

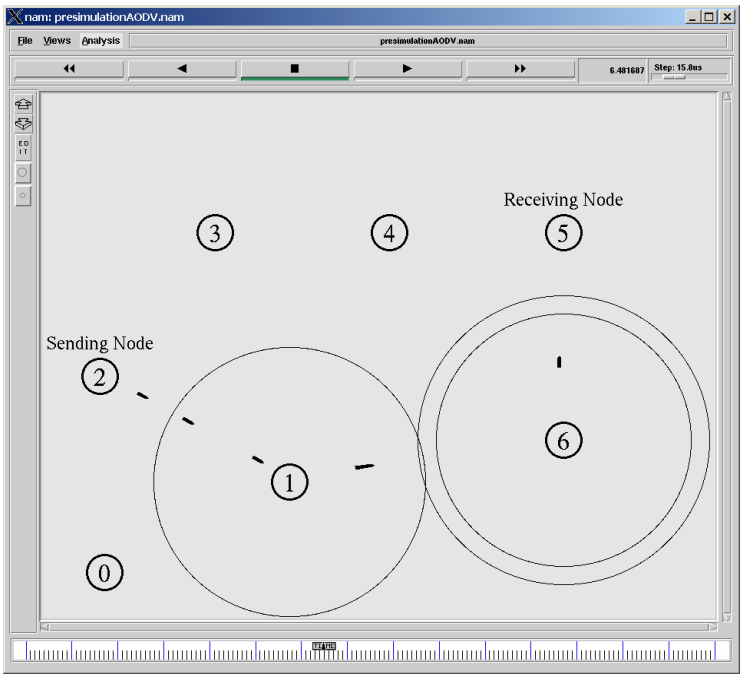

Figure 19 – Data flow between Node 2 and Node 5 via Node 1 and Node 6

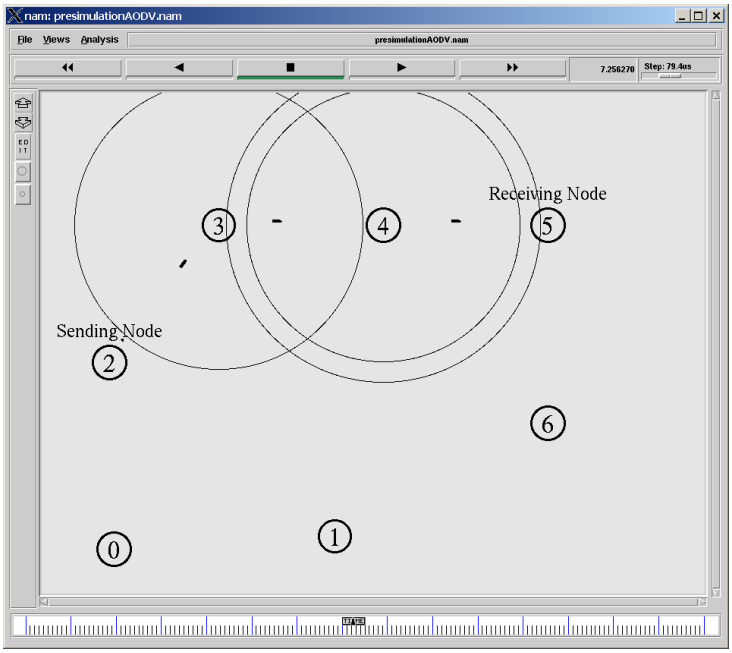

Figure 20 – Data flow between Node 2 and Node 5 via Node 3 and Node 4

In the second scenario, commenting out the three statements in the Tcl script, shown in Figure 21, we could easily add the Black Hole behavior to Node 0. The first statement, "$ns node-config -adhocRouting blackholeAODV" is to add the Black Hole AODV behavior to the nodes created from this point on. But we only define Node 0 as a Black Hole AODV and we have to change to AODV protocol after Node 0 again with the third statement. The second statement just puts a notification to Node 0 defining it as a Black Hole Node.

```
# $ns node-config          -adhocRouting blackholeAODV
set node_(0) [$ns node]
# $ns at 0.0 "$node_(0) label \"BlackHoleAODV Node\""

# $ns node-config          -adhocRouting AODV
set node_(1) [$ns node]
set node_(2) [$ns node]
$ns at 0.0 "$node_(2) label \"Sending Node\""

set node_(3) [$ns node]
set node_(4) [$ns node]
set node_(5) [$ns node]
$ns at 0.0 "$node_(5) label \"Receiving Node\""

set node_(6) [$ns node]
```

Figure 21 - Node creation and configuration in Tcl script

Node 0 being a Black Hole AODV Node absorbs the packets in the connection from Node 2 to Node 5. Figure 22 shows how the Black Hole AODV Node absorbs the traffic.

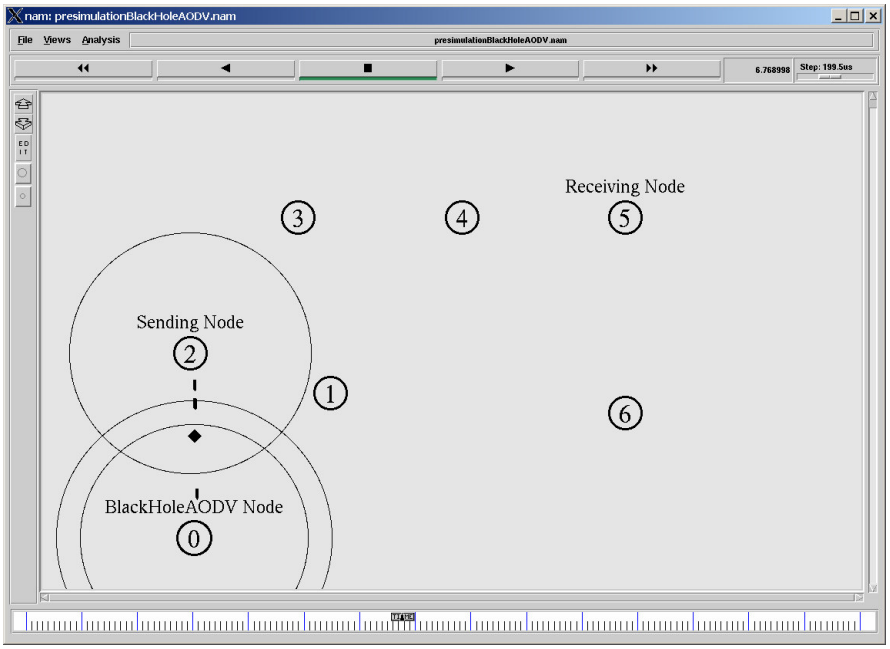

Figure 22 - Node 0 (Black Hole Node) absorbs the connection Node 2 to Node 5

In our test, we ensured that the Black Hole AODV implementation is correctly working. Then, we performed the actual simulation we will describe in the next section. Because we cannot easily see the effects of the Black Hole AODV Node in the large number of Nodes and connections, we will carry out in the actual simulation, we had to test the implementation in a small sized simulation that has a small number of nodes

5.3. Simulation of Black Hole Attack

5.3.1. Simulation Parameters and Measured Metrics

UDP connections are established between even numbered nodes (0 (zero) included) and odd numbered nodes and we used 20 nodes in the scenarios where Node 18 and Node 19 did not have a connection to any other node in the network.

In the scenarios, even numbered nodes (Node 0 - Node 16) are the sending nodes and odd numbered nodes (Node 1 - Node 17) are the receiving nodes and the even numbered nodes send the packets to the next odd numbered nodes, for example Node

0 to Node 1, Node 2 to Node 3, Node 4 to Node 5 etc. Thus, we could count the sent and received packets between any 2 nodes. In the scenarios, UDP agents are attached to the even numbered nodes and NULL agents are attached to odd numbered nodes.

In all the scenarios, we have a total of 9 connections between 18 nodes and all of these connections are always between the same nodes. But, in each scenario, every single node is placed in different coordinates and exhibits different movements. This helps us get different results with the same nodes. Node positions and movements are randomly generated by "*./setdest*", the third party application of NS and we saved in the "/scenarios" directory of the simulation root. Each scenario is named using the parameters of the "*./setdest*", for example; "*scen1forAODV-n20-t500-x750-y750*". "*./setdest*" application generates a scenario between 20 nodes that move from a random starting point to a random destination with a speed that is randomly chosen, during 500 seconds, in a 750 x 750 meter flat space.

We attach the CBR (Constant Bit Rate) application that generates constant packets through the UDP connection. Duration of the scenarios is 500 seconds and the CBR connections started at the first second of the scenario and lasts until 450. seconds of the scenarios. In our scenarios CBR parameters are;
Packet Size : 512 bytes
Data Rates : 10 Kbits
and we did not use random packets in the simulation.

The connection types are generated by "*./cbrgen*", the third party application of NS and saved as the file named "*cbr*" in the "*/scenarios*" directory of the simulation root. Same cbr connections would be created by for loop commented out in Appendix A.

Nodes in the simulation are generated by "*for*" loop statement of the Tcl language. These statements that create the nodes are shown in Figure 23. The first loop creates the first 19 nodes that use the configuration in Figure 24. "*$ns_ node-config -adhocRouting blackholeAODV*" statement changes routing protocol of the node configuration as "*blackholeAODV*" that we implemented in NS. After this statement the second loop creates the last node. Changing the *"$val(nnaodv)*" variable we can create AODV and Black Hole AODV nodes as we wish.

```
for {set i 0} {$i < $val(nnaodv)} {incr i} {
       set node_($i) [$ns_ node]
       $node_($i) random-motion 0;           # disable random motion
       }

# The last node behave as blackhole

$ns_ node-config    -adhocRouting blackholeAODV

for {set i $val(nnaodv)} {$i < $val(nn)} {incr i} {
       set node_($i) [$ns_ node]
       $node_($i) random-motion 0;           # disable random motion
       $ns_ at 0.01 "$node_($i) label \"blackhole node\""
}
```

Figure 23 - "*for*" loop statement that create wireless nodes

```
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(rp)       AODV                         ;# routing protocol
```

Figure 24 - Wireless Node Configurations

Our simulation files are named with their simulation number and "*BlackHole*" definition, for example, "sim1forBlackHole.tcl" is used for simulation 1. To compare the simulation that has the Black Hole AODV node with that does not have, we changed the "*$val(nnaodv)*" variable to 20 and put the comment "*#*" in front of the "$*ns_ node-config -adhocRouting blackholeAODV*" statement and then we copied the Tcl script in same directory changing "*BlackHole*" definition of the file name

with "*AODV*", for example, "*sim1forAODV.tcl*" is used for simulation 1. The content of the first simulation file of the Black Hole AODV is shown in Appendix A.

5.3.2. Examining The Trace File and Getting The Results

We get the simulation results from output trace file of the Tcl scripts, which has .tr extension. 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 our simulations we use "new-trace" file format that is especially used in wireless networks and includes detailed event information. The new-trace file sample is shown in Appendix B. Its fields are explained in Appendix D.

To get the results from the trace files we needed only the event type in Field 0, node id (-Ni) and trace level (-Nl) in Field 4, source address, destination address and packet type in Field 5. To identify the above information from the trace file we used "**cat**" command of UNIX and wrote its outputs to a file for all trace files of the simulations. Of all the outputs, we only need;

"*s*" value of the event information in the Field 0, to count how many CBR packets are sent by the *sending node*

"r" value of the event information in the Field 0, to count how many CBR packets are received by the receiving node

"*node id*" value of the node id information in the Field 4, for the sending nodes or receiving nodes

"*MAC*" value of the trace level information in the Field 4, to filter MAC level.

"*source address*" and "*destination address*" values of the source and destination address information in Field 5, to count the packets that goes from the sending node to the receiving node.

"*cbr*" value of the packet type information in the Field 5, to filter CBR packets.

To filter this information we used "**grep**" command of UNIX reading the file generated by "**cat**" command and gave its output to "**wc**" (word count) command of UNIX as an input to count how much information is filtered and wrote the result to a new file. For example; to count CBR packets sent by Node 0 (sending node) the command "*grep "s 0 MAC --- 0.0 1.0 cbr" sim1forBlackHole.txt | wc -l >> result.txt*" is used.

On the other hand, to count CBR packets received by Node 1 (receiving node), "*grep "r 1 MAC --- 0.0 1.0 cbr" sim1forBlackHole.txt | wc -l >> result.txt*" is used.

These commands are applied for all nodes in the all simulations and are written as a batch file. Content of this file is shown in Appendix C.

5.3.3. Evaluation of Results

Each scenario has two simulations. In the first one every node is working in cooperation with each other to keep the network in communication. The second simulation has one malicious node that carries out the Black Hole Attack. In our study, we try to compare the results of these two simulations to understand the network and node behaviors.

We first try to evaluate the packet loss. Therefore we counted how many packets are sent by the sending nodes and how many of them reached the receiving nodes. In the previous section, we described how we obtain the numbers of the packets. The tables in the Appendix E compares the normal and Black Hole networks. In the tables, the second column shows how many packets are sent by sending nodes and the third column shows how many of them reached the receiving nodes. By calculating the difference between the tables of normal and Black Hole AODV network we try to evaluate how many of the packets which could not reach the destination node are absorbed in the Black Hole Node. Packets lost in the Black Hole Node are shown in the fourth column of the table of the Black Hole network. The rest of the columns show percentage of the packets lost and additionally in the table of Black Hole

network, we added percentage of loss packets which are absorbed in the Black Hole Node.

We noticed that the percentage of data loss of the Black Hole AODV is increased more than the normal AODV network simulations in all scenarios. The first table of the Appendix G shows how the packet loss has increased.

We also understand from tables the packet loss already exists in the network. This is because packets drop at the node interface queue due to the density of data traffic. To minimize the data traffic we alter node and packet parameters. Needing to evaluate the Black Hole effect in the network, we have to minimize the packet loss which happens at the network, except the Black Hole. In a wireless ad-hoc network which does not have any Black Hole, the data traffic might be dense and packets might get lost, for instance in a FTP traffic. In our simulations of normal AODV network, we saw that data loss is increased up to %40 when we change parameters. Therefore, the data loss does not always say there was a Black Hole Node in the network.

6. SOLUTION FOR BLACK HOLE ATTACK AND ITS EFFECTS

In the two previous chapters, we explain how Black Hole Attack is implemented in NS2 and which the results are obtained from the simulations. When we examine the trace file of the simulations that include one black hole node, we saw that after a while second RREP message came to source node from the real destination node.

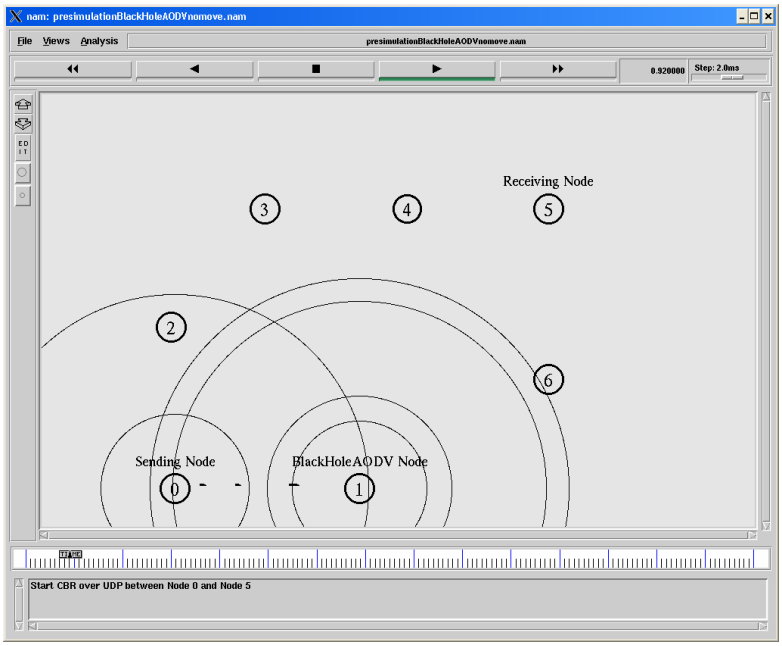

Figure 25 – Test simulation to show two RREP message

To figure out how the second packet came to source node, we created a simulation scenario with node positions shown in Figure 25. In the scenario, Node 0 is the sending node, Node 1 is black hole node and Node 5 is the receiving node. In Table 3 we can easily see these two RREP messages. The first RREP message came from the black hole node (Node 1) and reached the source node (Node 0) at "0.205976533" of simulation time. The second RREP message arrived from the destination node (Node 5) and reached the Sending Node at "1.276544989" of simulation time.

			MAC Header		IP Header		AOVD Packet			
Event	Time (-t)	Node ID (-Ni)	Destination Address (-Md)	Souce Address (-Ms)	Destination IP.Port Address (-Id)	Source IP.Port Address (-Is)	Packet Type (-Pc)	Destination Node (-Pd)	Destination Seq No (-Pds)	Hop Count (-Ph)
r	0.205976533	0	0	1	0.255	1.255	REPLY	5	-1	1
r	1.276544989	0	0	2	0.255	5.255	REPLY	5	4	4

Table 3 – Receving two RREP messages

As the black hole send an RREP message witout checking the tables, we assume that it is more likely for the first RREP to arrive from the Black Hole. In some cases, this idea may not work. For instance; the second RREP can be received at source node from an intermediate node which has fresh enough information about the destination node or the second RREP message may also came from the black hole node if the real destination node is nearer than the black hole node. These examples are extendable according to node condition in the network topology. In our work, we tried to find how this solution eliminates the black hole effects in an AODV network and if it deteriorates the network performance.

6.1. Implementing the Solution in NS-2

To evaluate effects of the proposed solution, we first needed to implement it in NS-2. Therefore, we cloned the "aodv" protocol, changing it to "idsaodv" as we did "blackholeaodv" before. To implement the black hole we changed the receive RREP function (recvRequest) of the blackholeaodv.cc file but to implement the solution we had to change the receive RREP function (recvReply) and create RREP caching mechanism to count the second RREP message.

Figure 26 shows the RREP chaching mechanism. "rrep_insert" function is for adding RREP messages, "rrep_lookup" function is for looking any RREP message up if it is exist, "rrep_remove" function is for removing any record for RREP message that arrived from defined node and "rrep_purge" function is to delete periodically from the list if it has expired. We chose this expire time "*BCAST_ID_SAVE*" as 6 (means 3 seconds).

```
void
idsAODV::rrep_insert(nsaddr_t id) {
  idsBroadcastRREP *r = new idsBroadcastRREP(id);
  assert(r);
  r->expire = CURRENT_TIME + BCAST_ID_SAVE;
  r->count ++;
  LIST_INSERT_HEAD(&rrephead, r, link);
}

idsBroadcastRREP *
idsAODV::rrep_lookup(nsaddr_t id) {
  idsBroadcastRREP *r = rrephead.lh_first;
  for( ; r; r = r->link.le_next) {
if (r->dst == id)
return r;
}
  return NULL;
}

void
idsAODV::rrep_remove(nsaddr_t id) {
  idsBroadcastRREP *r = rrephead.lh_first;
  for( ; r; r = r->link.le_next) {
  if (r->dst == id)
LIST_REMOVE(r,link);
delete r;
break;
}
}

void
idsAODV::rrep_purge() {
  idsBroadcastRREP *r = rrephead.lh_first;
  idsBroadcastRREP *rn;
  double now = CURRENT_TIME;
  for(; r; r = rn) {
rn = r->link.le_next;
if(r->expire <= now) {
LIST_REMOVE(r,link);
delete r;
}
}
}
```

Figure 26 – RREP Caching Mechanism

In the “recvReply” function, we first control if the RREP message arrived for itself and if it did, function looks the RREP message up if it has already arrived. If it did not, it inserts the RREP message for its destination address and returns from the function. If the RREP message is cached before for the same destination address, normal RREP function is carried out. Afterwards, if the RREP message is not meant for itself the node forwards the message to its appropriate neighbor. Figure 27 shows how the receive RREP message function of the idsaodv is carried out.

```
idsAODV::recvReply(Packet *p) {
idsBroadcastRREP * r = rrep_lookup(rp->rp_dst);
   if(ih->daddr() == index) {
       if (r == NULL) {
          count = 0;
          rrep_insert(rp->rp_dst);
       } else {
          r->count ++;
          count = r->count;
       }
       UPDATE ROUTE TABLE
   } else {
       Forward(p);
   }
}
```

Figure 27 - Receive RREP function of the IDSAODV

6.2. Testing the IDSAODV

Having implemented the IDSAODV 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 are the same as in the test simulation of the two RREP messages, shown in Figure 25. In this simulation IDSAODV protocol is used instead of AODV for all nodes except the black hole node (Node 1). To change the AODV protocol to IDSAODV 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 28 shows that CBR packets are reaching the destination node as expected.

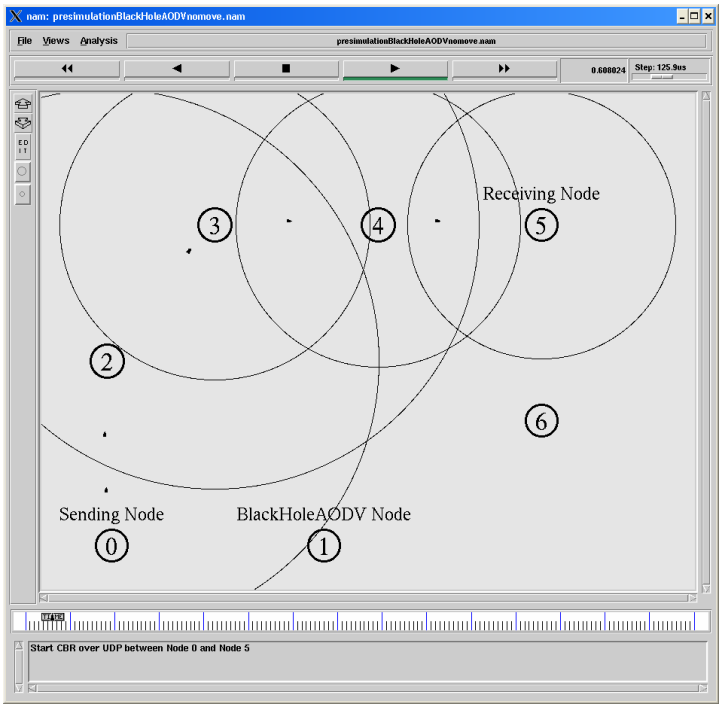

Figure 28 - CBR packet are reached to destination node properly

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 Chapter 5 to compare the performance of IDS approach.

6.3. Simulation of IDSAODV and Evaluation of Results

To be able to evaluate if our solution has succeeded we used same scenarios and simulation parameters as described at Chapter 5.3.1. and also to be able to obtain the simulation results we used a similar batch file adapted for idsaodv. The tables in Appendix F compare IDSAODV network with Black Hole network. Appendix G shows the solution affected the packet loss, but the concern we had at the beginning are valid.

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

In this study, we analyzed effect of the Black Hole in an AODV Network. For this purpose, we implemented an AODV protocol that behaves as Black Hole in NS-2. We simulated five scenarios where each one has 20 nodes that use AODV protocol and also simulated the same scenarios after introducing one Black Hole Node into the network. Moreover, we also implemented a solution that attempted to reduce the Black Hole effects in NS-2 and simulated the solution using the same scenarios. Our simulation results are analyzed below:

Having simulated the Black Hole Attack, we saw that the packet loss is increased in the ad-hoc network. In Appendix E, tables of simulation results show the difference between the number of packets lost in the network with and without a Black Hole Attack. This also shows that Black Hole Attack affects the overall network connectivity and the data loss could show the existence of the Black Hole Attack in the network. If the number of Black Hole Nodes is increased then the data loss would also be expected to increase.

We can understand from Appendix G; AODV network has normally 3,21 % data loss and if a Black Hole Node is introducing in this network data loss is increased to 92,59 %. As 3,21 % data loss already exists in this data traffic, Black Hole Node increases this data loss by 89,38 %. When we used IDSAODV protocol in the same network, the data loss decreased to 65 %. These two results show that our solution reduces the Black Hole effects by 24,38 % as packet loss in a network using IDSAODV and where there is no black holes increases to 75,62 %.

7.2. Future Work

We simulated the Black Hole Attack in the Ad-hoc Networks and investigated its affects. In our study, we used the AODV routing protocol. But the other routing protocols could be simulated as well. All routing protocols are expected to present different results. Therefore, the best routing protocol for minimizing the Black Hole Attack may be determined.

In our thesis, we try to eliminate the Black Hole effect in the network. But detection of the Black Hole Node is another future work. In our work, we assume the black hole node is detected and tried to eliminate its effects. There are many Intrusion Detection Systems (IDS) for ad-hoc networks. These IDSs could be tested to determine which one is the best to detect the Black Hole.

Our solution tries to eliminate the Black Hole effect at the route determination mechanism of the AODV protocol that is carried out before the nodes start the packets. Additionally, we used UDP connection to be able to count the packets at sending and receiving nodes. If we had used the TCP connection between nodes, the sending node would be the end of the connection, since ACK packets do not reach the sending node. This would be another solution for finding the Black Hole Node. This takes place after the route determination mechanism of the ADOV protocol and finds the route in a much longer period. Our solution finds the path in the AODV level. Finding the black hole node with connection oriented protocols could be another work as a future study.

8. LIST OF REFERENCES

[1] http://en.wikipedia.org/wiki/Personal_area_network, 25 July 2005.

[2] T. Franklin, "Wireless Local Area Networks", Technical Report http://www.jisc.ac.uk/uploaded_documents/WirelessLANTechRep.pdf. 25 July 2005.

[3] J. Reynold, "Going Wi-Fi", Chapter 6, The Wi-Fi Standards Spelled out, Pg. 77.

[4] http://certifications.wi-fi.org/wbcs_certified_products.php 25 July 2005.

[5] P. Misra,. "Routing Protocols for Ad Hoc Mobile Wireless Networks", http://www.cse.wustl.edu/~jain/cis788-99/adhoc_routing/index.html, 14 May 2006.

[6] P. Yau and C. J. Mitchell, "Security Vulnerabilities in Adhoc Network".

[7] G. Vigna, S. Gwalani and K. Srinivasan, "An Intrusion Detection Tool for AODV-Based Ad hoc Wireless Networks", Proc. of the 20th Annual Computer Security Applications Conference (ACSAC'04).

[8] P. Ning and K. Sun, "How to Misuse AODV: A Case Study of Insider Attacks Against Mobile Ad-Hoc Routing Protocols", Proc.of the 2003 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY., June 2003.

[9] S. Marti, T. J. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Ad Hoc Networks", Proc. 6th Annual Int'l. Conf. Mobile Comp. and Net., Boston, MA. pp. 255-265. August 2000.

[10] D. Johnson, D. Maltz and J. Broch, "DSR the Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks". Ad Hoc networking, Chapter 5, page 139-172. Addison-Wesley, 2001.

[11] H. Deng, W. Li and D. P. Agrawal, "Routing Security inWireless Ad Hoc Networks". University of Cincinnati, IEEE Communication Magazine, October 2002.

[12] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks", Security Protocols, 7th International Workshop Proceedings, Lecture Notes in Computer Science, 1999. University of Cambridge Computer Laboratory.

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

[14] K Fall and K. Varadhan, The NS Manual, November 18, 2005, http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf. 25 July 2005.

[15] Virtual InterNetwork Testbed, http://www.isi.edu/nsnam/vint, 14 May 2006.

[16] NS by Example, http://nile.wpi.edu/NS/overview.html, 14 May 2006.

[17] F. J. Ros and P. M. Ruiz, "Implementing a New Manet Unicast Routing Protocol in NS2", December, 2004, http://masimum.dif.um.es/nsrt-howto/pdf/nsrt-howto.pdf, 25 July 2005.

[18] Webopedia, An Internet Dictionary, 14 May 2006 http://www.webopedia.com/TERM/T/Tcl.html.

9. APPENDICES

Appendix A - sim1forBlackHole.tcl

```
# 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)             20                                      ;# total number of mobilenodes
set val(nnaodv)         19                                      ;# number of AODV mobilenodes
set val(rp)             AODV                                    ;# routing protocol
set val(x)              750                                     ;# X dimension of topography
set val(y)              750                                     ;# Y dimension of topography
set val(cstop)          451                                     ;# time of connections end
set val(stop)           500                                     ;# time of simulation end
set val(cp)             "scenarios/scen1forAODV-n20-t500-x750-y750"          ;#Connection Pattern
set val(cc)             "scenarios/cbr"                         ;#CBR Connections

# Initialize Global Variables

set ns_           [new Simulator]

$ns_ use-newtrace
set tracefd       [open sim1forBlackHole.tr w]
$ns_ trace-all $tracefd

set namtrace            [open sim1forBlackHole.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 and #2
set chan_1_ [new $val(chan)]
set chan_2_ [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 AODV nodes for simulation

puts "Creating nodes..."
for {set i 0} {$i < $val(nnaodv)} {incr i} {
      set node_($i) [$ns_ node]
      $node_($i) random-motion 0    ;#disable random motion
      }

# Creating Black Hole nodes for simulation

$ns_ node-config        -adhocRouting blackholeAODV
for {set i $val(nnaodv)} {$i < $val(nn)} {incr i} {
      set node_($i) [$ns_ node]
      $node_($i) random-motion 0    ;#disable random motion
      $ns_ at 0.01 "$node_($i) label \"blackhole node\""
}
```

```
# Adding connection pattern which is created using setdest, parameters shown below
# ./setdest -n 20 -p 1.0 -M 20.0 -t 500 -x 750 -y 750 > scen1forAODV-n20-t500-x750-y750

puts "Loading random connection pattern..."
set god_ [God instance]
source $val(cp)


# ################### CBRGEN GENERATE SAME CODE ##############################
# set j 0
#
# for {set i 0} {$i < 18} {incr i} {
#
#   #Create a UDP and NULL agents, then attach them to the appropriate nodes
#     set udp_($j) [new Agent/UDP]
#     $ns_ attach-agent $node_($i) $udp_($j)
#     set null_($j) [new Agent/Null]
#     $ns_ attach-agent $node_([expr $i + 1]) $null_($j)
#
#    #Attach CBR application;
#     set cbr_($j) [new Application/Traffic/CBR]
#     puts "cbr_($j) has been created over udp_($j)"
#     $cbr_($j) set packet_size_ 512
#     $cbr_($j) set interval_ 1
#     $cbr_($j) set rate_ 10kb
#     $cbr_($j) set random_ false
#     $cbr_($j) attach-agent $udp_($j)
#     $ns_ connect $udp_($j) $null_($j)
#     puts "udp_($j) and null_($j) agents has been connected each other"
#     $ns_ at 1.0 "$cbr_($j) start"
#
#     set j [expr $j + 1]
#     set i [expr $i + 1]
# }
# ##############################################################################

# CBR Connections generated by cbrgen
source $val(cc)
```

```
# Define initial node position

for {set i 0} {$i < $val(nn) } {incr i} {
      $ns_ initial_node_pos $node_($i) 30
      }

# CBR connections stops

for {set i 0} {$i < 9 } {incr i} {
      $ns_ at $val(cstop) "$cbr_($i) stop"
      }

# Tell all nodes when the simulation ends

for {set i 0} {$i < $val(nn) } {incr i} {
      $ns_ at $val(stop).000000001 "$node_($i) 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).00000001 "puts \"NS EXITING...\" ; $ns_ halt"


proc finish {} {
      global ns_ tracefd namtrace
      $ns_ flush-trace
      close $tracefd
      close $namtrace
#     exec nam sim1forBlackHole.nam &
      exit 0
      }

puts "Starting Simulation..."
$ns_ run
```

Appendix B – Trace File Example

```
s -t 1.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 252.64 -Ny 235.76 -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 4

r -t 1.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 252.64 -Ny 235.76 -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 4

s -t 1.000000000 -Hs 2 -Hd -2 -Ni 2 -Nx 245.06 -Ny 433.66 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 2.0 -Id 3.0 -It cbr -Il 512 -If 0 -Ii 1 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

r -t 1.000000000 -Hs 2 -Hd -2 -Ni 2 -Nx 245.06 -Ny 433.66 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 2.0 -Id 3.0 -It cbr -Il 512 -If 0 -Ii 1 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

s -t 1.000000000 -Hs 4 -Hd -2 -Ni 4 -Nx 240.73 -Ny 48.04 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 4.0 -Id 5.0 -It cbr -Il 512 -If 0 -Ii 2 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 4

r -t 1.000000000 -Hs 4 -Hd -2 -Ni 4 -Nx 240.73 -Ny 48.04 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 4.0 -Id 5.0 -It cbr -Il 512 -If 0 -Ii 2 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 4

s -t 1.000000000 -Hs 6 -Hd -2 -Ni 6 -Nx 564.15 -Ny 416.62 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 6.0 -Id 7.0 -It cbr -Il 512 -If 0 -Ii 3 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

r -t 1.000000000 -Hs 6 -Hd -2 -Ni 6 -Nx 564.15 -Ny 416.62 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 6.0 -Id 7.0 -It cbr -Il 512 -If 0 -Ii 3 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

s -t 1.000000000 -Hs 8 -Hd -2 -Ni 8 -Nx 521.86 -Ny 462.74 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 8.0 -Id 9.0 -It cbr -Il 512 -If 0 -Ii 4 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

r -t 1.000000000 -Hs 8 -Hd -2 -Ni 8 -Nx 521.86 -Ny 462.74 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 8.0 -Id 9.0 -It cbr -Il 512 -If 0 -Ii 4 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 1

s -t 1.000000000 -Hs 10 -Hd -2 -Ni 10 -Nx 77.85 -Ny 399.97 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 10.0 -Id 11.0 -It cbr -Il 512 -If 0 -Ii 5 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2

r -t 1.000000000 -Hs 10 -Hd -2 -Ni 10 -Nx 77.85 -Ny 399.97 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 10.0 -Id 11.0 -It cbr -Il 512 -If 0 -Ii 5 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2

s -t 1.000000000 -Hs 12 -Hd -2 -Ni 12 -Nx 71.03 -Ny 536.86 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 12.0 -Id 13.0 -It cbr -Il 512 -If 0 -Ii 6 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2
```

```
r -t 1.000000000 -Hs 12 -Hd -2 -Ni 12 -Nx 71.03 -Ny 536.86 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 12.0 -Id 13.0 -It cbr -Il 512 -If 0 -Ii 6 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2

s -t 1.000000000 -Hs 14 -Hd -2 -Ni 14 -Nx 376.00 -Ny 662.55 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 14.0 -Id 15.0 -It cbr -Il 512 -If 0 -Ii 7 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2

r -t 1.000000000 -Hs 14 -Hd -2 -Ni 14 -Nx 376.00 -Ny 662.55 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 14.0 -Id 15.0 -It cbr -Il 512 -If 0 -Ii 7 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2

s -t 1.000000000 -Hs 16 -Hd -2 -Ni 16 -Nx 732.66 -Ny 382.51 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 16.0 -Id 17.0 -It cbr -Il 512 -If 0 -Ii 8 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 3

r -t 1.000000000 -Hs 16 -Hd -2 -Ni 16 -Nx 732.66 -Ny 382.51 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms
0 -Mt 0 -Is 16.0 -Id 17.0 -It cbr -Il 512 -If 0 -Ii 8 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 3

s -t 1.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 252.64 -Ny 235.76 -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 1.000000000 -Hs 2 -Hd -2 -Ni 2 -Nx 245.06 -Ny 433.66 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 2.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 3 -Pds 0 -Ps 2 -Pss
4 -Pc REQUEST

s -t 1.000000000 -Hs 4 -Hd -2 -Ni 4 -Nx 240.73 -Ny 48.04 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 4.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 5 -Pds 0 -Ps 4 -Pss
4 -Pc REQUEST

s -t 1.000000000 -Hs 6 -Hd -2 -Ni 6 -Nx 564.15 -Ny 416.62 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0
-Mt 0 -Is 6.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 7 -Pds 0 -Ps 6 -Pss
4 -Pc REQUEST
```

Appendix C – The File For Getting The Results From The Trace Files

```
for fn in sim1forBlackHole sim2forBlackHole sim3forBlackHole sim4forBlackHole sim5forBlackHole ; do
      rm ${fn}.tr -f
      rm ${fn}.nam -f
      rm ${fn}.txt -f
done

for fn in sim1forAODV sim2forAODV sim3forAODV sim4forAODV sim5forAODV ; do
      rm ${fn}.tr -f
      rm ${fn}.nam -f
      rm ${fn}.txt -f
done

rm result.txt -f
Echo Files are deleted
Echo

for fn in sim1forBlackHole sim2forBlackHole sim3forBlackHole sim4forBlackHole sim5forBlackHole ; do
      ns ${fn}.tcl
      Echo ${fn}.tcl is interpreted
      echo ------------------------------------------------------------------
done

for fn in sim1forAODV sim2forAODV sim3forAODV sim4forAODV sim5forAODV ; do
      ns ${fn}.tcl
      Echo ${fn}.tcl is interpreted
      echo ------------------------------------------------------------------
done
echo

for fn in sim1forBlackHole sim2forBlackHole sim3forBlackHole sim4forBlackHole sim5forBlackHole ; do
      cat ${fn}.tr | awk  '{print $1 " " $9 " " $19 " " $21 " " $31 " " $33 " " $35}' >> ${fn}.txt
      Echo ${fn}.txt is created
done

for fn in sim1forAODV sim2forAODV sim3forAODV sim4forAODV sim5forAODV ; do
      cat ${fn}.tr | awk  '{print $1 " " $9 " " $19 " " $21 " " $31 " " $33 " " $35}' >> ${fn}.txt
      Echo ${fn}.txt is created
done
```

```
echo

for fn in sim1forBlackHole sim2forBlackHole sim3forBlackHole sim4forBlackHole sim5forBlackHole ; do
      # Simulation results for $fn
      echo Simulation_results_for_$fn >> result.txt
      ss=0;
      sr=0;
      for i in 0 2 4 6 8 10 12 14 16; do
            j=`expr $i + 1`
            s=`grep "s $i MAC --- ${i}.0 ${j}.0 cbr" ${fn}.txt | wc -l`
            r=`grep "r $j MAC --- ${i}.0 ${j}.0 cbr" ${fn}.txt | wc -l`
            d=`grep "d 19 RTR LOOP ${i}.0 ${j}.0 cbr" ${fn}.txt | wc -l`
            ss=`expr $ss + $s`
            sr=`expr $sr + $r`
            echo "$s    $r    $d" >> result.txt
      done
      per=`expr \( $sr \* 100 \) / $ss`
      echo "Total: $per" >> result.txt
done

for fn in sim1forAODV sim2forAODV sim3forAODV sim4forAODV sim5forAODV ; do
      # Simulation results for $fn
      echo Simulation_results_for_$fn >> result.txt

      for i in 0 2 4 6 8 10 12 14 16; do
            j=`expr $i + 1`
            s=`grep "s $i MAC --- ${i}.0 ${j}.0 cbr" ${fn}.txt | wc -l`
            r=`grep "r $j MAC --- ${i}.0 ${j}.0 cbr" ${fn}.txt | wc -l`
            echo "$s    $r" >> result.txt
      done
echo Results of $fn is written into result.txt
done


echo
echo .........ALL DONE.............
```

Appendix D - 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” which consists of the type of application like 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. For values of the fields 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 E - Packet Loss of The Normal and Black Hole Network

Simulation Results of Scenario 1 for AODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1097	1061	**3,28**
Node 2 -> Node 3	1091	1057	**3,12**
Node 4 -> Node 5	1098	1066	**2,91**
Node 6 -> Node 7	1025	986	**3,80**
Node 8 -> Node 9	1087	1066	**1,93**
Node 10 -> Node 11	1117	1064	**4,74**
Node 12 -> Node 13	1091	1059	**2,93**
Node 14 -> Node 15	1083	1015	**6,28**
Node 16 -> Node 17	1096	1067	**2,65**
TOTAL	**9785**	**9441**	**3,52**

Simulation Results of Scenario 1 for Black Hole AODV					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1103	108	439	**90,21**	39,80
Node 2 -> Node 3	1100	80	381	**92,73**	34,64
Node 4 -> Node 5	1083	180	497	**83,38**	45,89
Node 6 -> Node 7	1046	54	447	**94,84**	42,73
Node 8 -> Node 9	1100	229	368	**79,18**	33,45
Node 10 -> Node 11	1093	252	562	**76,94**	51,42
Node 12 -> Node 13	1089	7	525	**99,36**	48,21
Node 14 -> Node 15	1031	13	722	**98,74**	70,03
Node 16 -> Node 17	1051	6	473	**99,43**	45,00
TOTAL	**9696**	**929**	**4414**	**90,42**	45,52

Simulation Results of Scenario 2 for AODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1067	1020	**4,40**
Node 2 -> Node 3	1090	1054	**3,30**
Node 4 -> Node 5	1011	969	**4,15**
Node 6 -> Node 7	1078	978	**9,28**
Node 8 -> Node 9	1078	983	**8,81**
Node 10 -> Node 11	1095	1074	**1,92**
Node 12 -> Node 13	729	709	**2,74**
Node 14 -> Node 15	993	969	**2,42**
Node 16 -> Node 17	911	887	**2,63**
TOTAL	**9052**	**8643**	**4,52**

Simulation Results of Scenario 2 for Black Hole AODV					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1105	250	775	**77,38**	70,14
Node 2 -> Node 3	1096	3	615	**99,73**	56,11
Node 4 -> Node 5	1088	27	505	**97,52**	46,42
Node 6 -> Node 7	1100	116	198	**89,45**	18,00
Node 8 -> Node 9	1058	34	378	**96,79**	35,73
Node 10 -> Node 11	1089	196	703	**82,00**	64,55
Node 12 -> Node 13	1110	202	597	**81,80**	53,78
Node 14 -> Node 15	1110	3	674	**99,73**	60,72
Node 16 -> Node 17	1040	96	183	**90,77**	17,60
TOTAL	**9796**	**927**	**4628**	**90,54**	47,24

Simulation Results of Scenario 3 for AODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	55	55	**0,00**
Node 2 -> Node 3	1089	1073	**1,47**
Node 4 -> Node 5	1045	1039	**0,57**
Node 6 -> Node 7	1097	1077	**1,82**
Node 8 -> Node 9	868	784	**9,68**
Node 10 -> Node 11	1025	1013	**1,17**
Node 12 -> Node 13	1073	1012	**5,68**
Node 14 -> Node 15	1089	1077	**1,10**
Node 16 -> Node 17	1099	1082	**1,55**
TOTAL	**8440**	**8212**	**2,70**

Simulation Results of Scenario 3 for Black Hole AODV					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1091	2	277	**99,82**	25,39
Node 2 -> Node 3	1098	6	61	**99,45**	5,56
Node 4 -> Node 5	1088	45	810	**95,86**	74,45
Node 6 -> Node 7	1091	6	534	**99,45**	48,95
Node 8 -> Node 9	1071	3	530	**99,72**	49,49
Node 10 -> Node 11	1092	0	437	**100,00**	40,02
Node 12 -> Node 13	1093	10	357	**99,09**	32,66
Node 14 -> Node 15	1103	26	563	**97,64**	51,04
Node 16 -> Node 17	1093	45	875	**95,88**	80,05
TOTAL	**9820**	**143**	**4444**	**98,54**	45,25

Simulation Results of Scenario 4 for AODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1094	1079	**1,37**
Node 2 -> Node 3	1095	1089	**0,55**
Node 4 -> Node 5	1109	1077	**2,89**
Node 6 -> Node 7	1076	1057	**1,77**
Node 8 -> Node 9	1091	1072	**1,74**
Node 10 -> Node 11	1035	1000	**3,38**
Node 12 -> Node 13	1060	1006	**5,09**
Node 14 -> Node 15	1101	1068	**3,00**
Node 16 -> Node 17	942	920	**2,34**
TOTAL	**9603**	**9368**	**2,45**

Simulation Results of Scenario 4 for Black Hole AODV					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1109	394	448	**64,47**	40,40
Node 2 -> Node 3	1079	99	195	**90,82**	18,07
Node 4 -> Node 5	1094	69	607	**93,69**	55,48
Node 6 -> Node 7	1086	71	489	**93,46**	45,03
Node 8 -> Node 9	1090	121	501	**88,90**	45,96
Node 10 -> Node 11	1062	130	663	**87,76**	62,43
Node 12 -> Node 13	1092	4	533	**99,63**	48,81
Node 14 -> Node 15	1153	183	77	**84,13**	6,68
Node 16 -> Node 17	944	93	405	**90,15**	42,90
TOTAL	**9709**	**1164**	**3918**	**88,01**	40,35

Simulation Results of Scenario 5 for AODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1106	1083	**2,08**
Node 2 -> Node 3	998	963	**3,51**
Node 4 -> Node 5	1065	1025	**3,76**
Node 6 -> Node 7	768	755	**1,69**
Node 8 -> Node 9	1095	1073	**2,01**
Node 10 -> Node 11	1111	1063	**4,32**
Node 12 -> Node 13	1100	1028	**6,55**
Node 14 -> Node 15	1093	1089	**0,37**
Node 16 -> Node 17	1087	1073	**1,29**
TOTAL	**9423**	**9152**	**2,88**

Simulation Results of Scenario 5 for Black Hole AODV					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1110	188	374	**83,06**	33,69
Node 2 -> Node 3	1071	114	740	**89,36**	69,09
Node 4 -> Node 5	1044	2	507	**99,81**	48,56
Node 6 -> Node 7	1175	2	443	**99,83**	37,70
Node 8 -> Node 9	1089	2	722	**99,82**	66,30
Node 10 -> Node 11	1130	3	766	**99,73**	67,79
Node 12 -> Node 13	1123	51	808	**95,46**	71,95
Node 14 -> Node 15	1115	87	699	**92,20**	62,69
Node 16 -> Node 17	1096	5	752	**99,54**	68,61
TOTAL	**9953**	**454**	**5811**	**95,44**	58,38

Appendix F - Packet Less of The IDSAODV and Black Hole Network

Simulation Results of Scenario 1 for IDSAODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1073	939	**12,49**
Node 2 -> Node 3	1060	1032	**2,64**
Node 4 -> Node 5	1055	1031	**2,27**
Node 6 -> Node 7	1041	989	**5,00**
Node 8 -> Node 9	1100	998	**9,27**
Node 10 -> Node 11	1101	972	**11,72**
Node 12 -> Node 13	1095	927	**15,34**
Node 14 -> Node 15	1090	897	**17,71**
Node 16 -> Node 17	1094	1004	**8,23**
TOTAL	**9709**	**8789**	**9,48**

Simulation Results of Scenario 1 for IDSAODV & Black Hole					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1053	351	250	**66,67**	23,74
Node 2 -> Node 3	1099	320	236	**70,88**	21,47
Node 4 -> Node 5	1109	466	274	**57,98**	24,71
Node 6 -> Node 7	1052	407	199	**61,31**	18,92
Node 8 -> Node 9	1108	397	275	**64,17**	24,82
Node 10 -> Node 11	1098	277	302	**74,77**	27,50
Node 12 -> Node 13	1090	189	534	**82,66**	48,99
Node 14 -> Node 15	1067	345	300	**67,67**	28,12
Node 16 -> Node 17	1032	258	245	**75,00**	23,74
TOTAL	**9708**	**3010**	**2615**	**68,99**	26,94

Simulation Results of Scenario 2 for IDSAODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1053	969	**7,98**
Node 2 -> Node 3	1085	1000	**7,83**
Node 4 -> Node 5	1041	991	**4,80**
Node 6 -> Node 7	1104	982	**11,05**
Node 8 -> Node 9	1060	941	**11,23**
Node 10 -> Node 11	1092	989	**9,43**
Node 12 -> Node 13	1100	1013	**7,91**
Node 14 -> Node 15	1059	1025	**3,21**
Node 16 -> Node 17	1051	965	**8,18**
TOTAL	**9645**	**8875**	**7,98**

Simulation Results of Scenario 2 for IDSAODV & Black Hole					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1087	89	272	**91,81**	25,02
Node 2 -> Node 3	1099	365	237	**66,79**	21,57
Node 4 -> Node 5	1100	271	301	**75,36**	27,36
Node 6 -> Node 7	1105	110	208	**90,05**	18,82
Node 8 -> Node 9	1076	106	347	**90,15**	32,25
Node 10 -> Node 11	1087	235	523	**78,38**	48,11
Node 12 -> Node 13	1107	600	178	**45,80**	16,08
Node 14 -> Node 15	1091	107	663	**90,19**	60,77
Node 16 -> Node 17	1024	111	141	**89,16**	13,77
TOTAL	**9776**	**1994**	**2870**	**79,60**	29,36

Simulation Results of Scenario 3 for IDSAODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1096	1069	**2,46**
Node 2 -> Node 3	1103	1028	**6,80**
Node 4 -> Node 5	1088	1079	**0,83**
Node 6 -> Node 7	1069	1053	**1,50**
Node 8 -> Node 9	808	717	**11,26**
Node 10 -> Node 11	1059	1026	**3,12**
Node 12 -> Node 13	1080	1035	**4,17**
Node 14 -> Node 15	1093	767	**29,83**
Node 16 -> Node 17	1094	1073	**1,92**
TOTAL	**9490**	**8847**	**6,78**

Simulation Results of Scenario 3 for IDSAODV & Black Hole					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1093	601	141	**45,01**	12,90
Node 2 -> Node 3	1097	116	95	**89,43**	8,66
Node 4 -> Node 5	199	63	90	**68,34**	45,23
Node 6 -> Node 7	1096	307	681	**71,99**	62,14
Node 8 -> Node 9	1044	44	463	**95,79**	44,35
Node 10 -> Node 11	1092	84	415	**92,31**	38,00
Node 12 -> Node 13	1045	155	240	**85,17**	22,97
Node 14 -> Node 15	1096	81	515	**92,61**	46,99
Node 16 -> Node 17	1086	507	423	**53,31**	38,95
TOTAL	**8848**	**1958**	**3063**	**77,87**	34,62

Simulation Results of Scenario 4 for IDSAODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1105	1054	**4,62**
Node 2 -> Node 3	1096	1089	**0,64**
Node 4 -> Node 5	1102	1022	**7,26**
Node 6 -> Node 7	1084	1036	**4,43**
Node 8 -> Node 9	1102	1072	**2,72**
Node 10 -> Node 11	1062	1004	**5,46**
Node 12 -> Node 13	1064	985	**7,42**
Node 14 -> Node 15	1112	878	**21,04**
Node 16 -> Node 17	967	937	**3,10**
TOTAL	**9694**	**9077**	**6,36**

Simulation Results of Scenario 4 for IDSAODV & Black Hole					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1104	680	310	**38,41**	28,08
Node 2 -> Node 3	1096	139	223	**87,32**	20,35
Node 4 -> Node 5	1091	584	376	**46,47**	34,46
Node 6 -> Node 7	1078	541	166	**49,81**	15,40
Node 8 -> Node 9	1090	220	349	**79,82**	32,02
Node 10 -> Node 11	1062	283	208	**73,35**	19,59
Node 12 -> Node 13	1098	575	258	**47,63**	23,50
Node 14 -> Node 15	1089	325	339	**70,16**	31,13
Node 16 -> Node 17	962	177	482	**81,60**	50,10
TOTAL	**9670**	**3524**	**2711**	**63,56**	28,04

Simulation Results of Scenario 5 for IDSAODV			
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Loss %**
Node 0 -> Node 1	1101	1088	**1,18**
Node 2 -> Node 3	1029	997	**3,11**
Node 4 -> Node 5	1078	944	**12,43**
Node 6 -> Node 7	1046	991	**5,26**
Node 8 -> Node 9	1093	1013	**7,32**
Node 10 -> Node 11	1092	1058	**3,11**
Node 12 -> Node 13	1073	955	**11,00**
Node 14 -> Node 15	1094	1088	**0,55**
Node 16 -> Node 17	1096	1061	**3,19**
TOTAL	**9702**	**9195**	**5,23**

Simulation Results of Scenario 5 for IDSAODV & Black Hole					
Sending Node -> Receiving Node	**Sent Packets**	**Received Packets**	**Black Hole Drop**	**Loss %**	**Black Hole Loss %**
Node 0 -> Node 1	1094	379	196	**65,36**	17,92
Node 2 -> Node 3	985	362	451	**63,25**	45,79
Node 4 -> Node 5	1050	313	331	**70,19**	31,52
Node 6 -> Node 7	1168	222	339	**80,99**	29,02
Node 8 -> Node 9	714	37	555	**94,82**	77,73
Node 10 -> Node 11	1098	374	275	**65,94**	25,05
Node 12 -> Node 13	998	559	163	**43,99**	16,33
Node 14 -> Node 15	1088	257	547	**76,38**	50,28
Node 16 -> Node 17	1101	213	206	**80,65**	18,71
TOTAL	**9296**	**2716**	**3063**	**70,78**	32,95

Appendix G - Comparison of The Normal and IDS AODV Network

AODV & Black Hole			
Scenarios	**Total Loss of AODV**	**Total Loss of Black Hole AODV**	**Increase**
Scenario 1	3,52	90,42	**86,90**
Scenario 2	4,52	90,54	**86,02**
Scenario 3	2,70	98,54	**95,84**
Scenario 4	2,45	88,01	**85,56**
Scenario 5	2,88	95,44	**92,56**
Average Loss	3,21	92,59	**89,38**

IDSAODV & Black Hole			
Scenarios	**Total Loss of IDS AODV**	**Total Loss of IDS Black Hole AODV**	**Increase**
Scenario 1	9,48	68,99	**59,52**
Scenario 2	7,98	79,60	**71,62**
Scenario 3	6,78	77,87	**71,10**
Scenario 4	6,36	63,56	**57,19**
Scenario 5	5,23	70,78	**65,56**
Average Loss	7,17	72,16	**65,00**