Gray Hole Attack in Mobile Ad Hoc Networks

By
DEBJIT DUTTA
SOUVIK MANDAL
ROUNAK KAR
SAIKAT DEBNATH

UNDER THE GUIDANCE OF

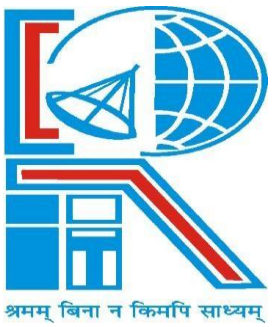
MONIKA SINGH

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Session 2015-2016



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY
[Affiliated to West Bengal University of Technology]
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA-700015

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY

TO WHOM IT MAY CONCERN

We hereby recommend that the Project entitled **Gray Hole Attack in Mobile Ad Hoc Network** prepared under my supervision by **DEBJIT DUTTA** (Reg. No.141170110027, Class Roll No. CSE/2014/072), **SOUVIK MANDAL** (Reg. No.141170110075, Class Roll No. CSE/2014/081) **ROUNAK KAR** (Reg. No.141170110053, Class Roll No. CSE/2014/067) **SAIKAT DEBNATH** (Reg. No.141170110056, Class Roll No. CSE/2014/071) of B.Tech (7^{th} Semester), may be accepted in partial fulfillment for the degree of **Bachelor of Technology in Computer Science & Engineering** under

West Bengal University of Technology
(WBUT)
.
.

……
Project Supervisor
Department of Computer Science and Engineering
RCC Institute of Information Technology

Countersigned:

……
Head
Department of Computer Sc. & Engg,
RCC Institute of Information Technology Kolkata – 700015.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY

CERTIFICATE OF APPROVAL

The foregoing Project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR
EVALUATION OF PROJECT

1. _____

2. _____

(Signature of Examiners)

ACKNOWLEDGEMENT

We are highly indebted to our guide for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

_____
DEBJIT DUTTA (CSE/2014/072)

_____
SOUVIK MANDAL (CSE/2014/081)

_____
SAIKAT DEBNATH (CSE/2014/071)

_____
ROUNAK KAR (CSE/2014/067)

TABLE OF CONTENTS

Introduction

A Mobile Ad Hoc Network (MANET) is a group of mobile nodes that cooperate and forward packets for each other. Such networks extend the limited wireless transmission range of each node by multi-hop packet forwarding, and thus they are ideally suited for scenarios in which pre-deployed infrastructure support is not available. MANETs have some special characteristic features such as unreliable wireless links used for communication between hosts, constantly changing network topologies, limited bandwidth, battery power, low computation power etc. While these characteristics are essential for the flexibility of MANETs, they introduce specific security concerns that are either absent or less severe in wired networks .MANETs are vulnerable to various types of attacks including passive eavesdropping, active interfering, impersonation, and denial-of-service. Intrusion prevention measures such as strong authentication and redundant transmission should be complemented by detection techniques to monitor security status of these networks and identify malicious behavior of any participating node(s). One of the most critical problems in MANETs is the security vulnerabilities of the routing protocols. A set of nodes may be compromised in such a way that it may not be possible to detect their malicious behavior easily. Such nodes can generate new routing messages to advertise non-existent links, provide incorrect link state information, and flood other nodes with routing traffic, thus inflicting Byzantine failure in the network. In this paper, we discuss one such attack known as Gray Hole Attack on the widely used AODV (Ad hoc On-demand Distance Vector)routing protocol in MANETs. A mechanism is presented to detect and defend the network against such an attack which may be launched cooperatively by a set of malicious nodes .The rest of the report is organized as follows. Section 2 discusses some related work on routing security in MANETs .Section 3 defines and discusses various types of gray holes attacks on MANETs. Section 4 describes the details of the proposed mechanism for detection of gray hole nodes. Section 5 presents the simulation conducted on the proposed mechanism and the performance analysis of the scheme. Section 6 concludes the paper while highlighting some future scope of work.

Literature Review

Jaydeep Sen at. el. [10] proposed a mechanism to detect gray hole attack by selecting alternate path towards the ultimate destination. They also proposed a technique to prevent ad-hoc network from this hazardous attack using alarm message and bypass malicious node. Due to irregular behavior of gray-hole attack, it is complex task to detect and prevent during communication. Proposed method increase the security mechanism and reliability factor of detecting malicious node by proactively involving the neighbor nodes of a malicious gray-hole attack. They proposed a mechanism to detect and defend the network against such an attack which may be launched cooperatively by a set of malicious nodes. The proposed security mechanism increases the reliability of detection by proactively invoking a collaborative and distributed algorithm involving the neighbor nodes of a malicious gray hole node. Detection decision works on an algorithm based on threshold cryptography. Simulation results show that the mechanism is effective and efficient with high detection rate and very low false positive rate and control overhead.

. ROUTING PROTOCOLS USED IN MANET:

Routing protocols are used to decide the path of the message packets from source to destination in a network by some set of rules. In MANET, various routing protocols are available and each of them is implemented as per the network circumstances.

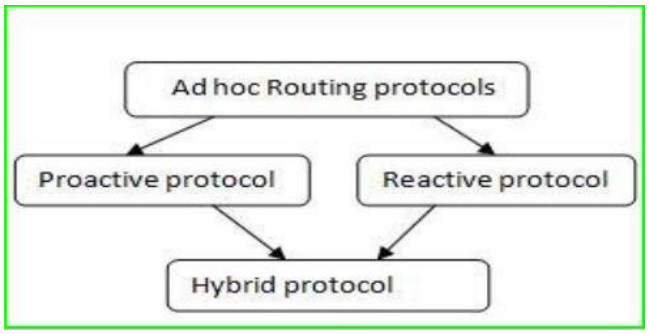


Fig 1: Routing protocol classification

The important two routing protocols in MANET are **Proactive** and **reactive**. Proactive protocols have fresh lists of destinations with their routes and they periodically distribute routing table throughout the network. While the reactive protocol looking for a route only when needed by the network and they do so with the help of Route Request packets.

Proactive Routing Protocol:

Proactive routing protocols are likewise called as table driven routing protocols. In this each node have directing table which contains data about the system topology. This component although helpful for datagram activity, causes significant movement and power utilization. The routing tables are updated occasionally at whatever point the system topology changes. Proactive protocols are not proper for large sized network as they must keep the node entries for each and every node in the routing table of every node. These protocols keep various number routing tables changing from protocol to protocol. There are different types of proactive protocols like WRP, DSDV, OLSR etc.

Reactive Routing Protocols:

Reactive routing protocol is also termed as on demand routing protocol. In this protocol , route is found at whatever point it is required. Route discovery fully rely on the nodes and it is used as per requirement. Source node checks its route cache for the possible routes from the source node to the destination node. If no route is determined, it starts route discovery process. Examples are AODV, LMR, DSR and TORA. It has two major components:

Route discovery:

During this phase the source initiates the process of route discovery only when there is a demand. The source node examines its route cache to authenticates which routes are available from source to the destination. If no route is determined, it starts a route discovery process. The packet sent by the source consists of destination node address and the address of the intermediate nodes to the destination.

Route maintenance:

As the network has a dynamic topology, sometimes there is a condition of route failure due to the link breakage. So for that route maintenance needs to be done. These protocols consists of acknowledgement mechanism which helps in route maintenance. But because of this mechanism latency gets added in the network. Every node that is involved in the route maintenance mechanism adds latency in the network. So they are generally preferred in the situations where low there is a requirement of low routing overhead.

Dynamic Source Routing (DSR):

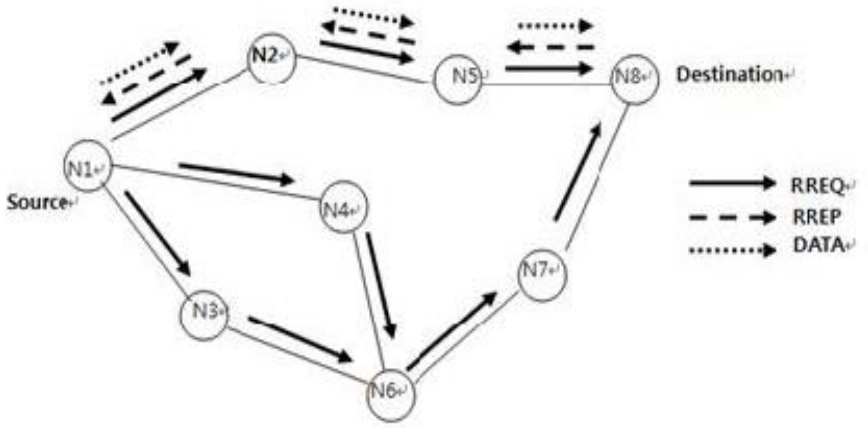
DSR is also lies in the reactive type protocol and working methodology is source route method. It works on the principle of the link state algorithm where the source initiates the route discovery only when there is a demand. First the sender will establish the route from source to destination and it also keeps a note of the address of the intermediate nodes and will save these details to the route record of the packet. This protocol was mainly designed for the multi node network having small diameter.

Ad Hoc On-Demand Distance Vector Routing (AODV):

Ad Hoc On-Demand Distance Vector Routing Protocol is working on the principle of fulfilling the demand on order. This will reduce the route broadcast messages as it will provide route on request. When source node requires transferring the packet to the destination it will simply broadcast the route-finding message to its neighbor. Once neighbor receives the broadcasted message it will do the same for transferring the packet to the destination until the packet does not reach the destination. Once the route is defined all nodes saves the route in their route table for acknowledgement to the source node. That node will then reply to the source node. Once source node will get reply from neighboring node, any route-reply after that will be removed by the source node. If network topology changes during the simulation, source node will copy the producer and again broadcast the route request message to carry on the communication in the network. If any link fault is detected, node of that link will simply send an error message to their neighbor node and the process of finding route will start again.

Control Messages in AODV

• Sequence Number and Routing Table Management. • Before starting a route discovery process, the node has to Increment its own sequence number. • A destination node has to update its own sequence number to the maximum of its current sequence number and the destination sequence number in RREQ packet immediately before transmitting the RREP packet. The sequence numbers in the routing table entries may be changed by the node only in the following circumstances. • Offer of a new route to itself, if it is the destination node. • Reception of an AODV message with new information about the sequence number for a destination. • Expiration of path or path breaks. When a node receives an AODV control message, either to create or to update a route for a particular destination, it searches its routing table for an entry to the destination. If there is no route entry, it creates a new one with the sequence number contained in the control packet, or else the sequence number is set invalid. Otherwise, the node compares the existing entry with the new information and updates it if either. • The new sequence number is higher than in the routing table entry. • The sequence numbers are equal and the new hop count plus one is smaller than in the existing route. • The sequence number is unknown. Besides the destination sequence numbers, the routing entry for each valid route contains a precursor list.



Route Reply Message (RREP):

A node having a requested identity or any intermediate node that has a route to the requested node generates a route reply RREP message back to the originator node.

Route Error Message (RERR):

Every node in the network keeps monitoring the link status to its neighbour's nodes during active routes. When the node detects a link crack in an active route, (RERR) message is generated by the node in order to notify other nodes that the link is down.

Attacks on Mobile Adhoc Network

Attacks on mobile ad hoc networks can be classified into following two categories: passive attacks and active attacks.

Passive attacks: A passive attack does not disrupt proper operation of the network. The attacker snoops the data exchanged in the network without altering it. Here, the requirement of confidentiality can be violated if an attacker is also able to interpret the data gathered through snooping. Detection of passive attacks is very difficult since the operation of the network itself does not get affected. they can reduced by using powerful encryption techniques.

Active attacks:

An active attack attempts to alter or destroy the data being exchanged in the network, thereby disrupting the normal functioning of the network. It can be classified into two categories external attacks and internal attacks. External attacks are carried out by nodes that do not belong to the network. These attacks can be prevented by using standard security mechanisms such as encryption techniques and firewalls. Both passive and active attacks can be made on any layer of the network protocol stack. Example: Wormhole, black hole, gray hole, information disclosure, resource consumption, routing attacks.

Black hole attack in AODV

Two types of black hole attack can be described in AODV in order to distinguish the kind of black hole attack. internal type of black hole attack has an internal malicious node which fits in between the routes of given source and destination. As soon as it gets the chance this malicious node make itself an active data route element.

External attacks physically stay outside of the network and deny access to network traffic or creating congestion in network or by disrupting the entire network. External attack can become a kind of internal attack when it take control of internal malicious node and control it to attack other nodes in MANET.

Wormhole attack in AODV

By categorizing the attacks into its types makes it easier for its prevention and detection so here wormhole attack has been classified as

a. Open Wormhole attack: In this type of wormhole, the attackers include themselves in the RREQ packet header following the route discovery procedure. Other nodes are aware that the malicious nodes lie on the path but they would think that the malicious nodes are direct neighbors

b. . b. Closed Wormhole Attack: The attackers do not modify the content of the packet, even the packet in a route discovery packet. Instead, they simply tunnel the packet form one side of wormhole to another side and it rebroadcasts the packet.

c. c. Half open wormhole attack: One side of wormhole does not modify the packet and only another side modifies the packet, following the route discovery procedure.

GRAY-HOLE ATTACK: A gray-hole attack is extension of black-hole attack used to bluff the source and monitoring system by partial forwarding. Here, attackers uses selective data packet dropping method to behave as genuine node and try to participate into full communication. Gray-hole malicious node participate into route discovery process and update the source route cache/ routing table as shortest path. Afterwards, source always consider malicious node as next hop node and forward packet to same. Malicious node captures all the incoming packets but drop on random basis. The complete phenomena create toughness against detection and prevention mechanism because harder because nodes can drop packets partially not only due to its malicious nature but also due to overload, congestion or selfish nature. Gray-hole attack may apply through two ways which are listed below;

1. Dropping all incoming UDP packets.
2. Partial dropping of UDP packets with random selection process.
3. Gray-hole is an attack that can switch from behaving genuine to sinkhole. Because it can act as normal node switch over to malicious node it becomes too typical to identify the state whether it us normal node or malicious node. In the ad-hoc on demand distance vector (AODV) routing process every node carry a routing table having ultimate destination and next hop information. This information is used to discover route from source to destination. Here, every node check routing table to know whether the route is available or not. In case of indirect communication it forward packets to next hop node to forward packet to destination. The gay-hole attack has two phases which are listed below;

Phase1: In this phase malicious node exploits the vulnerabilities of AODV routing protocol and update the source routing table as shortest route in next hop column. The main objective of this update is to divert all the packets to malicious node rather than genuine route. B.

Phase 2: It is the implementation phase of gray-hole attack where malicious node dropped the interrupted packets with a certain probability. A probabilistic method is use for packet selection. In the normal situation, attacker node changes the behaviors rapidly. Thus, sometime it transfer packet and some time it drop the packets. Furthermore, in the state of malicious node it also forwards some packet to create illusion of genuine nodes. Due to this behavior it is very hard to find out in the network to figure out such kind of attack.

Figure 1 shows the block representation of selective dropping

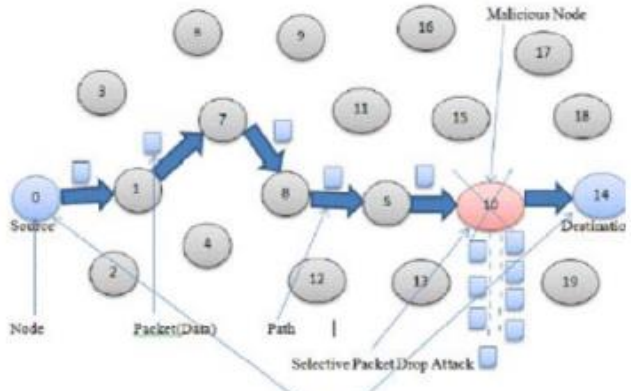


Figure 1.Gray-hole Attack

Problem Investigation

The AODV routing protocol is a popular reactive routing protocol in wireless networks, but AODV routing protocol designed for better performance of the network not for security of node, secure protocols are generally designed to have features such as authentication, integrity, confidentiality and non-repudiation. For security purpose it have vulnerabilities and it is easily manipulate by malicious node to destroy its network routing. The open nature of wireless medium also makes it easy for outsider attackers to interfere and interrupt the legitimate traffic. This concept classifies the attacks into two broad categories, namely Passive and Active attacks. In Passive attack, the adversary only eavesdrop upon the packets content, while packets may get dropped or altered on way in case of Active attacks. One of the widely known attacks is the Gray Hole Attack. It is the variation of Black hole attack. Black hole attack is one of the security threat in which the traffic is redirected to such a node that actually does not exist in the network and that node drops the entire packet. But in Gray-Hole attack, nodes will drop the packets selectively. The complete study observes that, AODV is a insecure routing protocol and does not incorporate any mechanism to detect and prevent communication from malicious affect.

SOLUTION:

The need and problem definition specifies that, proposed strategy should detect network vulnerabilities in the MANET. The study will be based on detection of GrayHole attack and prevent the network from same. Here, complete study observes that, there are several techniques proposed to detect and prevent gray-hole attack using multipath solution. Ahmed, M. et. al.[1] proposed a side technique with voting attribute to identify attacker node and create difference between trusted node and attacker node.. A dynamically strong technique has been proposed in this section which describes the complete methodology to detect and prevent malicious node. The basic idea behind the proposed technique is based on Intrusion Detection System. In the proposed solution every mobile node carries intrusion detection system which monitors the complete network structure with in-built mechanism. IDS estimate the count value of sequence number to measure the suspicious factor according to RREQ and RREP packet counting. When a suspicious value for a neighboring node exceeds a threshold, then that node is isolated from the network as other nodes do not forward packets through the suspected malicious node.

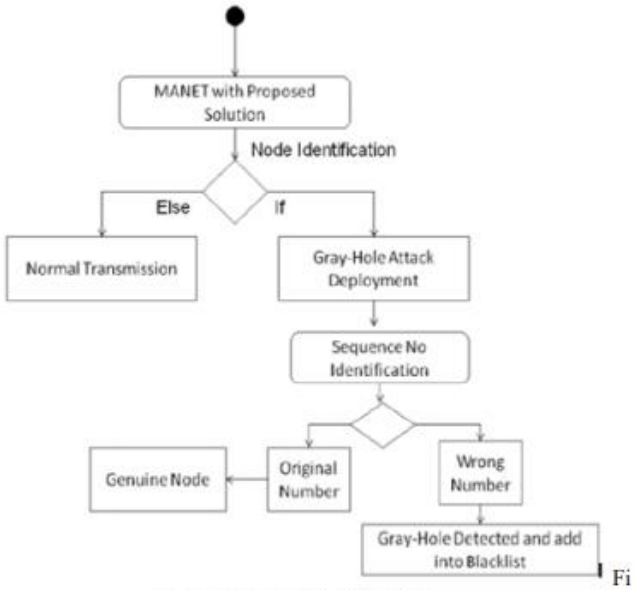


Figure 2: Proposed Architecture

Packet Delivery Ratio (PDR):

The packet delivery ratio is nothing but the ratio of data or packets send at the source to the data or packets receive at the destination. To improve the performance of the network system the packet delivery ratio must be high as possible. if the packet delivery ratio is 100 % then we can say that the network is more reliable.

Packet Loss Ratio

Packet loss ratio is again one of the metrics deal with the performance side. Packet loss occurs when one or more packets of data traveling across a computer network fail to reach their destination. Packet loss is distinguished as one of the three main error types encountered in digital communications. The other two being bit error and spurious packets caused due to noise. Packet loss can be caused by a number of factors including signal degradation over the network medium due to multi-path fading, packet drop because of channel congestion corrupted packets rejected intransit, faulty networking hardware, faulty network drivers or normal routing routines (such as DSR in ad-hoc networks).In addition to this, packet loss probability is also affected by signal-to-noise ratio and distance between the transmitter and receiver.

PROCESS FOR FINDING OUT SUSPECTED NODE OR MALICIOUS NODE IN MANET

Malicious node or suspected node indirectly affect on the adhoc network. It damages the data or packets during the network data transmission process, due to which the network gets disturb and ultimately performance of the network get reduced. This method focus on both suspected behavior of the node in the network. Once a node is recognize to be really malicious or suspected, the scheme has a notification mechanism for sending messages to all the nodes that are not yet suspected to be malicious, so that the spiteful node can be separated and not allowed to use any network resources. The mechanism consists of suspected node finding way which is invoked sequentially. This security procedure is invoked by a node when it identifies a suspicious node by examining its DRI table. We call the node that initiates the suspected node recognition procedure as the Initiator Node (IN). The IN first chooses a Cooperative Node (CN) in its neighborhood based on its DRI records and broadcasts a RREQ message to its 1- hop neighbors requesting for a route to the CN. In reply to this RREQ message the IN will receive a number of RREP messages from its neighboring nodes. It will certainly receive a RREP message from the Suspected Node (SN) if the latter is really a gray hole (since the gray holes always send RREP messages but drop data packets probabilistically). After receiving the RREP from the SN, the IN sends a probe packet to the CN through the SN. After the time to live (TTL) value of the probe packet is over, the IN enquires the CN whether it has received the probe packet. If the reply to this query is affirmative, (i.e., the probe packet is really received by the CN) then the IN updates its DRI table by making an entry „1“ under the column „Check Bit against the node ID of the SN.

History of NS2:

Ns started as a variant of the important network simulator software in 1989 and has progressed considerably over the past few years. In 1995 ns development was encouraged by agency called DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is sustained through Defense Advanced Research Projects Agency (DARPA) with SAMAN and thru National Science Foundation with CONSER, each united with alternative researchers like ACIRI. Ns has invariably enclosed substantial contributions from alternative researchers, that includes wireless code from the UCB Daedelus and CMU Monarch comes and Sun Microsystems.

What is NS2?

NS2 is an open-source simulation tool that works on Linux and Windows (using Cygwin). NS2 is a discrete event simulator. It maintains the queue of events and each event is associated with the time. It is focused at networking analysis and supplies substantial pillar for simulation of routing, multicast protocols and Internet protocols such as UDP, TCP, RTP and SRM over wired and wireless (local and satellite) networks. At each loop takes in an event, executes it and moves on to the next event executes it and moves on. NS2 is implemented in OTcl and C++. A class can be installed entirely in C++ or in OTcl. Classes that implement in C++ are typically of the lower functionality and the classes that implement in OTcl provide the flexibility of gluing different objects. It also provides hooks to configure the parameters in the C++ objects.

Design of NS2:

Two different working language of NS2 : (1) an object oriented simulator that's written in C++, and (2) a OTcl(Object oriented extension of Tcl) interpreter, accustomed to execute user's command scripts. NS contains a reach library of network and protocol objects. There are 2 class hierarchies: the compiled C++ hierarchy and the depicted OTcl one, with one on one correspondence between them. The compiled C++ hierarchy permits the user to get efficiency in the simulation and quicker execution times. This is helpful for the elaborated definition and operation of protocols. So the processing time for the packet and the event gets reduced. Then within the OTcl script provided by the user, one can define a network topology, the particular protocols and applications that we want to simulate and the kind of the output that we expect from the simulator. The OTcl uses the objects which are compiled in C++ through an OTcl linkage that makes a matching of OTcl object for each of the C++ codes.

Tool Command Language (TCL):

The inventor of TCL programing language is John Ousterhout. It is a very dynamic as well as powerful programing language that is easy to learn. It can be used widely for web and desktop applications, testing, networking and much more. It is an open source programing language that can work with different platforms (allows a very easy integration with other languages) and can be deployed easily. It has a graphical user interface development toolkit called a tk; which is used to make desktop applications very easily then the conventional approaches.

DOWNLOADING AND INSTALLING NS-2 AND NAM IN UBUNTU 16.04:

NS-2 works on Linux platform but you can also run it on windows by using Cygwin software.

Installation steps:

First, download the ns all in one package (ns-allin-one) for ns2 from the link https://sourceforge.net/projects/nsnam/files/latest/download. The name of the downloaded package will be "ns- allinone-2.35.tar.gz". Copy it to the home folder. Then open terminal and use the following two commands to epand the contents of the package.

```
cd ~/
tar -xvzf ns-allinone-2.35.tar.gz
```

Using this command all the files will be extracted into a folder called "ns-allinone-2.35". You can also extract the files directly without using this command. It is important for the NS2 that a few packages to be preinstalled. It also requires the GCC- version 4.3 to work properly. So all the package in the NS2 will be installed by using the command mentioned below:

```
sudo apt-get install build-essential autoconf automake libxmu-dev
```

One of the reliance declared is the compiler GCC-4.3, which is not available for a long time, and because of that we must nedd to install GCC-4.4 version. The version 4.4 is the previous version we can get. To install it we need to use the following command:

```
sudo apt-get install gcc-4.4
```

Navigate to the folder "link-state", use the following command. Here it is supposed that the expanded ns folder is in the home folder of your system

SOFTWARE TOOLS USED WITH NS-2:

4.1.NAM:

Nam gives a clear to be seen interpretation of the network configuration developed. As a part of the VINT project,

A new application has been developed.

NAM's features are as follows:

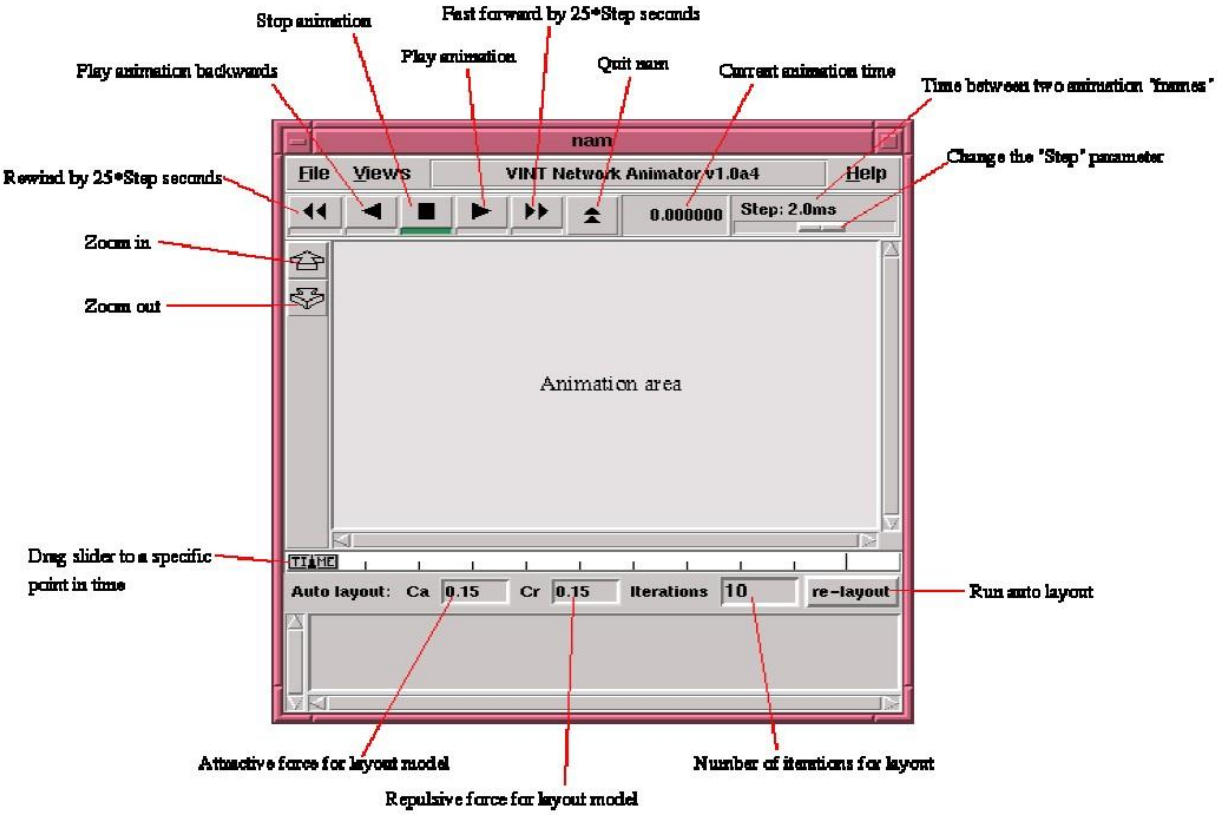
- It gives a visual Elucidation of the network created.

NAM can be executed directly from the TCL script

We can control the simulation by different buttons like play, stop, rewind, fast forward, pause, display speed controller and a packet monitor facility.

It displays information such as throughput and number of packets on each link.

We can just use drag and drop function to create network topology



Nscript:

Nscript is developed in Java. It is a GUI used for building NS-TCL scripts. The network topology can be built by

simply drawing it in the edit screen. The Nodes & agents can be added by using the simple drag and drop function in the edit screen. Once we create the network topology, the TCL code for this topology is automatically generated in the TCL script screen.

The Nscript is used to create different network topologies by simply adding nodes and their respective links. Transport agents like UDP, TCP can be created and added in the topology and the simulation events can be scheduled. The scripts created using Nscript can be exported and executed in NS-2.

TRACE DATA ANALYZING APPLICATIONS:

The applications used for analyzing trace files produced from the simulation are XGraph and TraceGraph.

5.1.X-Graph:

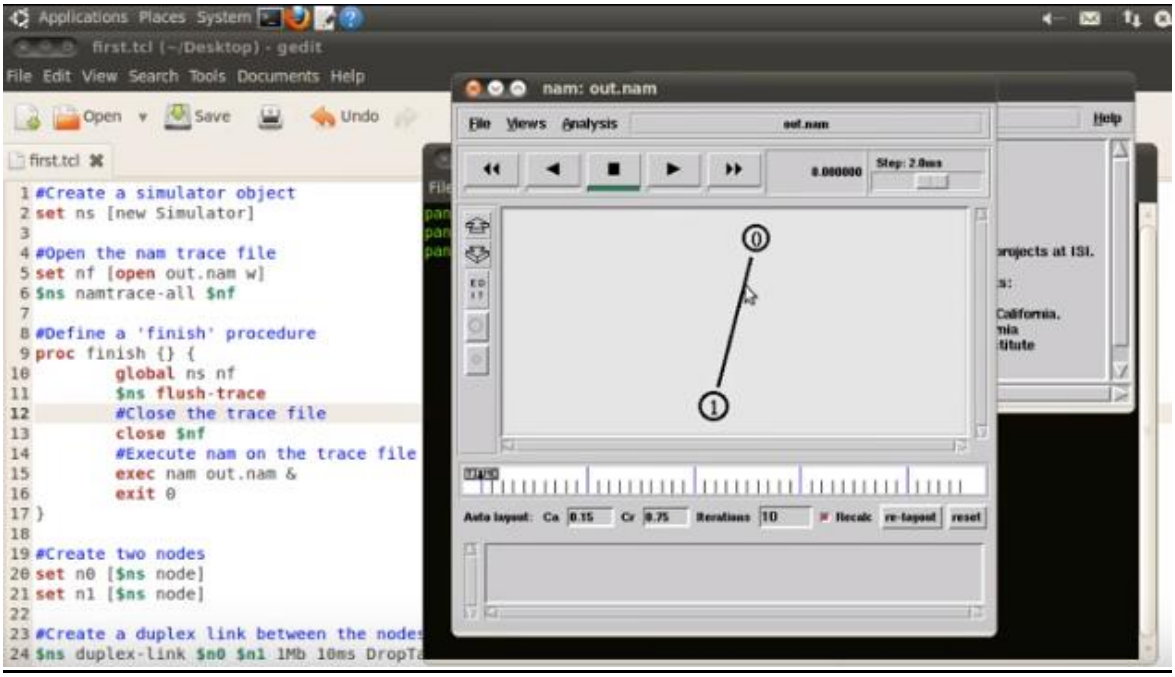
The XGraph is used for lucrative plotting and generating graphs. To use XGraph in NS-2, it should be called within a TCL Script. It will load a graph showing the visual information of the trace file produced in the simulation.

SIMULATION

The first Tcl script to add two node:

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
        global ns nf
        $ns flush-trace
        close $nf
        exec nam out.nam &
        exit 0
}
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns at 5.0 "finish"
$ns run
```

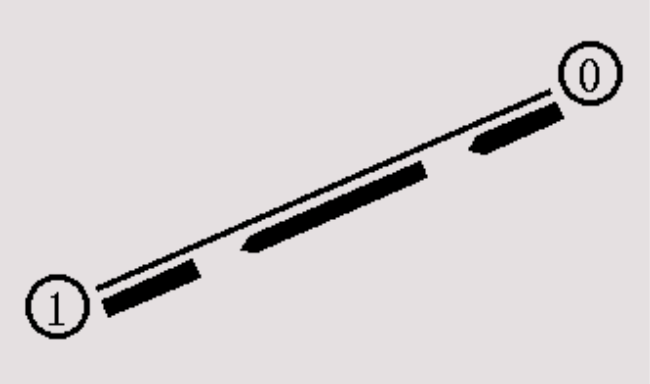
Generated nam:



Sending data between two node:
Tcl script:

```
#Create a UDP agent and attach it to node n0 set
udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to udp0 set
cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $udp0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

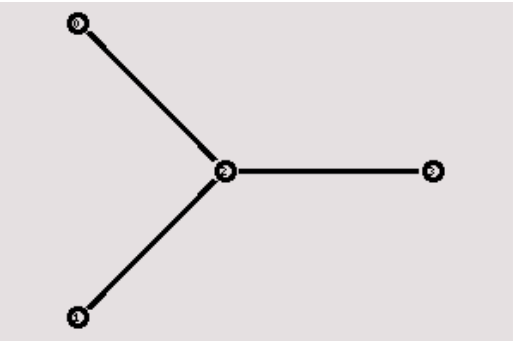
Generated nam:



Tcl script to add three node:

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms DropTail
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
```

Generated nam:

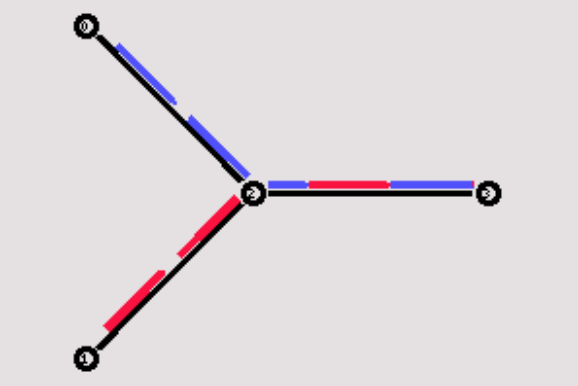


Sending data between three node:
Tcl script:

```
#Create a UDP agent and attach it to node n0 set
udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to udp0 set
cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

#Create a UDP agent and attach it to node n1 set
udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1

# Create a CBR traffic source and attach it to udp1 set
cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
```

```
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1

set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
$ns connect $udp0 $null0
$ns connect $udp1 $null0
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
$udp0 set class_ 1
$udp1 set class_ 2
```

Generated nam:



Adding malicious node for greyhole attack:

Adding a malicious node is ns2 using aodv protocol. The node which is declared as malicious will simply drop the router packet (DROP_RTR_ROUTE_LOOP).

Two files have to be modified.
1. aodv.h
2. aodv.cc

aodv.h file changes

Declare a boolean variable malicious as shown below in the protected scope in the class AODV

```
bool malicious;
```

aodv.cc file changes

1. Initialize the malicious varible with a value "false". Declare it inside the constructor as shown below

```
AODV::AODV(nsaddr_t id):Agent(PT_AODV)...
{
.....
malicious = false;
}
```

2. Add the following statement to the aodv.cc file in the "if(argc==2)" statment.

```
if(strcmp(argv[1], "malicious") == 0) {
    malicious = true;
   return TCL_OK;
}
```

3. Implement the behavior of the malicious node by setting the following code in the rt_resolve(Packet *p) function. The malicious node will simply drop the packet as indicated below.

```
if(malicious==true)
{
drop(p,DROP_RTR_ROUTE_LOOP);
}
```
Once done, recompile ns2 as given below
Open Terminal -> Go to ~ns-2.35/ directory and type the command make to compile

```
$ cd /home/pradeep/ns-allinone-2.35/ns-2.35/
$ make clean
$ make                                  # it will take time to compile
$ sudo make install
```

Once the compilation is done, Check the malicious behavior using the Tcl Script by setting any one node as malicious node. The command to set the malicious node is

```
$ns at 0.0 "[$n(1) set ragent_] malicious"
```

The variable referred for node2 is n1 (set n(1) [$ns node])

you can disable the packet dropping by adding # before above line

```
#$ns at 0.0 "[$n(1) set ragent_] malicious"
```

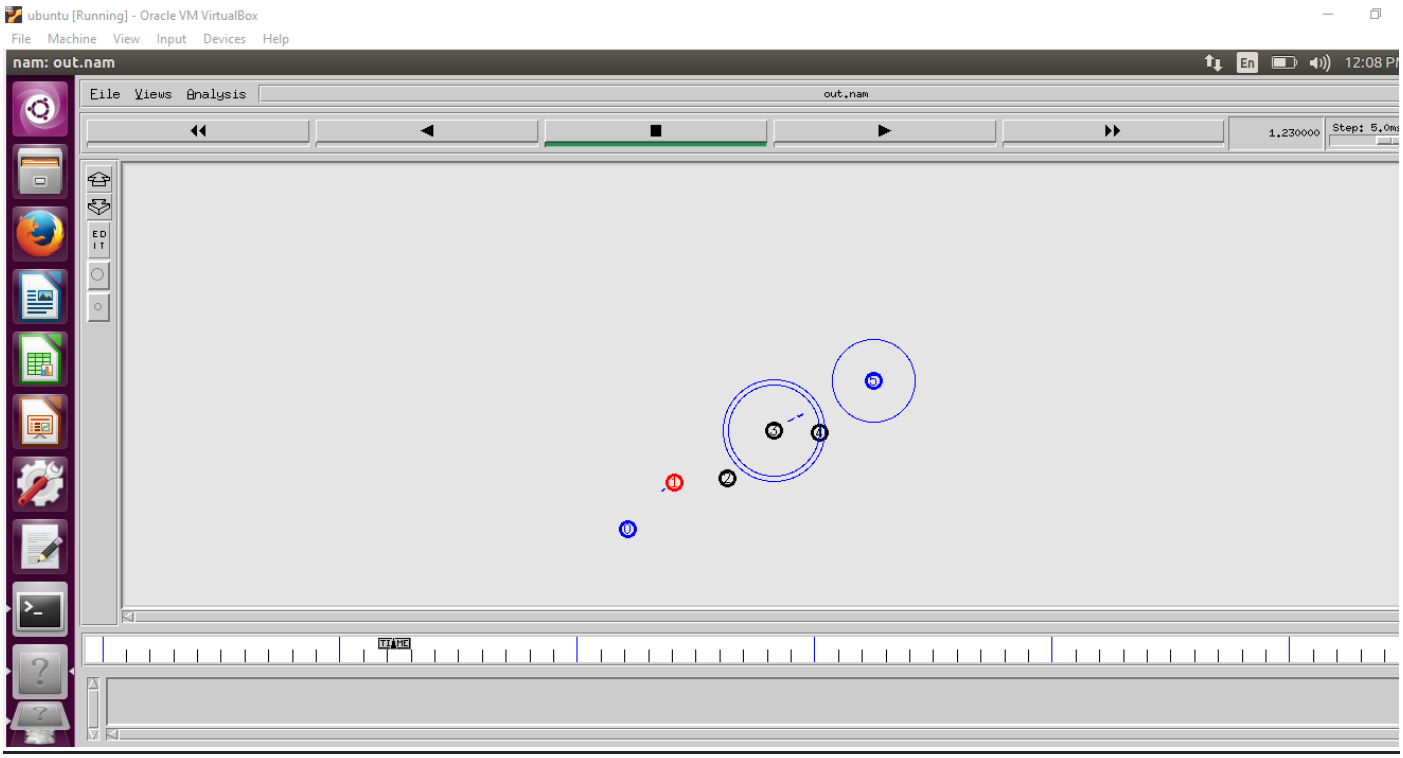
Tcl Script for adding malicious node:

```
#===================================================================
# Define options
#===================================================================
 set val(chan)         Channel/WirelessChannel  ;# channel type
 set val(prop)         Propagation/TwoRayGround ;# radio-propagation model
 set val(ant)          Antenna/OmniAntenna      ;# Antenna type
 set val(ll)           LL                       ;# Link layer type
 set val(ifq)          Queue/DropTail/PriQueue  ;# Interface queue type
 set val(ifqlen)       50                       ;# max packet in ifq
 set val(netif)        Phy/WirelessPhy          ;# network interface type
 set val(mac)          Mac/802_11               ;# MAC type
 set val(nn)           6                        ;# number of mobilenodes
 set val(rp)           AODV                     ;# routing protocol
 set val(x)            800
 set val(y)            800

set ns [new Simulator]
#ns-random 0
set f [open out.tr w]
$ns trace-all $f
set namtrace [open out.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
set topo [new Topography]
$topo load_flatgrid 800 800

create-god $val(nn)

set chan_1 [new $val(chan)]
set chan_2 [new $val(chan)]
set chan_3 [new $val(chan)]
set chan_4 [new $val(chan)]
set chan_5 [new $val(chan)]
set chan_6 [new $val(chan)]

# CONFIGURE AND CREATE NODES

$ns node-config  -adhocRouting $val(rp) \
         -llType $val(ll) \
               -macType $val(mac) \
               -ifqType $val(ifq) \
               -ifqLen $val(ifqlen) \
```

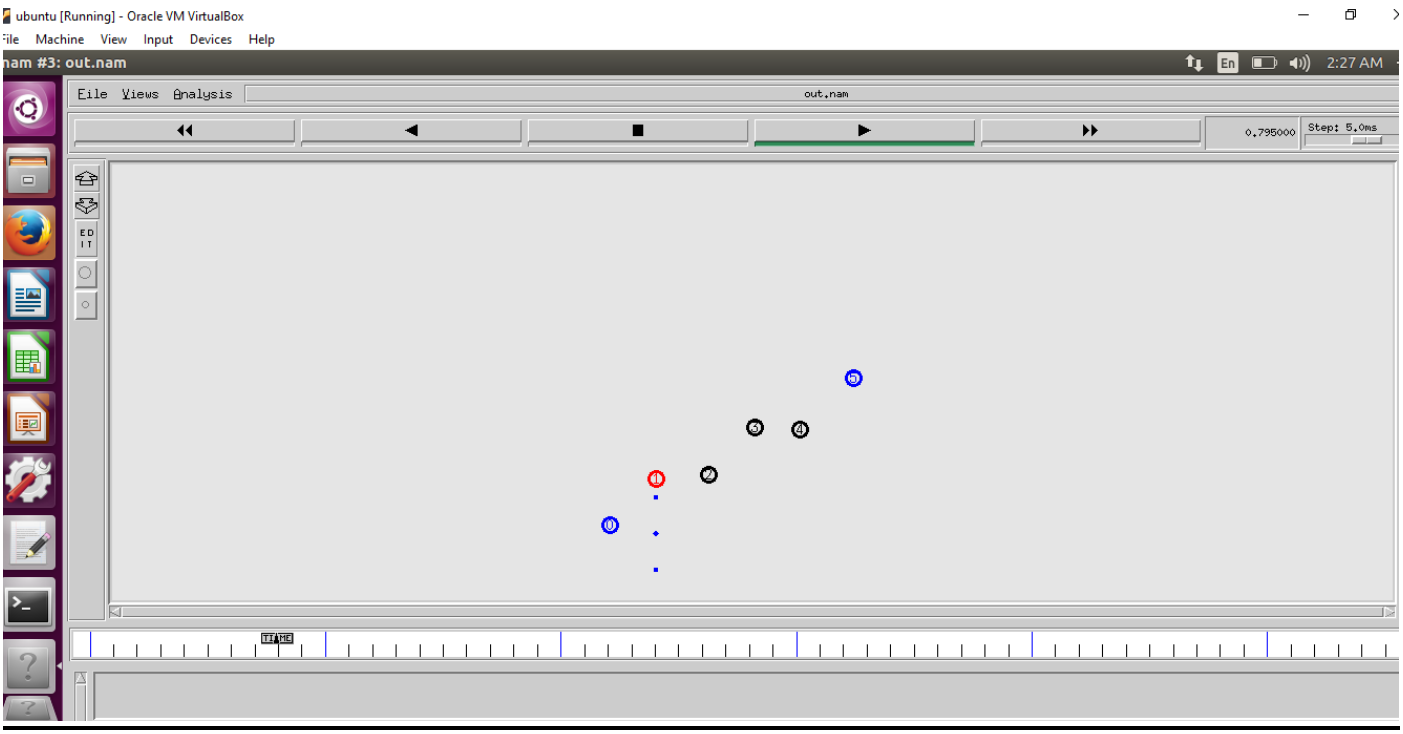
```
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                #-channelType $val(chan) \
                -topoInstance $topo \
                -agentTrace ON \
                -routerTrace ON \
                -macTrace ON \
                -movementTrace OFF \
                -channel $chan_1

proc finish {} {
    global ns namtrace
    $ns flush-trace
        close $namtrace
        exec nam -r 5m out.nam &
    exit 0
}

# define color index
$ns color 0 blue
$ns color 1 red
$ns color 2 chocolate
$ns color 3 red
$ns color 4 brown
$ns color 5 tan
$ns color 6 gold
$ns color 7 black

set n(0) [$ns node]
$ns at 0.0 "$n(0) color blue"
$n(0) color "0"
$n(0) shape "circle"
set n(1) [$ns node]
$ns at 0.0 "$n(1) color red"
$n(1) color "blue"
$n(1) shape "circle"
set n(2) [$ns node]
$n(2) color "tan"
$n(2) shape "circle"
set n(3) [$ns node]
$n(3) color "red"
$n(3) shape "circle"
set n(4) [$ns node]
$n(4) color "tan"
$n(4) shape "circle"
set n(5) [$ns node]
$ns at 0.0 "$n(5) color blue"
$n(5) color "red"
$n(5) shape "circle"


for {set i 0} {$i < $val(nn)} {incr i} {
    $ns initial_node_pos $n($i) 30+i*100
}
#$ns at 0.0 "[$n(1) set ragent_] malicious"

$ns at 0.0 "$n(0) setdest 100.0 100.0 3000.0"
$ns at 0.0 "$n(1) setdest 200.0 200.0 3000.0"
$ns at 0.0 "$n(2) setdest 300.0 200.0 3000.0"
$ns at 0.0 "$n(3) setdest 400.0 300.0 3000.0"
$ns at 0.0 "$n(4) setdest 500.0 300.0 3000.0"
$ns at 0.0 "$n(5) setdest 600.0 400.0 3000.0"
```

```
# CONFIGURE AND SET UP A FLOW

set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]
set sink3 [new Agent/LossMonitor]
set sink4 [new Agent/LossMonitor]
set sink5 [new Agent/LossMonitor]
$ns attach-agent $n(0) $sink0
$ns attach-agent $n(1) $sink1
$ns attach-agent $n(2) $sink2
$ns attach-agent $n(3) $sink3
$ns attach-agent $n(4) $sink4
$ns attach-agent $n(5) $sink5

#$ns attach-agent $sink2 $sink3
set tcp0 [new Agent/TCP]
$ns attach-agent $n(0) $tcp0
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
set tcp2 [new Agent/TCP]
$ns attach-agent $n(2) $tcp2
set tcp3 [new Agent/TCP]
$ns attach-agent $n(3) $tcp3
set tcp4 [new Agent/TCP]
$ns attach-agent $n(4) $tcp4
set tcp5 [new Agent/TCP]
$ns attach-agent $n(5) $tcp5


proc attach-CBR-traffic { node sink size interval } {
   #Get an instance of the simulator
   set ns [Simulator instance]
   #Create a CBR  agent and attach it to the node
   set cbr [new Agent/CBR]
   $ns attach-agent $node $cbr
   $cbr set packetSize_ $size
   $cbr set interval_ $interval

   #Attach CBR source to sink;
   $ns connect $cbr $sink
   return $cbr
  }

set cbr0 [attach-CBR-traffic $n(0) $sink5 1000 .030]
$ns at 0.5 "$cbr0 start"
$ns at 5.5 "finish"
puts "Start of simulation.."
$ns run
```

Generated nam without packet loss:



Generated nam with packet loss:



Detection Process:

Gray Hole Attack detection process by source node:

Data packets are divided into k equal parts. A message is sent to the destination which will be having the number of messages. After that messages are broadcasted to all neighbouring nodes of the route. After making sure that the destination node knows count of messages sent, source will start to send data. A timer is set until number of data packets are received at the destination node. If the number declared previously is less than the packets received at the destination then the removal process for gray hole attack is initiated. If after termination of the timer any message is not received at the destination then also the removal process is initiated.

Gray Hole Attack detection process by the destination node:

After sending the number of data packets to be sent by the source node timer is set to zero and data packets are counted at the initiation of the timer. After timeout the number of data packet number received are sent to the source node.

Gray Hole Attack detection process by the neighbouring nodes:

When monitoring message is received by the source node, counter is started by each neighbouring node which will count the number of data packets.

1. Evaluation of Results

There are two simulations for each scenarios. To keep the communication on in the network every node work in cooperation with each other. The second simulation will be having one adversary node which carries out the Gray Hole Attack. In our study, we will try to compare the results of the two simulations so that network and node behaviours can be understood. The packet loss is tried to be evaluated first. Therefore we will count the number of data packets sent by the sending node and the number of data packets received by the receiving node. In previous section it is described how we will get the number of packets. And then the network having Gray hole network and the network which is not having it is compared. Then it is noticed by us that how many packets are sent by the sending node and how much packets are received by the receiving node. After that it is calculated that how many packets will reach the destination node and how much packets will drop or absorbed by the Gray Hole Node, by calculating the difference of the tables of Gray Hole AODV network and the network without the Gray Hole Node. We then noticed that the data loss percentage of Gray Hole AODV will be increased as compared to the normal AODV network scenarios simulations. It is also understood by the analysis of the network that the packet loss already exist in the network which is because of density of data traffic because of which at the node interface queue the packets are dropped. Node and packet parameters are altered to minimize the data traffic. To evaluate the Gray Hole effect in the network, the packet loss has to be minimized which will happen in the network. In wireless ad-hoc network which is not having any Gray Hole, data packets will get lost due to dense data traffic in FTP traffic for instance. In our simulations of AODV network without gray hole node, we see that data loss has increased up to 35% to 40% when the parameters are changed. Therefore, the data loss does not always mean that there was a Gray Hole Node in the network. So it is not easy to detect the Gray Hole Node in the network.

Conclusion

From this project, we learnt the basic operation of NS-2 network simulator. Got to learn about the Mobile Ad-hoc Network(MANET); how it works and also had a detailed study of different routing protocols. We have simulated AODV and ZRP protocol using NS-2 and have compared the AODV routing protocol with ZRP for some parameters like PDR, End-to-end delay and throughput using X-Graph. Then we have studied the Black hole attack in MANET; have implemented it using AODV routing protocol in NS-2 and have also proposed a solution for preventing it using Sequence Number Comparison method with a simulation in NS-2. With that we have also compared the parameters like Delay, PDR and Throughput for normal AODV, AODV under attack and AODV with prevention of Blackhole using X-Graph.

REFERENCE

- Progress Report 1,2,3,4 and the midterm report for Mobile Computing Fall-2016. Cleveland State University.
- Vipin Khandelwal and Dinesh Goyal. BlackHole Attack and Detection Method for AODV Routing Protocol in MANETs. In International Journal of Advanced Research in Computer Engineering & Technology (IJARCET); Volume 2, Issue 4, April 2013
- S.S.Dhenakaran and A.Parvathavarthini. An Overview of Routing Protocols in Mobile Ad-Hoc Network.

  International Journal of Advanced Research in Computer Science and Software Engineering; Volume 3, Issue
  2, February 2013
- Neha Jain and Yogesh Chaba. Simulation based Performance Analysis of Zone Routing Protocol in Manet.

  International Journal of Computer Applications; Volume 88 – No.4, February 2014

  www.wikipedia.org
- www.ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-4-525-529.pdf www.ijcst.org/Volume3/Issue7/p10_3_7.pdf www.slogix.in
- www.ijcsits.org/papers/Vol2no32012/27vol2no3.pdf
- www.ripublication.com/irph/ijict_spl/ijictv4n4spl_07.pdf
- www.arxiv.org/ftp/arxiv/papers/0909/0909.2371.pdf
- www.youtube.com/watch?v=8B_of6FK9Lc