Intrusion Detection Scheme to Identify Wormhole Attack in Ad-Hoc Networks

BY

Avani Dadhania

11MCEC06

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May 2013

Intrusion Detection Scheme to Identify Wormhole Attack in Ad-Hoc Networks

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

BY

Avani Dadhania

11MCEC06

GUIDED BY

Prof. Sharada Valiveti

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May-2013

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in Computer Science and Engineering at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgement has been made in the text to all other material used.

Avani Dadhania

Certificate

This is to certify that the Major Project entitled ”Intrusion Detection Scheme to Identify Wormhole Attack in Ad-Hoc Networks” submitted by Avani Dadhania (11MCEC06), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Sharada Valiveti	Prof. Vijay Ukani
Guide, Associate Prof.	Associate Prof. and PGCSE-Coordinator,
Department of C.S.E.,	Department of C.S.E,
Institute of Technology,	Institute of Technology,
Nirma University, Ahmedabad.	Nirma University, Ahmedabad.

Dr. Sanjay Garg	Dr. Ketan Kotecha
Professor and Head,	Director,
Department of C.S.E.,	Institute of Technology,
Institute of Technology,	Nirma University, Ahmedabad.
Nirma University, Ahmedabad.	

Abstract

Security techniques have been designed to obtain certain objectives. All networked devices try to manage the network resources themselves which is termed as Availability. Availability ensures that the legitimate users should not be deprived of the basic network resources. Intrusion detection system is required to monitor the network, detect misbehavior or anomalies and notify other nodes in the network to avoid or detect misbehaving nodes. Due to dynamic infrastructure-less nature and lack of centralized monitoring points, the ad hoc networks are vulnerable to attacks. The network performance and reliability is break by attacks on ad hoc network routing protocols. AODV is a important on-demand reactive routing protocol for mobile ad hoc networks. The ultimate goal of the security solution for wireless network is to provide security services such as authentication, confidentiality, integrity, anonymity and availability to mobile users. This report provides a survey on wormhole attack and its counter measures in ad-hoc wireless network. Also we propose a method to detect Wormhole attack. Wormhole attack and TFSM based Intrusion Detection System is implemented using NS-2 simulator.

Acknowledgements

My deepest thanks to **Prof. Sharada Valiveti**, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad the Guide of the project that I undertook for giving her valuable inputs and correcting various documents of mine with attention and care. She has taken the pain to go through the project and make necessary amendments as and when needed.

My deep sense of gratitude to **Prof. Vijay Ukani**, PGCSE-Coordinator of Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for an exceptional support and continual encouragement throughout part one of the Major project.

I would like to thanks **Dr. Sanjay Garg**, Hon'ble Head of Department, Institute of Technology, Nirma University, Ahmedabad for his unmentionable support, providing basic infrastructure and healthy research environment.

I would like to thanks **Dr. Ketan Kotecha,** Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for his unmentionable support, providing basic infrastructure and healthy research environment.

I would also thank my Institution, all my faculty members in Department of Computer Science, my husband and my colleagues without whom this project would have been a distant reality. Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom I am able to complete the first part of my dissertation work successfully.

- Avani Dadhania
11MCEC06

Abbreviations

AODV	Ad hoc On-Demand Distance Vector Routing Protocol
IDS	Intrusion Detection System
RREQ	Route Request
RREP	Route Reply
RRER	Route Error
MANETs	Mobile Ad-hoc Networks
IDPS	Intrusion Detection and Prevention System
DoS	Denial of Service
HIDS	Host-based Intrusion Detection System
NIDS	Network-based Intrusion Detection System
TFSM	Time finite state machine
OLSR	Optimized Link State Routing Protocol
WRP	Wireless Routing Protocol
TORA	Temporally Ordered Routing Algorithm
DSDV	Destination-Sequenced Distance Vector Routing Protocol
CGSR	Cluster Gateway Switch Routing protocol
NS	Network Simulator
PDR	Packet Delivery Ratio
CBR	Constant Bit Rate

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A network is a collection of mobile nodes that establish a communication protocol dynamically. The nodes may join the network at any time and communicate with entire network via neighboring nodes. Each member of such a network is responsible for accurate routing of information. Due to arbitrary physical configuration of an ad hoc network, there is no central decision-making mechanism of any kind rather, the network employs distributed mechanisms of coordination and management [1]. Wireless ad-hoc networks do not rely on a preexisting network infrastructure, and are characterized by wireless multi-hop communication. Unlike fixed wired networks, wireless ad-hoc networks have many operational limitations. For example, the wireless link is constrained by transmission range and bandwidth, and the mobile nodes may be constrained by battery life, CPU, and memory. Wireless ad-hoc networks are used in situations where a network must be deployed rapidly, without an existing infrastructure [2].

This thesis contains a general overview of wireless adhoc network technology, Survey of vulnerabilities in ad hoc networks, attacks possible on ad hoc networks, intrusion detection system and the research achievements in the field of IDS . Active routing attack against AODV protocol like Wormhole attack is implemented for analyzing the effect of attacks on performance of ad hoc network and are used for detection system .

TFSM Timed Finite State Machine based intrusion detection system is developed for detecting attacks. The NS-2.34 simulator was used for simulation.The results were then analyzed based on the suggested evaluation metrics in order to evaluate damage due to attack and to verify effectiveness of detection system.

1.1 Objective of the Work

The main objective of the work is to design a Timed Finite State Machine (TFSM) based intrusion detection approach for AODV protocol which is basically a knowledge based approach that defines normal behavior of the protected networks to detect active routing attacks with less false alarm rate and take steps to minimize the effect of attack on ad hoc network.

- To implement and analyze effect of availability attack on Ad hoc networks and implement IDS.
- To study existing intrusion detection system for AODV based ad hoc network.
- To propose TFSM based intrusion detection system.

1.2 Scope of the Work

- The scope of this work is to decrease false alarm rate and to improve the detection efficiency of the system and improving the detection system for different protocols.

1.3 Motivation of the Work

- The mobility of wireless devices demands more resilient, stronger and effective security schemes. Intrusion detection, which is an indispensable part of a security system, presents also a particular challenge due to the dynamic nature of ad hoc networks, the lack of central points, and their highly constrained nodes.

Chapter 2

Literature Survey

An ad hoc network is an infrastructure-less network where each node acts as a router for establishing connection between source and destination. As there is no centralized administration for controlling the network, every node participating in the network is responsible for the reliable operation of the whole network. Due to node mobility, network topology changes frequently. Under all above conditions, it is important to manage routing information efficiently. Ad hoc networks work on the basis of the cooperation between nodes. To make this procedure feasible, trust between nodes is necessary. However, most ad hoc routing protocols do not take security threats into account, and therefore ad hoc networks are inherently vulnerable to them. This report contains detailed survey of characteristics of ad hoc network, how they pose challenges in ad hoc network security, attacks in ad hoc network and brief description of some existing intrusion detection system.

Ad hoc routing protocols can be classified as either proactive or reactive, depending on the method used to discover and maintain routes. Proactive routing protocols discover and maintain a complete set of routes for the lifetime of the network. In contrast, reactive routing protocols only find routes when needed, and maintain those routes for the duration of communication.

2.1 Routing in Ad-Hoc Networks

Ad-hoc networks have special limitation such as limited bandwidth and power and properties like highly dynamic topology, high error rates etc. Compared to wired networks, in an ad-hoc network, all nodes are mobile and are connected dynamically in an arbitrary manner. Nodes of ad-hoc network behave as router and take part in discovery and maintenance to establish a reliable route. Therefore, routing protocols for wired networks cannot be directly used in wireless networks.Routing protocols for ad-hoc network are divided into two categories based on management of routing tables [3]. These categories are Table Driven Routing Protocols and On-Demand Routing Protocols.

2.1.1 Table Driven Routing Protocols

These types of protocols are called table driven protocols in which, the route to all the nodes is maintained in routing table. Packets are transferred over the predefined route specified in the routing table. In this scheme, the packet forwarding is done faster but the routing overhead is greater because all the routes have to be defined before transferring the packets. Proactive protocols have lower latency because all the routes are maintained at all the times. Example protocols: DSDV (Destination-Sequenced Distance Vector Routing Protocol), OLSR (Optimized Link State Routing), Wireless routing protocol (WRP), Fish eye State Routing protocol (FSR), Cluster Gateway Switch Routing protocol (CGSR), Topology Dissemination Based on Reverse Path Forwarding.

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 like

- Periodically updating of the network topology increases bandwidth overhead
- Periodically updating of 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

2.1.2 On-Demand Routing Protocols

In this category of protocol there is an initialization of a route discovery mechanism by the source node to find the route to the destination node when the source node has data packets to send. When a route is found, the route maintenance is initiated to maintain this route until it is no longer required or the destination is not reachable. The advantage of these protocols is that overhead messaging is reduced. One of the drawbacks of these protocols is the delay in discovering a new route.

On-Demand Routing Protocols creates route tables 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 happens until the destination is found. Afterwards, the destination node sends a reply packet to the source node through 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) are On-Demand Routing protocols.

2.2 Ad-Hoc On-Demand Distance Vector Routing Protocol

The Ad hoc On-demand Distance Vector (AODV) protocol creates routes on demand, trying to minimize the number of control messages. It is assumed that nodes which are not in the selected path does not maintain routing information or exchange routing table information, and that the process is source initiated.

The path discovery process starts when a source node desires to send a message to a destination node and does not have a valid route. The source node broadcasts a route request packet (RREQ) to its neighbor nodes, which then forward the request to their neighbor nodes, and so on. The process continues until either the destination node, or an intermediate node with an updated (fresh enough) route to the destination, is reached by this request. Then, the node responds with a route reply packet (RREP) back to the neighbor from which it first received the RREQ. The AODV protocol only supports symmetric links. The reply packets are routed back along the reverse path established by the request packets. The reply packets that travel along the intermediate nodes setup forwarding entries in the routing tables. These table entries point to the node from which the RREP was received.

There is a timer associated with each route entry. The entries expire if not used by data packets. Destination sequence numbers are used by AODV to ensure loop free routes and up to date routing information. With the mobility and radio interferences, links in the network can go down and a route repair procedure may be necessary. If a node moves out of the radio range of its neighbor, the upstream neighbor propagates a link failure notification (routing error packet - RERR) to each of its upstream neighbors to inform the failure of part of the route. The failure notification is propagated until the source node is reached. When the source node is reached by the routing error packet it initiates a new path discovery process. Connectivity information can be obtained using hello messages. Hello messages are routing reply packets which are periodically broadcasted by a node to inform its existence to its neighbors.

Hello messages may be used to detect and monitor links to neighbors. If Hello messages are used, each active node periodically broadcasts a Hello message that all its neighbors receive. Because nodes periodically send Hello messages, if a node fails to receive several Hello messages from a neighbor, a link break is detected.

When a source has data to transmit to an unknown destination, it broadcasts a Route Request (RREQ) for that destination. At each intermediate node, when a RREQ is received a route to the source is created. If the receiving node has not received this RREQ before, is not the destination and does not have a current route to the destination, it rebroadcasts the RREQ. If the receiving node is the destination or has a current route to the destination, it generates a Route Reply (RREP). The RREP is unicast in a hop-by- hop fashion to the source. As the RREP propagates, each intermediate node creates a route to the destination. When the source receives the RREP, it records the route to the destination and can begin sending data. If multiple RREPs are received by the source, the route with the shortest hop count is chosen.

As data flows from the source to the destination, each node along the route updates the timers associated with the routes to the source and destination, maintaining the routes in the routing table. If a route is not used for some period of time, a node cannot be sure whether the route is still valid; consequently, the node removes the route from its routing table.

If data is flowing and a link break is detected, a Route Error (RERR) is sent to the source of the data in a hop-by-hop fashion. As the RERR propagates towards the source, each intermediate node invalidates routes to any unreachable destinations. When the source of the data receives the RERR, it invalidates the route and reinitiates route discovery if necessary.

2.3 Vulnerabilities of Ad Hoc Networks

Ad hoc networks have characteristics such as dynamically changing topology, weak physical protection of nodes, the absence of centralized administration, and high dependence on inherent node cooperation. Due to dynamic topology, ad hoc networks do not have a well-defined boundary, and thus, mechanisms such as were walls are not applicable. Vulnerabilities in ad hoc network described in are:

- Nodes of mobile ad hoc networks have limited ranges and because of that it requires multi hop communication. Ad hoc network runs on an assumption that once the node has promised to transmit the packet, it will not cheat but this does not holds true when nodes in the networks have contradictory goals. Due to this, neighbors of intermediate nodes can use the reputation of intermediate nodes to transmission.
- Node mobility leads to frequent change in network topology
- Use of wireless links into network increases the risk of link attacks
- Relatively poor protection
- Long life of network requires distributed architecture
- Risk of Denial of Service (DoS) attacks due to lack of infrastructure and chances of link breakage and channel errors due to mobility
- Need of scalability
- Nodes in Ad Hoc Networks have limited services and security provision due to limited memory and computational power

2.4 Types of Attacks

In intrusion detection, one needs to analyze anomalies due to both the consequence and technique of an attack. Consequence gives evidence about the success of attack and technique helps in identifying attack and some time attacker too.

Attacks in ad hoc network can be categorized according to their consequences into passive attack which does not involve disruption of information but they are merely intended to steal information and to eavesdrop on the communication within the network vs. active attack in which data are altered by attacker which involves overloading of network or preventing nodes from using the networks services effectively. The Passive attacks typically involve only eavesdropping of data, whereas the active attacks involve actions performed by adversaries such as replication, modification and deletion of exchanged data. In particular, attacks in MANET can cause congestion, propagate incorrect routing information, prevent services from working properly or shutdown them completely. Nodes that perform the active attacks are considered to be malicious, and referred to as compromised, while nodes that just drop the packets they receive with the aim of saving battery life are considered to be selfish anymore [5].

Internal attack which comes from compromised node inside the network vs. external attack in which an unauthenticated attackers can replay old routing information or inject false routing information to partition the network or increase the network load. Another category is of the attacks that affect the routing of traffic in network. These types of attacks are classified into following subcategories:

- Denial-of-Service: A node is prevented from receiving and sending data packets to its destinations [6].
- Fabricated route Attacks: Fabrication refers to attacks performed by generating false routing messages. E.g. launching attack by sending false route error

message. On receiving the route error messages, the nodes using that route will delete the route table entry for that destination node.

- Resource Consumption Attacks: Malicious node prevents other nodes from getting fair share of bandwidth by flooding RREQ for random addresses or two malicious nodes send large volume of data between themselves, thereby depleting the available network bandwidth. An attacker tries to consume or waste away resources of other nodes present in the network [7].

- Packets Dropping: Either malicious node first advertise correct path to the destination and later drops data packets or malicious node drops control packets sent by another node but behaves properly when it itself wants to send data.

- Selfishness: In these attacks, a malicious node behaves selfishly, using the network for its own needs, without participating in the overall routing process. Selfish nodes use the network but do not cooperate, saving battery life for their own communications.

- Wormhole Attacks: In this attack, an adversary receives packets at one point in the network, tunnels them to another point in the network, and then replays them into the network from that point. This tunnel between two adversaries is called wormhole [22].

Here we described the different attacks possible in different Layers shown in Table.

Layer	Attacks
Application Layer	Repudiation, Syn Flooding,Viruses
Transport Layer	TCP/UDP SYN Flood
Network Layer	Blackhole, Wormhole, Byzantine, Information Disclosure [8]
Data Link Layer	Monitoring, Traffic Analysis
Physical layer	Eavesdropping, Active Interference [9]

Table I: Summary Of Protocol Layers And Specific Attacks

2.5 Wormhole Attack Survey

In wormhole attack, a malicious node receives packets at one location in the network and tunnels them to another location in the network, where these packets are resent into the network. This tunnel between two colluding attackers is referred to as a wormhole. It could be established through wired link between two colluding attackers or through a single long-range wireless link. In this form of attack the attacker may create a wormhole even for packets not addressed to itself because of broadcast nature of the radio channel [10].

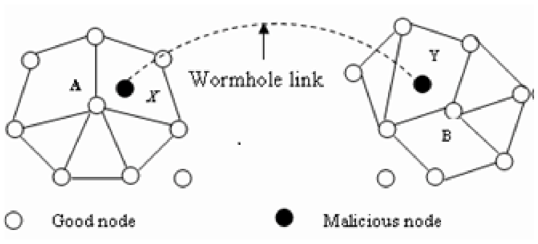


Figure 2.1: A Network Under an active attack [22]

The wormhole attack is particularly dangerous for many ad hoc network routing protocols in which the nodes that hear a packet transmission directly from some node consider themselves to be in range of (and thus a neighbor of) that node. For example, when used against an on-demand routing protocols such as AODV , a powerful application of the wormhole attack can be mounted by tunneling each route request packet directly to the destination target node of the request. When the destination nodes neighbors hear this request packet, they will follow normal routing protocol processing to rebroadcast that copy of the request and then discard without processing all other received route request packets originating from this same route discovery. This attack thus prevents any routes other than through the wormhole from being discovered, and if the attacker is near the initiator of the route discovery. This attack can even prevent routes more than two hops long from being discovered. Possible ways for the attacker to then exploit the wormhole include discarding rather than forwarding all data packets, thereby creating a permanent Denial-of-Service attack or selectively discarding or modifying certain data packets. So, if proper mechanisms are not employed to protect the network from wormhole attacks, most of the existing routing protocols for ad hoc wireless networks may fail to find valid routes [10].

Table II: Summary Of Detection Methods Of Wormhole Attack

Method	Mobility	QoS Parameter	Synchronization	False detection
Deworm[11]	Location of nodes not required	Low overhead	Time synchronization not considered	Low false positive rate
TTM[12]	Not considered	Low overhead	Transmission time considered, background traffic	Low false positive rate and low false negative rate.
Trust model Based[30]	Trust mode considered	Less power consumed	Clock synchronization not required	Not Handled
Delphi[14]	Not considered	Delay	Not required	Not Handled
Wormeros[15]	Topological change is not considered	Not considered	Time synchronization not required.RTT between source node and destination node is considered	Both false positive and false negative alarms are considered
Hop Count Based[16]	Out Of Band Channel	Packet Delivery Ratio, Throughput, Overhead,Delay	Not required	Not Handled
Whop[17]	Directional Antenna not required	Not considered	Clock synchronization not required	Not Handled

2.6 Classification of Intrusion Detection System (IDS)

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents. Intrusion detection and prevention systems (IDPS)are primarily focused on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

IDPSs typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDPSs can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attacks content [32].

2.6.1 Based on data collection mechanisms

- Network Based (NIDS): Network-based IDS runs on a gateway of a network and captures and examines the network traffic that flows through it. Obviously this approach is not suitable for ad hoc networks since there is no central point that allows monitoring of the whole network. The NIDS are broader in scope, are able to detect attack from outside, examine packet header and entire packet. The problem with NIDS is that it has high false positive rate. A network-based IDPS monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity. This section provides a detailed discussion of network-based IDPS technologies.

- Host Based(HIDS): A HIDS relies on capturing local network traffic to the specific host. This data is analyzed and processed locally to the host and is used either to secure the activities of this host, or to notify another participating node for the malicious action of the node that performs the attack. It is better for detecting attack from inside but it responds after suspicious log entry.

2.6.2 Based on detection mechanisms

- Signature Based IDS: A knowledge-based (Signature-based) Intrusion Detection Systems (IDS) references a database of previous attack signatures and known system vulnerabilities. The meaning of word signature, when we talk about Intrusion Detection Systems (IDS) is recorded evidence of an intrusion or attack. Each intrusion leaves a footprint behind (e.g., nature of data packets, failed attempt to run an application, failed logins, file and folder access etc.). These footprints are called signatures and can be used to identify and prevent the same attacks in the future. Based on these signatures Knowledge-based (Signature-based) IDS identify intrusion attempts.

- Anomaly-Based IDS: A Behavior-based (Anomaly-based) Intrusion Detection Systems (IDS) references a baseline or learned pattern of normal system activity to identify active intrusion attempts. Deviations from this baseline or pattern cause an alarm to be triggered.

2.7 Architecture of IDS

Intrusion detection system is software that detects attacks on a network or computer system. Intrusion Detection Systems are normally categorized into misuse detection and anomaly detection. In misuse detection systems, previous attack patterns are stored in a database. Any data similar to this data is classified as attack. Anomaly detection refers to statistical knowledge about normal activity. Intrusions correspond to deviations from the normal activity of system. The anomaly detection system has high false positive/ negative alarm rate compared to misuse detection systems. However, it is more effective in detecting new attacks or deviation from the nominal usage. Furthermore, the IDS is classified based on the data source: Network IDS (NIDS) and Host-based IDS (HIDS) systems. The NIDS analyzes the data through the network. The HIDS check for attacks or intrusions in the single host only. It does this by analyzing the audit logs in the system. Current intrusion detection systems are futile to cope with new, elegant and structured attacks, due to several practical and theoretical limitations. These limitations have lead many researchers to apply different machine learning approaches for detecting anomalies [18].

2.7.1 Detection Capabilities

Network-based IDPSs typically offer extensive and broad detection capabilities. Most products use a combination of signature-based detection, anomaly-based detection, and stateful protocol analysis techniques to perform in-depth analysis of common protocols, organizations should use network-based IDPS products that use such a combination of techniques. The detection methods are usually tightly interwoven; for example, a stateful protocol analysis engine might parse activity into requests and responses, each of which is examined for anomalies and compared to signatures of known bad activity.

The following are the detection capabilities:

- Types of events detected
- Detection accuracy
- Tuning and customization
- Technology limitations

2.7.2 Types Of IDS Architecture

There are four main architectures on the network.

a. Standalone IDS

In the standalone architecture, the IDS runs on each node to determine intrusions independently. There is no cooperation and no data exchanged among the IDSes on the network. This architecture is more suitable for flat network infrastructure than for multilayered network infrastructure [19].

In this architecture, one IDS is executed independently for each node, and the necessary decision taken for that node is based on the data collected, because there is no interaction among network nodes and therefore no data is interchanged. In addition, each node has no knowledge of the position of other nodes in that network and no alert information crosses the network. Even though, due to its limitations, they are not effective, but they can be suitable for networks where nodes are not capable of executing an IDS or where an IDS has been installed.

b. Distributed and Collaborative IDS

The distributed and collaborative architecture has a rule that every node in the Ad-Hoc Network must participate in intrusion detection and response by having an IDS agent running on them. The IDS agent is responsible for detecting and collecting local events and data to identify possible intrusions, as well as initiating a response independently [20].

c. Hierarchical IDS

The hierarchical architecture is an extended version of the distributed and collaborative IDS architecture. This architecture proposes using multi-layered network infrastructures where the network is divided into clusters. The architecture has cluster heads, in some sense, act as control points which are similar to switches, routers, or gateways in wired networks [21].

Hierarchical IDS architecture is the well developed distributed and cooperative IDS architecture and has been presented for multi-layered network infrastructure in such a way that network is divided into clusters. The cluster-heads of each cluster has more responsibilities compared to other members, For example, sending routing packets between clusters. In this way, these cluster-heads, behave just like control points, for example switches, routers or gateways, in wired networks. The name multi-layer IDS is also used for hierarchical IDS architecture.

d. Mobile Agent for IDS

The mobile agent for IDS architecture uses mobile agents to perform specific task on a nodes behalf the owner of the agents. This architecture allows the distribution of the intrusion detection tasks. Mobile agents have been deployed in many techniques for IDSs in MANETs. Due to its ability of moving in network, each mobile agent is considered for performing just one special task and then one or more mobile agents are distributed amongst network nodes. This operation allows the distributed intrusion detection in the system. There are advantages for using mobile agents. Some responsibilities are not delegated to every node, and so it helps in reducing the energy consumption, which is also an important factor in MANET network. It also provides for fault tolerance in such a way that if the network is segmented or some of the agents break down, they can still continue to function. Individual IDS agents are placed on each and every node. Each the IDS agent runs independently and monitors local activities (user and systems activities, and communication activities within the radio range). The agent detects intrusion from local traces and initiates response. If anomaly is detected in the local data, or if the evidence is inconclusive and a broader search is warranted, neighboring IDS agents will cooperatively participate in global intrusion detection actions. These individual IDS agents collectively form the IDS system to defend the wireless ad-hoc network [22].

2.8 Intrusion Detection Systems for MANET

Since the IDS for traditional wired systems are not well-suited to MANETs, many researchers have proposed several IDS especially for MANETs, which some of them will be reviewed in this section.

2.8.1 Distributed and Cooperative IDS

As described in Section, Zhang and Lee also proposed the model for a distributed and cooperative IDS as shown in Figure 2.2. The model for an IDS agent is structured into six modules. The local data collection module collects real-time audit data, which includes system and user activities within its radio range. This collected data will be analyzed by the local detection engine module for evidence of anomalies. If an anomaly is detected with strong evidence, the IDS agent can determine independently that the system is under attack and initiate a response through the local response module (i.e., alerting the local user) or the global response module (i.e., deciding on an action), depending on the type of intrusion, the type of network protocols and applications, and the certainty of the evidence. If an anomaly is detected with weak or inconclusive evidence, the IDS agent can request the cooperation of neighboring IDS agents through a cooperative detection engine module, which communicates to other agents through a secure communication module.

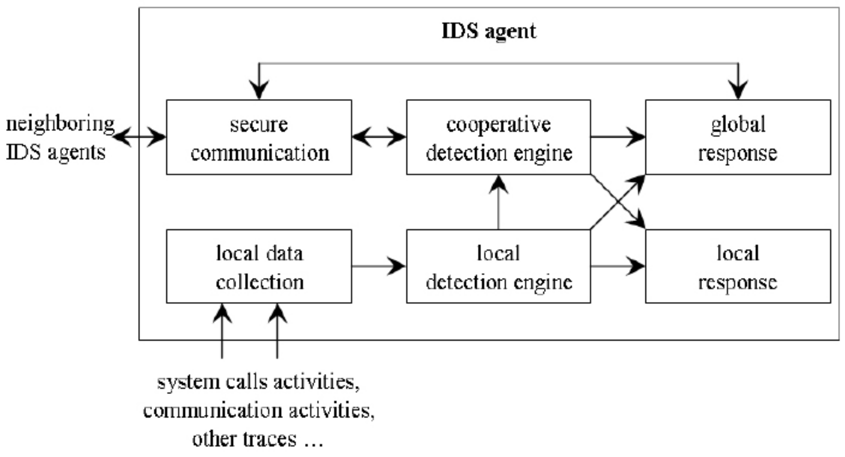


Figure 2.2: A sample IDS System[23]

2.8.2 Local Intrusion Detection System (LIDS)

LIDS is implemented on every node for local concern, which can be extended for global concern by cooperating with other LIDS. Two types of data are exchanged among LIDS: security data (to obtain complementary information from collaborating nodes) and intrusion alerts (to inform others of locally detected intrusion). In order to analyze the possible intrusion, data must be obtained from what the LIDS detects on, along with additional information from other nodes. Other LIDS might be run on different operating systems or use data from different activities such as system, application, or network activities. therefore, the format of this raw data might be different, which makes it hard for LIDS to analyze. However, such difficulties can be solved by using Simple Network Management Protocol (SNMP) data located in Management Information Base (MIBs) as an audit data source. Such a data source not only eliminates those difficulties, but also reduces the increase in using additional resources to collect audit data if an SNMP agent is already run on each node. For the methodology of detection, Local IDS Agent can use either anomaly or misuse detection. However, the combination of two mechanisms will offer the better model [24].

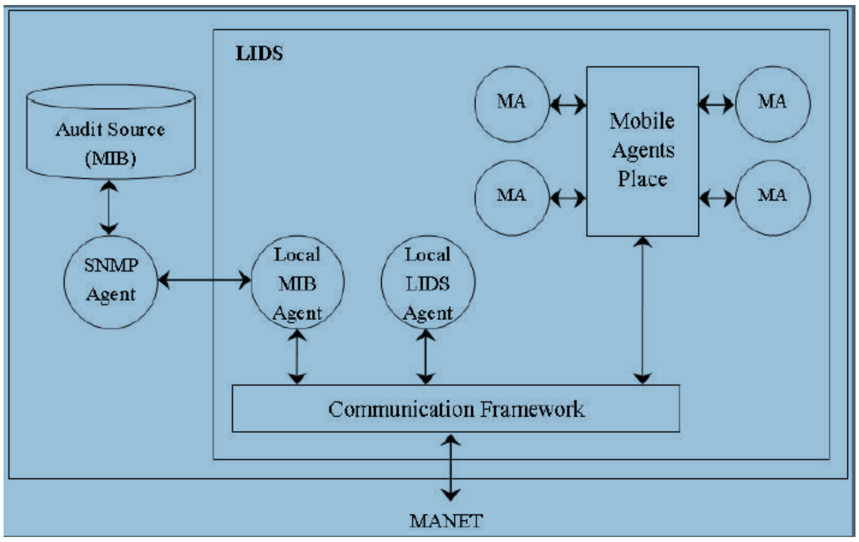


Figure 2.3: Local Intrusion Detection System[23]

Once the local intrusion is detected, the LIDS initiates a response and informs the other nodes in the network. Many intrusion detection systems have been proposed in traditional wired networks, where all trac must go through switches, routers, or gateways. Hence, IDS can be added to and implemented in these devices easily. On the other hand, MANETs do not have such devices. Moreover, the medium is wide open, so both legitimate and malicious users can access it. Furthermore, there is no clear separation between normal and unusual activities in a mobile environment. The network infrastructures that MANETs can be configured to are either at or multi-layer, depending on the applications. Therefore, the optimal IDS architecture for a MANET may depend on the network infrastructure itself [25]. In a network infrastructure, all nodes are considered equal, thus it may be suitable for applications such as virtual classrooms or conferences. Intrusion detection system serves as an alarm mechanism for a computer system. It detects the security comprises happened to a computer system and then issues an alarm message to an entity, such as a site security officer so that the entity can take some actions against the intrusion (Axelsson, 2000;Greg, 2004). In the discussion of IDS in MANET, two concepts need to be distinguished: intrusion detection techniques and intrusion detection architecture. Intrusion detection techniques refer to the concepts such as anomaly and misuse detection. They mainly solve the problems how an IDS detects an intrusion with a certain algorithm, given some audit data as input data. It can be viewed as an algorithm. The intrusion detection architecture, however, deals with problems in a larger scope. The intrusion detection technique is basically independent from the architecture or environment. In other words, anomaly and misuse detection can be utilized in wireless environment just as they are in wired network. The difference in implementation is mainly on what audit data to take as input to the algorithm. However, most IDS in MANET utilize anomaly detection because of the special nature of MANET. Intrusion detection may sometimes produce false alarms, for example as a result of malfunctioning network interface or sending attack description or signatures via email [31].

Chapter 3

Study of NS-2 Simulator

After detailed study of different existing simulators, NS-2 was selected as simulator for implementation.

3.1 Network Simulator-2

For making simulation trustworthy, simulation process should be repeatable so that it can be used for further reviews, unbiased means should be used to variety of scenarios, rigorous must truly test the aspects of environment being studied and statistically sound. After detailed study of different existing simulators NS-2 was selected as simulator for implementation. NS-2 is an event driven packet level network simulator developed as part of the VINT project targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks [27].

NS began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is supported through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. NS has always included substantal contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems.

3.1.1 Installation Steps for NS-2.34

Before start your NS-2 installation make sure that you logged in as administrator. For ensuring that you are the administrator type su and give your password given by during installation of FEDORA 10 Terminal window can be opened from the menu Applications and then from sub-menu System Tools, you can click Terminal from that list. A terminal window is opened. Then type the following command one by one for installing these packages(when you type the first command you will see a file download and then you should install these packages by clicking y)

```
yum install autoconf
yum install automake
yum install gcc-c++
yum install libX11-devel
yum install xorg-x11-proto-devel
yum install libXt-devel
```

And for nam installation (optional):

```
yum install libXmu-devel
yum install make *.*
```

after completing this step proceed further as follows. Steps to Install ns2.34 in Fedora 10

Step 1: Download ns-allinone-2.34.tar.gz

Step 2: Execute the following commands to install ns2.34 from the terminal. You must login as root.

```
tar -xzf ns-allinone-2.34.tar.gz
cd ns-allinone-2.34
./install
```

Step 3: After a long wait and a whole lot of text, you should see the installation finish up with text like the following: Please put /opt/ns-allinone-2.34/bin:/opt/ns-allinone-2.34/tcl8.4.18/unix:/opt/ns allinone-2.34/tk8.4.18/unix into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.
IMPORTANT NOTICES:
After these steps,you can now run the ns validation suite with cd ns-2.34,
./validate
For trouble shooting, please first read ns problems page
Step 4: To change PATH environment and others execute following commands in terminal.
gedit /.bashrc
Step 5: Add required lines at the end of bashrc file.
Step 6: To make the change in the path and others permanent, type the following command in terminal.
source /.bashrc
You can check what is written in your path by writing echo PATH
Step 7: Now, the installation has been completed. If you try:ns
Step 8: You can test/validate the installation by doing the following: [It is not necessary] cd /opt/ns-allinone-2.34/ns-2.34/ ./validate Note that this validation takes a really long time. If it starts out ok, you probably have a good installation.
Step 9: Simple ns2 Simulation program
cd /opt/ns-allinone-2.34/ns-2.34/tcl/ex ns wireless-demo.tcl
nam demo.nam If output comes, then everything is ok
Testing: Way 1: ns Way 2:
Step 1: Create a file named sample.tcl and add the following lines in the file. set ns [new Simulator] ns at 1 ”puts Ḧello World!̈” ns at 1.5 ”exit” ns run
Step 2: Run the file by executing following command ns sample.tcl
Step 3: Output will be: Hello World!

3.1.2 Network Animation (NAM)Trace

Nam is a Tcl/Tk based animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various data inspection tools. It supports topology layout, packet level animation, and various data inspection tools. Nam began at LBL. It has evolved substantially over the past few years. The nam development effort was an ongoing collaboration with the VINT project. Currently, it is being developed as an open source project hosted at Sourceforge.

3.2 Programming Languages

Tcl is fairly simple and if you already have some programming skills, you should be able to learn most of what you need for simple scenarios as you go along

3.2.1 Tcl/Otcl Programming

NS2 uses OTcl to create and configure a network, and uses C++ to run simulation. All C++ codes need to be compiled and linked to create an executable file. Since the body of NS2 is fairly large, the compilation time is not negligible. A typical Pentium 4 computer requires few seconds (long enough to annoy most programmers) to compile and link the codes with a small change such as including int i=0; into the codes.

OTcl, on the other hand, is an interpreter, not a compiler. Any change in an OTcl file does not need compilation. Nevertheless, since OTcl does not convert all the codes into machine language, each line needs more execution time. In summary, C++ is fast to run but slow to change. It is suitable for running a large simulation. OTcl, on the other hand, is slow to run but fast to change. It is therefore suitable to run a small simulation over several repetitions (each may have different parameters).

NS2 is constructed by combining the advantages of these two languages. NS2 manual provides the following guidelines to choose a coding language: Use OTcl for configuration, setup, or one time simulation, or to run simulation with existing NS2 modules. This option is preferable for most beginners, since it does not involve complicated internal mechanism of NS2. Unfortunately, existing NS2 modules are fairly limited. This option is perhaps not sufficient for most researchers. Use C++ when you are dealing with a packet, or when you need to modify existing NS2 modules.

Tcl is a general purpose scripting language. While it can do anything other languages could possibly do, its integration with other languages has proven even more powerful. Tcl runs on most of the platforms such as Unix,Windows, and Mac. The strength of Tcl is its simplicity. It is not necessary to declare a data type for variable prior to the usage. At runtime, Tcl interprets the code line by line and converts the string into appropriate data type (e.g., integer) on the fly.

OTcl is an object-oriented version of Tcl, just like C++ is an object-oriented version of C. The basic architecture and syntax in OTcl are much the same as those in Tcl. The difference, however, is the philosophy behind each of them. In OTcl, the concepts of classes and objects are of great importance. A class is a representation of a group of objects which share the same behavior(s) or trait(s). Such a behavior can be passed down to child classes. In this respect, the donor and the receiver of the behaviors are called a superclass (or a parent class) and a subclass (or a child class), respectively. Apart from inheriting behaviors from a parent class, a class defines its own functionalities to make itself more specific. This inheritance is the very main concept for any OOP including OTcl.

3.3 Awk Script

AWK is a general-purpose programming language designed for processing of text files. AWK refers to each line in a file as a record. Each record consists of fields, each of which is separated by one or more spaces or tabs. Generally, AWK reads data from a file consisting of fields of records, processes those fields with certain arithmetic or string operations, and outputs the results to a file as a formatted report. An AWK program is a series of pattern action pairs, written as: "condition" "action" where condition is typically an expression and action is a series of commands. The input is split into records, where by default records are separated by newline characters so that the input is split into lines. The program tests each record against each of the conditions in turn, and executes the action for each expression that is true. Either the condition or the action may be omitted. The condition defaults to matching every record. The default action is to print the record [28].

To process an input file, AWK follows an instruction specified in an AWK script. An AWK script can be specified at the command prompt or in a file. While the strength of the former is the simplicity (in invocation), that of the later is the functionality. In the later, the programming functionalities such as variables, loops, and conditions can be included into an AWK script to perform desired actions. In what follows we give a brief introduction to the AWK language.

Chapter 4

Implementation Of Wormhole Attack

4.1 Introduction Of Wormhole Attack

Here, we present a brief overview of Wormhole Attack. During this attack, a malicious node captures packets from one location in the network and ”tunnels” them to another malicious node at a distant point which replays them locally. The tunnel can be established in many ways e.g.in-band and out-of-band channel.

In-band channel: Here malicious node m1 tunnels the received route request packet to another malicious node m2 using encapsulation even though there is one or more nodes between two malicious nodes, the nodes following m2 nodes believe that there is no node between m1 and m2 [26].

Out-of-band channel: Two malicious nodes m1 and m2 employ an physical channel between them by either dedicated wired link or long range wireless link [26].

4.2 Implementation Of Wormhole Attack

In the Wormhole attack two malicious node will make a tunnel. One malicious node forward a packet to second malicious node instead of their neighbors which are in the radio range and second malicious node will drop all the packets. So that both the malicious node will make tunnel and transfer packet to each other and dropping packets.

In this attack one malicious node will receive the request packet and reply it there without knowing path and without having path or destination detail. It is doing this because malicious node wants to participate in the route and want to drop the data packet. In receive request function of aodv.cc file there it directly reply the packet with highest sequence number to the source and the when source received Reply it start to send data packets and malicious node start to drop the data packets. The procedure for adding the malicious node as wormhole in AODV

Step 1: modify aodv.cc and aodv.h files. In aodv.h add following line
bool malicious;

Step 2: In aodv.cc add malicious = false;
With this variable define if the node is malicious or not.
The above code is needed to initialize, and all nodes are initially not malicious. Then add a code to catch which node is set as malicious. In aodv.cc variable hacker as string and pass its value malicious = true.

Step 3: Code in TCL to set a malicious node. Add following line to set node 5 as malicious node so define node 5 as hacker in the tcl file. Now define another variable as malicious node in the tcl file.
$\lfloor node\,(5)\; setragent \rfloor$ hacker

Step 4: change in the AODV:Forward function so that it directly forward the packet to next malicious node and that malicious node will start to drop the packet.

Step 5: Here Set malicious node but did not tell malicious node what to do. As it is known, rt resolve(Packet *p) function is used to select next hop node when routing data packets. So, we tell malicious node just drop any packet when it receives.

4.3 Simulation Parameter

The experiments were carried out using the network simulator (ns-2). The scenarios developed to carry out the tests use as parameters the mobility of the nodes and the number of active connections in the network. The module explained above was tested with the previously developed attacks. The choices of the simulator parameters that are presented in table 4.1 consider both the accuracy and the efficiency of the simulation.

Table I: Simulation parameters

Property	Value
Nodes	10,20,30,40,50
Simulation time	600Sec
Mobility model	Random way point
Coverage area	750m*750m
Maximum speed	20 m/s
Pause time	1.0 Sec
Traffic type	Constant Bit Rate(CBR/UDP)
Send Rate	10 packets/sec
Packet size	512 bytes

4.4 Performance Metrics

- Packet Delivery Ratio: The percentage of the number of packets that are received by destination to the number of packets sent by source. The larger this metric, the more efficient MANET will be.

  PDR= $\frac{\Sigma \text{PacketReceivedbyDestination}}{\Sigma \text{PacketReceivedbySource}} \times 100$

- Throughput: Throughput is the average rate of successful message delivery over a communication channel. It is defined as total number of delivered data packets divided by the total duration of simulation time. The throughput is usually measured in bits per second (bit/s or Kbps), and sometimes in data packets per second or data packets per time slot.

  Throughput[Kbps]= $\frac{\text{PacketReceivedsize}}{\text{Stoptime}-\text{Starttime}} \times 0.008$

- End-to-End Delay: It is defined as time taken for a packet to be transmitted across network from source to destination. The metric should have lower value for the efficient network.

  End-to-End Delay= $\frac{Sumofdelaysofeach CBRpacketreceived}{NumberofCBRpacketreceived}$

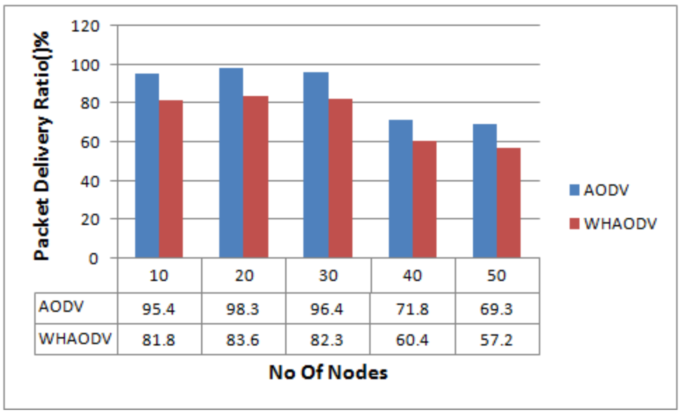
4.5 Evaluation Of Wormhole Attack

Two metrics that were used in the evaluation of the Wormhole attack detection system are the packet delivery ratio and the Throughput. All the metrics are plotted against: Number of active connections with mobility.

Table II shows the simulation results of packet delivery ratio and Table III shows simulation results for throughput for different nodes with respect to varying number of active connections and node mobility. Figure 4.1 shows the packet delivery ratio with respect to varying number of active connections and Figure 4.2 shows throughput with respect to varying number of active connections. Average packet delivery ratio of AODV with respect of number of active connections is 86.24 and with WHAODV is 73.6. Figure 4.1 shows that when number of nodes increase and when two node are wormhole, then PDR is decreased compared to AODV. Figure 4.2 shows that when number of nodes increase and when two node are wormhole, then throughput is decreased compared to AODV. Figure 4.3 shows that when number of nodes increase and when two node are wormhole, then End-to End Delay is increased compared to AODV.

Table II: Packet Delivery Ratio

No Of Nodes	AODV	WHAODV
10	95.4	81.8
20	98.3	83.6
30	96.4	82.3
40	71.8	60.4
50	69.3	57.2

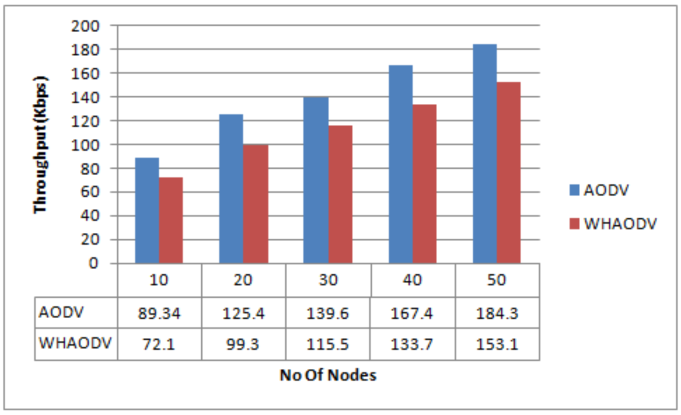


	10	20	30	40	50
AODV	95.4	98.3	96.4	71.8	69.3
WHAODV	81.8	83.6	82.3	60.4	57.2

Figure 4.1: Comparison of Packet Delivery Ratio in AODV and WHAODV

Table III: Throughput

No Of Nodes	AODV	WHAODV
10	89.34	72.1
20	125.4	99.3
30	139.6	115.5
40	167.4	133.7
50	184.3	153.1

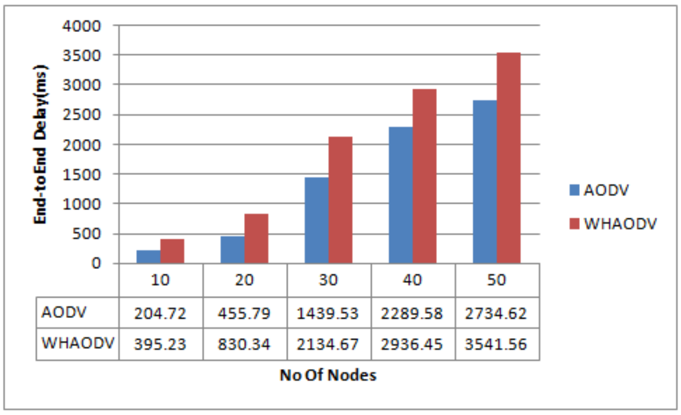


	10	20	30	40	50
AODV	89.34	125.4	139.6	167.4	184.3
WHAODV	72.1	99.3	115.5	133.7	153.1

Figure 4.2: Comparison Of Throughput in AODV and WHAODV

Table IV: End-to-End Delay

No Of Nodes	AODV	WHAODV
10	204.72	395.23
20	455.79	830.34
30	1439.53	2134.67
40	2289.58	2936.45
50	2734.62	3541.56



	10	20	30	40	50
AODV	204.72	455.79	1439.53	2289.58	2734.62
WHAODV	395.23	830.34	2134.67	2936.45	3541.56

Figure 4.3: Comparison Of ENd-to-End Delay in AODV and WHAODV

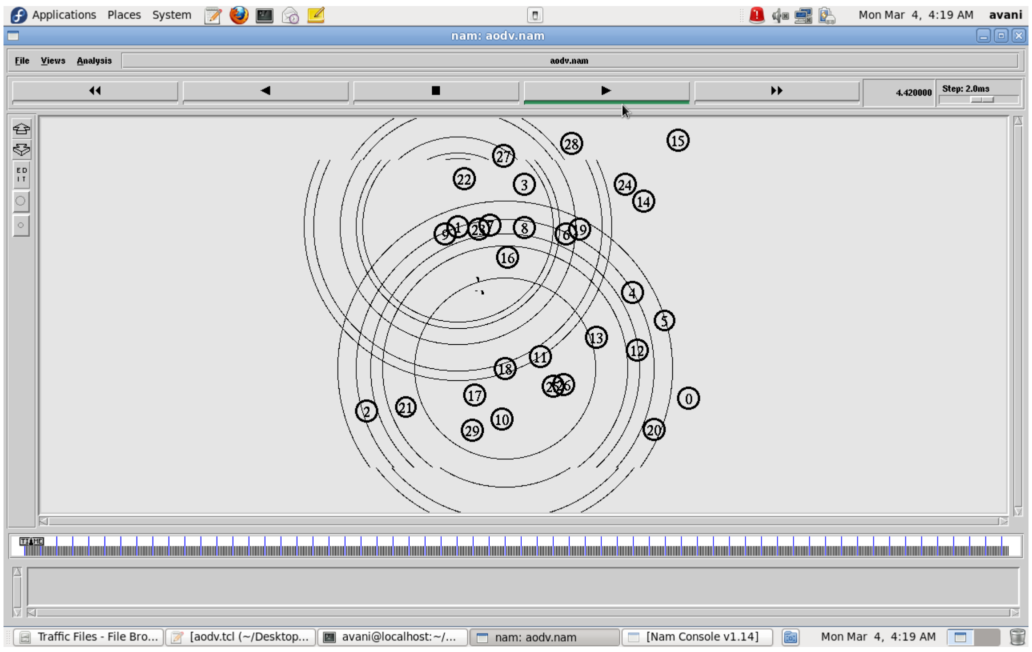


Figure 4.4: Running Aodv.tcl File

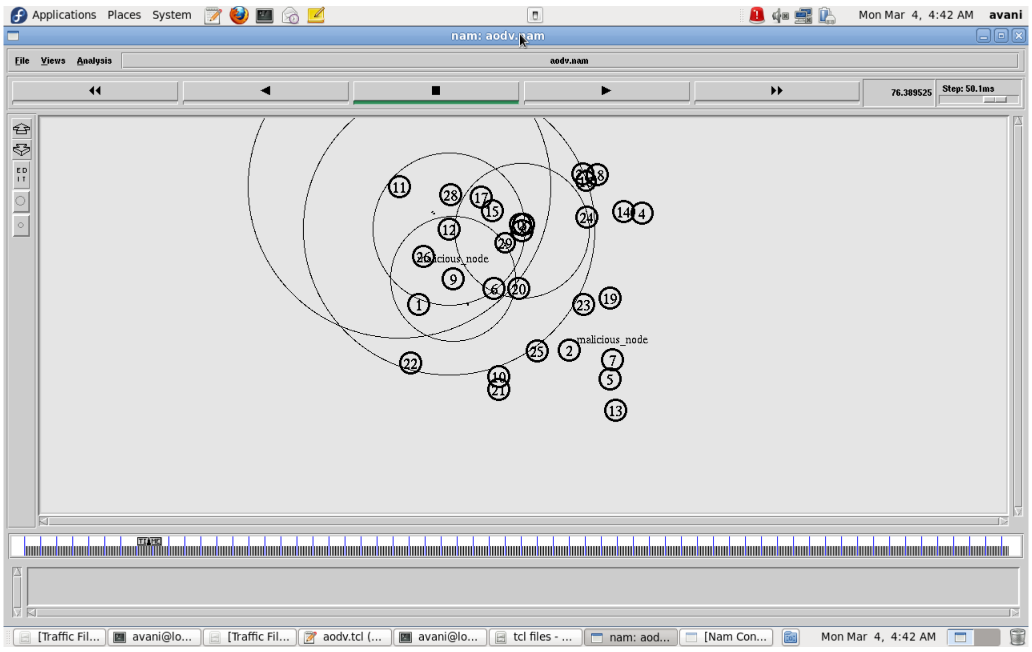


Figure 4.5: Running Wormhole.tcl File

Chapter 5

Finite State Machine based Intrusion Detection System

5.1 Introduction

Knowledge based intrusion detection system has knowledge about baseline system. It examines traffic and tries to identify patterns indicating suspicious activity which are different from the normal behavior of the system. This type of system needs to update knowledge base frequently. Knowledge base systems are attractive due to high efficiency and low false alarm rate and ability to detect unknown attacks.

5.2 Timed Finite State Machine

The Timed-FSM is 7-tuple S = $\langle S, I, O, s_o, \lambda_s, \sigma_s, \Delta_s \rangle$ where S is a finite nonempty set of states with the initial state So, I and O are finite disjoint input and output alphabets, $\lambda_s \subseteq$ S × I × O × S is a transition relation, $\sigma_S : \lambda_S \rightarrow$N is a speed function, and $\Delta_S :$ H ⊆ S→ S × N is a delay function.
If $\langle s, i, o, s' \rangle \epsilon \lambda_s$ and $\sigma_s \langle s, i, o, s' \rangle$ = t, denoted as s we say, that TFSM S, being in state s, accepts the input i and within t time units produces the output o, moves to the state s' and resets the clock (that is in the state s' and time units are counted

from 0).

If for a state s the function Δ_s is defined and $\Delta_s(S) = \langle s', t \rangle$, if no input is applied to the TFSM in state s within t time units then TFSM moves to the state s' and resets the clock (that is in the state s' and time units are counted from 0).

Similarly, when considering initialized machines, it is assumed that there exists a special reset that takes the TFSM from each state to the initial state and the clock is reset to 0. Correspondingly, it is assumed that the TFSM always starts from the initial state with the clock being set to 0. The next input can be applied to the TFSM after the TFSM has already produced an output to the previous input. After each input or output action the clock is reset to 0 [20].

5.3 Real Time Intrusion Detection

To understand the need of real time intrusion detection system we first have to understand the effect of attack on the system. The security incident cycle has to deal with threats to the confidentiality, integrity and availability. A defender first of all takes prevention measures which prevent a threat from becoming a reality. Reduction measures are measures that are performed in advance to reduce possible damage of an incident such as an intrusion. Deception measures are a special type of security measures meant for prevention, reduction and deception. Prevention, reduction and deception measures reduce the probability and the impact of an incident. However, this does not exclude possible occurrence. Therefore, the defender takes detection measures. After an intrusion is detected, the defender takes reaction measures. These reaction measures can be repressive in order to block the repetition of the intrusion. When an intrusion results in damage to the integrity or availability of information, the next step in the security incident cycle is to take correction measures to undo the damage that was done. The final step in the security-incident cycle consists of an effectiveness evaluation of the security measures taken. The time-axis model as shown in Figure can be related to the incident cycle described above. Both models have a point where an intrusion leads to damage. Where the incident-cycle models the defenders actions in the periods of time before and after the damage, the time-axis model shows the attackers actions for these two periods of time. When an attack is detected in the pre-attack stage, this is called pre-attack detection. Similarly when an attack is detected in the actual attack stage this is called attack-detection. It is also important to recognize that a system was attacked and that possible damage has occurred or that there is a security breach. The detection of an attack in the post-attack stage is defined as damage detection or post-attack detection. For intrusion detection this implies that IDSs should operate in a real-time manner. Real-time intrusion detection can be seen from two viewpoints. Firstly, within the boundaries of technology an alarm should be available to the response managers as soon as possible. And secondly, an intrusion should be detected as early as possible on the timeline of an attack. An important property of the analyzer to achieve the latter is the ability to correlate data [21].

Chapter 6

The Proposed Approach

6.1 Proposed TFSM Based IDS

The proposed approach for Wormhole attack detection employs the concept of Timed Finite State Automata (TFSM) to detect intrusion. A Timed Finite State Automata or a Timed Automata is a five tuple <Q, $\sum$, C, E, qo> where
Q is a finite set of symbols denoting states
$\sum$ is a set of symbols denoting the possible inputs
C is finite set of clocks
E is set of transitions
qo ϵ Q is the initial state
The firing of a transition <q, q', α , λ , δ> ϵ E from current state q to state q' is controlled by the enabled inputs α of $\sum$ and the clock constraint $\delta \epsilon \phi$(C) .

Source node sends route request. Intermediate nodes forward the request and count the number of hops every time. If intermediate nodes constitute a wormhole, they tunnel the packets to next end. If the packet reaches destination, it sends reply to source node. If malicious nodes receive the route reply they tunnel the RREP packets to the other end of the tunnel. Destination sends Route Reply through intermediate node to source node. When source receives Route Reply it records routes to destination. Source sends data packets over multiple paths. Here we propose that source would forward the packets through three best routes. Destination receives the packet from multiple path with different time stamp. Hence, firstly, the nodes which form malicious tunnel are not involved in routing and packet delivery itself because the assumption says that the route would be longer.

If the destination receives packets from three different paths, then it won't be able to determine the malicious activity. But if it is not able to receive same message from three different paths, it executes an FSM. Before initiating the TFSM, the destination finds intermediate nodes of every path. In that case it first finds the shortest path's node then the next shorter one and so on. So fastest path's nodes are good nodes and find common node's of all the paths which are good nodes. Now it would check the remaining nodes of longest path's node which may be detected as malicious nodes. To determine the same we employ a TFSM as shown below:

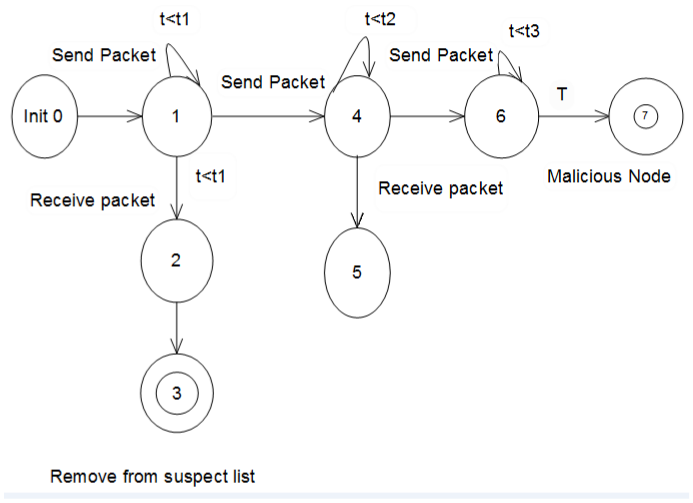


Figure 6.1: TFSM Based Wormhole Detection[30]

6.2 Implementation Of TIDS for Wormhole Detection

step 1: TFSM is triggered from the destination node and moves from init 0 state to state 1.

step 2: Node Broadcast hello packets to their neighbours. If RREQ does not arrive within time period defined FSM RESET and reset to its initial state(init 0)

step 3: Upon the reception of first RREP FSM moves to state 2.

step 4: If it reply within time interval it goes to state 3 and remove node from the suspected list.

step 5: If it is not, it waits in the same state for time T recv. If the timer expires without receiving another RREP it gets Normal Reset (N RESET) and moves to initial state.

step 6: If within the timer it receives another RREP(s) it moves to the state 5 and remove it from suspected list.

step 7: If it does not receive RREP within the timer it checks for another time delta and At global detection state the source node sends Alarm inquiry request packet and wait for reply till T alarm time, if it receives reply the pre alarm gets reset (A reset) else if node will not receive reply it declare as malicious node.

Chapter 7

Analysis of results

7.1 Simulation Parameters

The experiments were carried out using the network simulator (ns-2). The scenarios developed to carry out the tests use as parameters the mobility of the nodes and the number of active connections in the network. The module explained above was tested with the previously developed attacks. The choices of the simulator parameters that are presented in table 7.1 consider both the accuracy and the efficiency of the simulation. Figure 7.1 shows nam file for ids system running to detect wormhole node. Figure 7.2 shows result for wormhole node detected.

Table I: Simulation parameters

Property	Value
Nodes	10,20,30,40,50
Simulation time	600Sec
Mobility model	Random way point
Coverage area	750m*750m
Maximum speed	20 m/s
Pause time	1.0 Sec
Traffic type	Constant Bit Rate(CBR/UDP)
Send Rate	10 packets/sec
Packet size	512 bytes

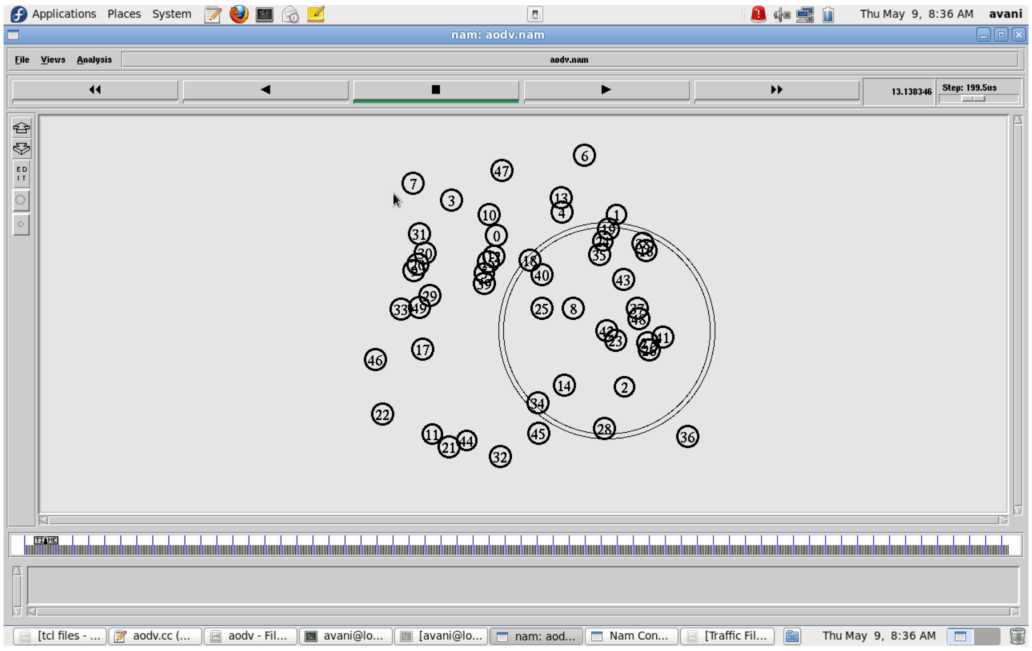
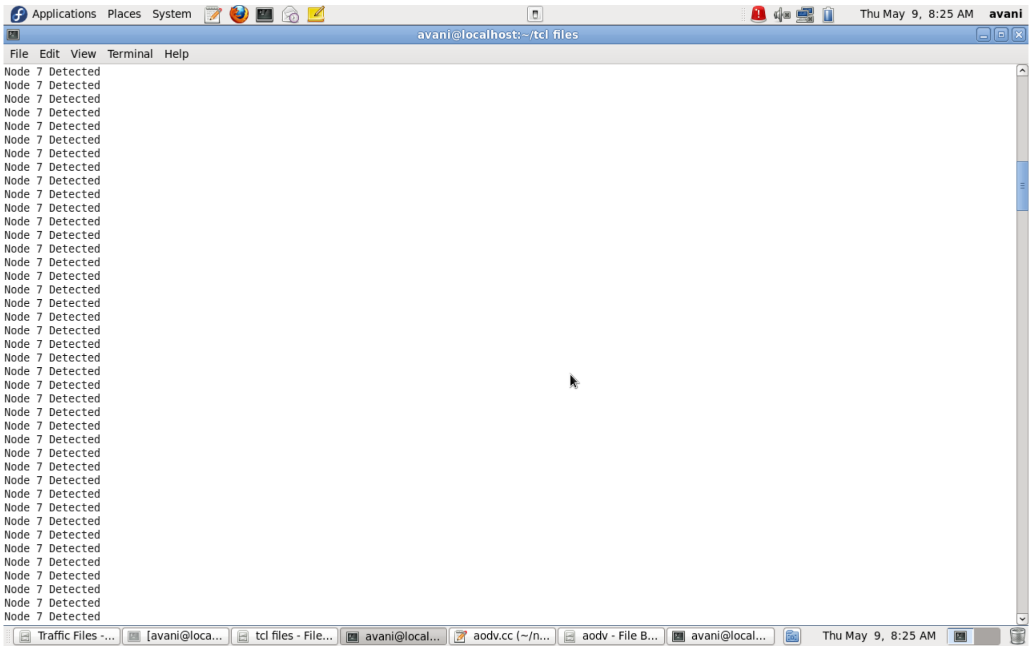


Figure 7.1: Running WormholeIDS.tcl file


Applications Places System
Thu May 9, 8:25 AM avani
avani@localhost:~/tcl files
File Edit View Terminal Help

```
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
Node 7 Detected
```

Figure 7.2: Wormhole Detection

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Here we analyzed different techniques of intrusion detection for MANET and their advantages and disadvantages. We have discussed the effect of wormhole attack in ad hoc networks and proposed a new approach for detection of wormhole attack. The solution detects the malicious nodes and isolates it from the active data forwarding. In this thesis, effect of the Wormhole attack in an AODV network is analyzed. For this purpose, AODV protocol is implemented and then wormhole node is appended in that to analyze its effectiveness. Different scenarios are simulated where in each scenario two node are made wormhole while other remain normal AODV nodes. In presence of wormhole node in routing, PDR is reduced and End-to-End Delay is increased. For detection of the wormhole attack we have proposed IDS. After implementing the IDS, wormhole node are detected.

8.2 Future Work

- To recover network from wormhole attack
- To choose another path for data transfer and try to find optimize solution.
- To extend TIDS for to provide security from more attacks.

Bibliography

[1] Oleg Kachirski, Ratan Guha, "Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks".

[2] Paul Brutch, Calvin Ko, "Challenges in Intrusion Detection for Wireless Ad-hoc Networks".

[3] Dr Chandra Shekar Reddy Putta, Dr K.Bhanu Prasad, Dilli Ravilla, Murali Nath R.S, M.L.Ravi Chandra, "Performance of Ad hoc Network Routing Protocols in IEEE 802.11", Intl Conf. on Computer and Communication Technology, IEEE, 2010.

[4] http://en.wikipedia.org/wiki/List of ad hoc routing protocols.

[5] Satria Mandala, Md. Asri Ngadi, A. Hanan Abdullah, "A Survey on MANET Intrusion Detection".

[6] Yi-an Huang, wenkee Lee, "A cooperative Intrusion Detection System For Ad Hoc Networks".

[7] Sumitra Menaria, Prof Sharada Valiveti, Dr. K Kotecha, Nirma University, "Comparative study of Distributed Intrusion Detection in Ad-hoc Networks", International Journal of Computer Applications (0975-8887) Volume8- No.9, October 2010.

[8] S.Bose, A.Kannan, "Detecting Denial of Service Attacks using Cross Layer based Intrusion Detection System in Wireless Ad Hoc Networks", IEEE-International Conference on Signal processing, Communications and Networking, Jan 4-6,2008.

[9] Sevil Sen, John A. Clark, Juan E. Tapiador, "Security Threats in Mobile Ad Hoc Networks".

[10] Abhay Kumar Rai, Rajiv Ranjan Tewari, Saurabh Kant Upadhyay, "Different Types of Attacks on Integrated MANET-Internet Communication", Department of Electronics and Communication, University of Allahabad.

[11] Thaier Hayajnesh, Prashant Krishnamurthy, David Tipper, "De-Worm: A Simple Protocol to Detect Wormhole Attack in Wireless Ad Hoc Networks", Third International Conference on Network and System Security 2009.

[12] Phuong Van Tran, Le Xuan Hung, Young-koo Lee, Sungyoung Lee, Heejo Lee, "TTM: An Efficient Mechanism to Detect Wormhole Attacks in Wireless Ad Hoc Networks", IEEE 2007.

[13] Asad Amir Pirzada, Chris McDonald, "Detecting and Evading Wormholes in Mobile Ad-hoc Wireless Networks", International Journal of Network Security, Vol.3, No.2, PP.191-202, Sept. 2006

[14] Hon Sun Chin, King-Shan Lui, "DelPHI: Wormhole Detection Mechanism for Ad Hoc Networks", IEEE 2006.

[15] Haivu, Ajay Kulkarni, Kamil Sarac, Neeraj Mittal, "WORMEROS: A New Framework for Defending against Wormhole Attacks on Wireless Ad Hoc Networks".

[16] A.VANI, D.Sreenivasa Rao, "A Simple Algorithm for Detection and Removal of Wormhole Attacks for Secure Routing In Ad Hoc Wireless Networks", International Journal on Computer Science and Engineering (IJCSE).

[17] Saurabh Gupta, Subrat Kar, S Dharmaraja, "WHOP: Wormhole Attack Detection Protocol using Hound Packet", International Conference on Innovations in Information Technology, 2011.

[18] http://www.windowsecurity.com/articles/ids-part2-classification-methods-techniques.html

[19] Ioanna Stamouli, "Real-time Intrusion Detection for Ad hoc Networks", in: Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, 2005.

[20] Yian Huang, Wenke Lee, "Cooperative Intrusion Detection System for Ad Hoc Networks", In Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, pages 135147, Fairfax, Virginia, 2003.

[21] H.A.M. Luiijf, R. Coolen, "Intrusion Detection Introduction and Generics", Proceeding of RTO IST Symposium on Real Time Intrusion Detection, 2002.

[22] O. Kachirski, R. Guha, ”Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks”, Proceedings of the 36th Annual Hawaii International Conference on System Sciences, January 2003.

[23] Tiranuch Anantvalee, ”A Survey on Intrusion Detection in Mobile Ad Hoc Networks”, Department of Computer Science and Engineering Florida Atlantic University, Boca Raton.

[24] Abadalla Mahmoud and Ahmed Sameh ”Reputed authenticated routing for ad hoc networks protocol”, ACM, 2005.

[25] Prabhakaran, P.Sankar, ”Impact of Realistic Mobility Models on Wireless Networks Performance”, IEEE International Conference, June 2006.

[26] Abhay Kumar Rai, Rajiv Ranjan Tewari, Saurabh Kant Upadhyay, ”Different Types of Attacks on Integrated MANET”-Internet Communication. International Journal of Computer Science and Security (IJCSS) Volume(4): Issue(3).

[27] J. Wang, ns-2 tutorial,” 2004.

[28] mohit p. tahiliani-blogspot,” Tech. Rep.

[29] http://www.windowsecurity.com/articles/intrusion-detection-systems-ids-part-i-network-intrusions-attack-symptoms-ids-tasks-and-ids-architecture.html

[30] R.Obermaisser, C.ElSalloum, B.Huber, H.Kopetz, ”Modeling and Verification of Distributed Real-Time Systems using Periodic Finite State Machines”.

[31] W. Zhang, R. Rao, ”Secure routing in ad hoc networks and a related intrusion detection problem”, IEEE Military Communications Conference (MILCOM), vol. 2, 13-16 p. 735- 740,2003.

[32] T. Anantvalee, J. Wu. ”A Survey on Intrusion Detection in Mobile Ad Hoc Networks”, Book Series Wireless Network Security, Springer, pp. 170 - 196, ISBN: 978-0-387-28040-0 (2007).

[33] K. Kumar, ”Intrusion Detection in Mobile Adhoc Networks” Master’s Thesis, University of Toledo, December 2009.

[34] http://www.cs.virginia.edu/ cs757/slidespdf/cs757-ns2-tutorial1.pdf