Distributed Intrusion Detection and Prevention System for Ad Hoc Networks

By

Sumitra Menaria

09MCE025

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May, 2011

Distributed Intrusion Detection and Prevention System for 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

Sumitra Menaria

(09MCE025)

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May, 2011

Declaration

This is to certify that

i) 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.

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

Sumitra Menaria

Certificate

This is to certify that the Major Project entitled “Distributed Intrusion Detection and Prevention System for Ad Hoc Networks” submitted by Sumitra Menaria (09MCE025), 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
Guide, Associate Professor,
Computer Science & Engineering Dept.,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. S.N. Pradhan
Professor and PG-Coordinator,
Computer Science & Engineering Dept.,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. D.J. Patel
Professor and Head,
Computer Science & Engineering Dept.,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. K. Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad.

Abstract

These days, Ad Hoc Networks are in demand in some crucial applications due to their open architecture and the mobility feature. Here, nodes cooperate with each other for communication. This very characteristic poses an immense problem in Ad Hoc Networks from the Security Point of view. Also due to the lack of Central Administration, Ad Hoc Networks fall prey to the Insider Attacks. Conventional cryptographic authentication methods are not enough to detect insider routing attacks. Implementation of good Intrusion Detection Systems are ideal for insider attacks. Objective of the work is to develop timed finite state machine based distributed intrusion detection and prevention approach for AODV enabled ad hoc network to detect active routing attacks and to minimize the effect of attack on ad hoc network.

As ad hoc networks are fully distributed in nature without centralized administration, they needs distributed IDS which can detect insider attacks effectively as the detector nodes have monitoring information from other nodes. Reason for adapting TFSM based detection system is that TFSMs enable the system to detect malicious activity in real-time rather than using statistical analysis of previously captured traffic, which helps in detecting intrusion as early as possible on the timeline of an attack. TFSM also helps in minimizing the impact of an attack and maintains the performance of the network within acceptable limits.

Attacks are implemented as a testbed and we analyzed their effect on performance of ad hoc network. TFSM based Distributed Intrusion Detection System and prevention system is implemented using NS-2 simulator. The results are then analyzed based on the suggested evaluation metrics in order to verify their suitability for use in ad hoc networks.

Acknowledgements

My deepest thanks to **Prof. Sharada Valiveti**, Associate Professor, 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 **Dr. S.N.Pradhan**, Professor and PG-Coordinator of Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad for an exceptional support and continual encouragement throughout the Major project.

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

Lastly and most importantly, no words are enough to acknowledge constant support and sacrifices of my family members, especially my parents and husband because of whom I am able to complete my dissertation work successfully.

- Sumitra Menaria
09MCE025

Contents

List of Tables

List of Figures

Abbreviations

AODV	Ad hoc On-demand Distance Vector
FSM	Finite State Machine
HIDS	Host Based Intrusion Detection System
IDS	Intrusion Detection System
MANET	Mobile Ad hoc Network
NAM	Network AniMator
NIDS	Network Based Intrusion Detection System
NRL	Normalized Routing Load
OTcl	Object-oriented Tcl
PLR	Packet Loss Ratio
RERR	Route Error
RREP	Route Reply
RREQ	Route Request
Tcl	Tool Command Language
TFSM	Timed Finite State Machine

Chapter 1

Introduction

In recent years ad hoc networks are widely used because of mobility and open architecture nature. By providing communication in the absence of a predetermined infrastructure they are very attractive for many applications such as tactical and disaster recovery operations and virtual conferences. On the other hand, this flexibility introduces new security risks. Moreover, different characteristics of ad hoc networks make traditional security methods ineffective and incompetent for this new environment. Intrusion detection, which is an essential part of a security system, also presents challenges due to the dynamic nature of ad hoc networks, the absence of central administration, and their highly constrained nodes. The mobility of wireless devices demands more flexible, stronger and efficient defense schemes.

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 grouping and the research achievements in the field of IDS . Active routing attack against AODV protocol like black hole attack, dropping routing traffic attack and RREQ flooding attack are implemented for analyzing the effect of attacks on performance of ad hoc network and are used as testbed for detection system . TFSM Timed Finite State Machine based intrusion detection and prevention 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 and prevention 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.

Specific Objectives

- To implement and analyze effect of attacks on ad hoc networks.
- To study existing intrusion detection system for AODV based ad hoc network.
- To propose TFSM based distributed intrusion detection and prevention system.
- To test and validate the effectiveness of proposed 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 currently, although we are not making changes in protocol but the system is designed for AODV only. There is still scope for 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.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2, *Literature Survey*, focuses on four sections, overview of vulnerabilities in ad hoc networks, attacks possible on ad hoc networks, survey of intrusion detection system and research achievements in IDS field.

Chapter 3, *Study of NS-2 Simulator*, focuses on basics of NS-2, a discrete event network simulator which is heavily used in ad-hoc networking research. Also it includes an overview of programming language and process of integrating new protocol.

Chapter 4, *The Implementation of Attacks*, includes overview of AODV protocol, implementation of active routing attacks on AODV and analysis of the results.

Chapter 5, *Finite State Machine Based Intrusion Detection System*, includes overview of TFSM, need of real time intrusion detection and research achievements in the field of FSM based intrusion detection system.

Chapter 6, *Proposed TFSM based distributed intrusion detection and prevention system*, includes proposed TFSM based distributed intrusion detection and prevention system in order to achieve the main objective of the project work. It will also cover the analysis of the results.

Chapter 7, *Testing and Analysis of results*, includes analysis of results in terms of three evaluation metrics: Packet Delivery Ratio, Packet Loss Ratio, Routing Overhead.

Finally, in **chapter 8** concluding remarks and future work is presented.

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.

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 [1]. These categories are Table Driven Routing Protocols and On-Demand Routing Protocols.

2.1.1 Table Driven Routing Protocols

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

Table driven routing protocols are: Destination-Sequenced Distance Vector Routing Protocol (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR), Hierarchical State Routing (HSR), Zone-based Hierarchical Link State Routing Protocol (ZHLS) and Clusterhead Gateway Switch Routing Protocol (CGSR)

2.1.2 On-Demand Routing Protocols

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.

For my thesis work I'll be using AODV protocol, for implementing different types of attacks and detection system. Hence details about AODV is given in next section.

2.2 Ad-Hoc On-Demand Distance Vector Routing Protocol

The Ad hoc On-demand Distance Vector (AODV) protocol [2] 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.

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 fire walls are not applicable. Vulnerabilities in ad hoc network described in [3, 4] are:

- *Dynamic topology:*

  Due to dynamic topology ad hoc networks require sophisticated routing protocols. A particular difficulty is that misbehaving node can generate wrong routing information which is hard to discover. Mobility of devices also creates a problem.

- *Absence of infrastructure:*

  Ad hoc networks do not have any fixed infrastructure which makes traditional security mechanism of cryptography and certification inapplicable.

- *Vulnerability of nodes:*

  Physical protection of nodes is not possible hence they can more easily be captured and falls under the control of an attacker.

- *Vulnerability of channels:*

  In wireless network, message eavesdropping and injection of fake messages into the network is easy without having physical access to network components.

2.4 Type of Attacks

For the purpose of 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 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 [3].

Another category is of the attacks that affect the routing of traffic in network. These types of attacks are classified into following subcategories [6]:

- *Availability Attacks :* Availability implies that requested services are available in a timely manner even though there is a potential problem in the system. Attacks against availability are classified as:
  - *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 behave properly when it itself wants to send data.
  - *Fabricated route Attacks:* Fabrication refers to attacks performed by generating false routing messages. For eg. 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 fare shares of bandwidth by flooding RREQ for random address or two malicious nodes send large volume of data between themselves, there by depleting the available network bandwidth.
  - *Selfishness:* In these attacks, a malicious node behaves selfishly, using the network for its own needs, without participating in the overall routing process.
- *Integrity Attacks:* Integrity guarantees that information received by the node has not been tempered in the transmission. Attacks against integrity are:
  - *False Route Propagation Attacks:* Malicious node advertises a route to the destination node with highest sequence number and tries to attract all traffic towards itself and then drops it.
  - *Misrouting Attacks:* In this class of attacks, a malicious node attempts to send a data packet to the wrong destination.
  - *Man-in-the-Middle Attack:* A malicious node can combine the spoofing attack and the dropping of packets to perform a man-in-the-middle attack.

- *Authentication and Non-Repudiation Attacks:* Authentication means identifying a peer node with which it is communicating and non-repudiation is to prove that particular sender has sent the message. Without Authentication, an attacker can impersonate any node, and in this way one by one node, it can gain control over the entire network

- *Confidentiality Attacks:* Confidentiality ensures that classified information in the network is never disclosed to unauthorized entities. It can be classified as:

  - *Location Disclosure Attack:* Attack based on disclosure of physical location of particular node.

  - *Content Disclosure Attack:* Attack focuses on disclosure of the content of the message.

2.5 Classification of Intrusion Detection System

2.5.1 Based on data collection mechanisms

An intrusion detection system (IDS) can be classified as network-based or host-based according to the audit data that is used [7, 8].

a. 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.

b. 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.

2.5.2 Based on detection techniques

a. Signature or Misuse based IDS:

   Misuse detection uses a priori knowledge on intrusions and tries to detect attacks based on specific patterns or signatures of known attacks. Although misuse detection systems are very accurate in revealing known attacks, their basic disadvantage is that attacking mechanisms are under a continuous evolution, which leads to the need for an up-to-date knowledge base.

b. Anomaly based IDS:

   Anomaly detection has the advantage of being able to discover unknown attacks while it adopts the approach of knowing what is normal. As a result it attempts to track deviations from the normal behavior that are considered to be anomalies or possible intrusion.

c. Specification based IDS:

   The system defines a set of constraints that describe the correct operation of a program or protocol. Then, it monitors the execution of the program with respect to the defined constraints.

2.6 Architecture of IDS

Based on the network infrastructures, the ad hoc network can be configured to either flat or multi-layer. The optimal IDS architecture for the ad hoc network may depend on the network infrastructure itself.

There are four main architectures on the network, as follows:

a. *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 also more suitable for flat network infrastructure than for multilayered network infrastructure [9].

b. *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 [10].

c. *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 which in some sense, act as control points which are similar to switches, routers, or gate ways in wired networks [11].

d. *The mobile agent for IDS architecture* uses mobile agents, a particular type of software which has the capability of moving from one host to another host. This architecture allows the distribution of the intrusion detection tasks. Mobile agents have features of reducing network overload, overcoming network latency, synchronous and autonomous execution, robustness and fault-tolerance, system scalability and operating in heterogeneous environments [15, 16].

2.7 Related Work

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

Yian Huang et al. in 2003 proposed an Cooperative and Distributed algorithm in

[10].

The model for an IDS agent is structured into six modules as shown in Figure 2.1. 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 initiates 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.

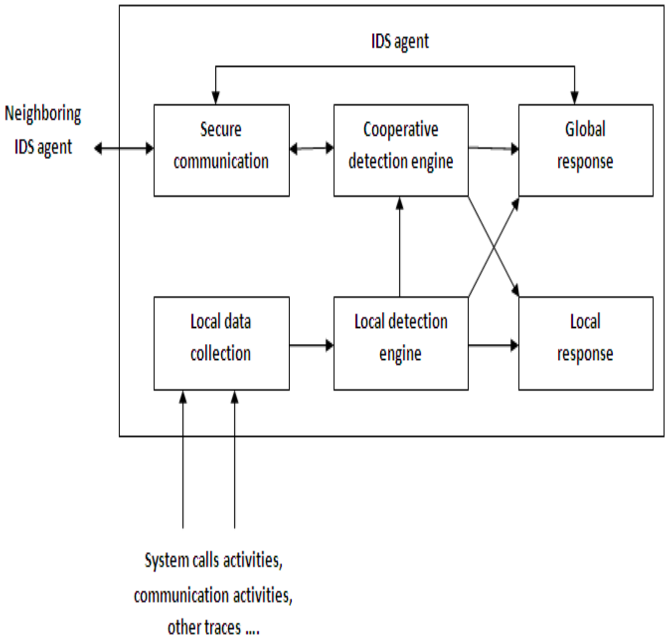


Figure 2.1: Cooperative and Distributed Model by Huang et al. [10]

Kachirski and Guha in 2002 has given distributed algorithm with multiple sensor in

[11]. They proposed a multi-sensor distributed intrusion detection system based on mobile agent technology. The system can be divided into three main modules, each of which represents a mobile agent with certain functionality: monitoring, decision-making or initiating a response. By separating functional tasks into categories and assigning each task to a different agent, the workload is distributed which is suitable for the characteristics of ad hoc networks. In addition, the hierarchical structure of agents is also developed in this intrusion detection system as shown in Figure 2.2.

- Monitoring agent: Network monitoring and Host monitoring are done by the agents of this class.

- Action agent: Due to the presence of host monitoring agents nodes can determine malicious activities. When there is strong evidence supporting the anomaly detection, this action agent can initiate a response, such as terminating the process or blocking a user from the network.

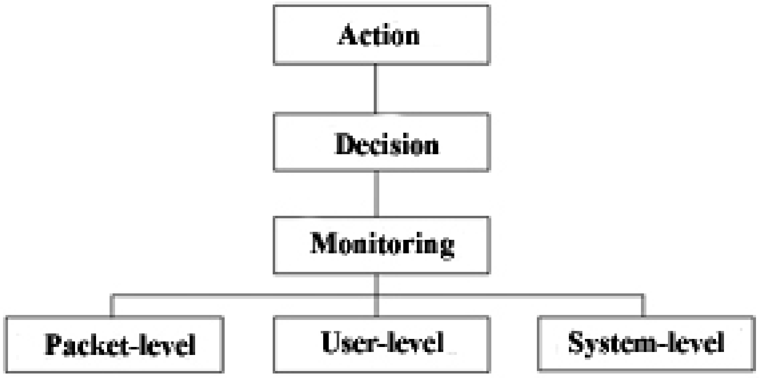


Figure 2.2: Distributed IDS Using Multiple Sensors by Kachirski et al. [11]

- Decision agent: Nodes having network monitoring agents can initiate decision agent for making decision based on analysis of collected data.

In case of insufficient evidence local detection agent reports to global decision agent for further investigation using packet-monitoring results that comes from the network-monitoring sensor that is running locally. If the decision agent concludes that the node is malicious, the action module of the agent running on that node will carry out the response. The network is logically divided into clusters with a single cluster head for each cluster. This cluster head have network monitoring agent (with network monitoring sensor) and the decision agent.

As decision agent performs the decision-making based on its own collected information from its network-monitoring sensor, other nodes have no influence on its decision. Mitrokotsa et al. in 2006 proposed a distributed model in [12]. The proposed intrusion detection system is composed of multiple local IDSs agents. Each IDS agent as shown in Figure 2.3 is responsible for detecting possible intrusions locally. The collection of all the independent IDS agents forms the IDS system for the mobile

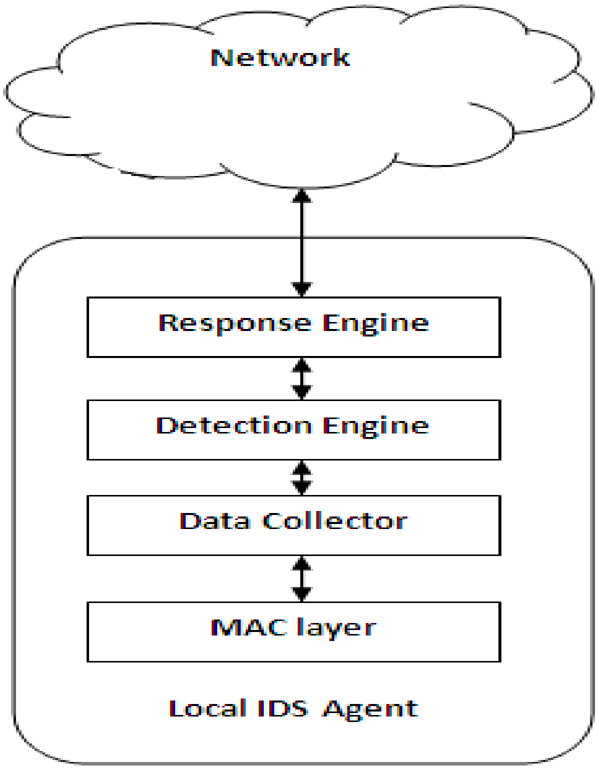


Figure 2.3: IDS with Multipule Local IDS by Mitrokotsa et al. [12]

wireless ad hoc network. Each local IDS agent is composed of Data Collector which is responsible for selecting local audit data and activity logs Detection Engine which is responsible for detecting local anomalies using local audit data. The local anomaly detection is performed using the eSOM classification algorithm.

Response Engine: If an intrusion is detected by the Detection Engine then the Response Engine is activated. The Response Engine is responsible for sending a local and a global alarm in order to notify the nodes of the mobile ad hoc network about the incident of intrusion. The local IDS agent could send ALERT messages to all potential traffic generators that exist in its routing table, thus achieving a global response to all nodes that are directly influenced by the malicious node.

Nakkeeran et al. in 2010 proposed an Agent Based cooperative and distributive model in [13]. This model provides the three different techniques to provide sufficient security solution to current node, Neighboring Node and Global networks. The following section outlines each module's work in detail.

System as shown in Figure 2.4 consist of home agents present on each node and gather local information from application layer to routing layer. If an attacker sends any packet to gather information or broadcast through this system, it calls the classifier construction to find out the attacks.

Neighboring node before transferring the message, sends mobile agent to the neighboring node and gathers all the information and returns back to the system and it calls classifier rule to find out the attacks. If there is no suspicious activity, then it will forward the message to neighboring node.

Data collection module is included for each anomaly detection subsystem to collect the values of features for corresponding layer in a system. Normal profile is created using the data collected during the normal scenario. Attack data is collected during the attack scenario. The audit data is collected in a file and it is smoothed so that it can be used for anomaly detection. Data preprocess is a technique to process the information with the test train data. In the entire layer of anomaly detection systems, the above mentioned preprocessing technique is used.

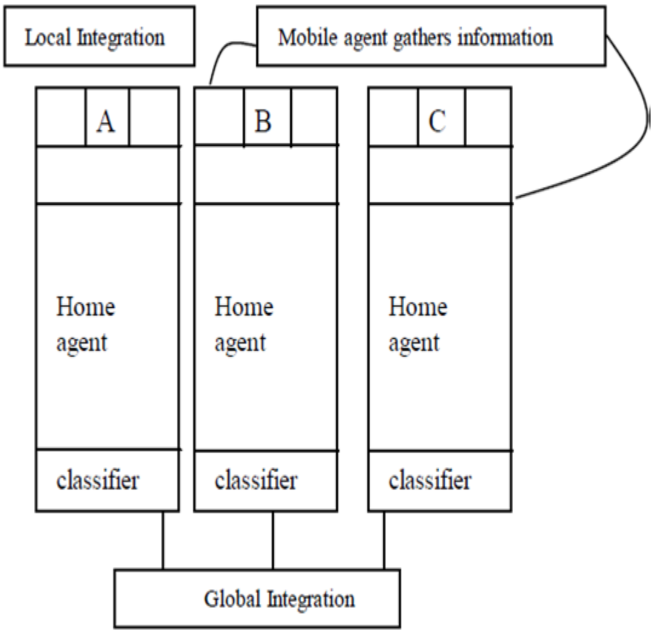


Figure 2.4: Agent Based cooperative and distributive model by Nakkeeran et al. [13]

Ping et al. in 2005 gave a distributed algorithm based on FSM in [14]. A network monitor is the node which monitors the behavior of nodes within its monitor zone. The monitor in a zone is selected by competition. A monitor employs a finite state machine(FSM) for detecting incorrect behavior in a node. It maintains a FSM for each data flow in each node. According to the author, node checks each ROUTE RREQ, ROUTE REPLY, ROUTE ERROR, DATA and if it find any maliciously modified entry then go to alarm. In this algorithm first monitor is selected for distributed monitoring of all nodes in networks. Secondly, it manually abstracts the correct behaviors of the node according as DSR and composes the finite state machine of node behavior. Intrusions, which usually cause node to behave in an incorrect manner, can be detected without trained date or signature.

Ricardo Puttini et al. in 2007 has developed a fully distributed algorithm. In fully distributed IDS distribution is not restricted to data collection but also applied to execution of the detection algorithm and alert correlation. Each node in the MANET runs a local IDS (LIDS) that cooperates with others LIDS. A mobile agent framework is used to preserve the autonomy of each LIDS while providing a flexible technique for exploring the natural redundancies in MANET to compensate for the dynamic state of wireless links between high mobility nodes. Attack detection is formally described by specification of data collection and attack signatures associated with such data and alerts generation and correlation.

2.8 Summary

In this chapter, routing protocol and vulnerabilities of ad hoc networks are discussed in detail. Classifications of attacks against ad hoc network are analyzed with their effect on network, which will be further useful in implementing testbed for detection of system's simulation. Classification of intrusion detection system based on collection mechanism and detection system is also described.

Most importantly architectures of intrusion detection systems are surveyed for finding out suitability of architecture for ad hoc network and was concluded that as ad hoc networks are distributed in nature, distributed intrusion detection system are more appropriate. Further the chapter contains research achievements in field of distributed intrusion detection system.

Chapter 3

Study of NS-2 Simulator

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.

3.1 Network Simulator-2

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 [15].

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. NS-2 has many and expanding uses including:

- To evaluate the performance of existing network protocols.
- To evaluate new network protocols before use.
- To run large scale experiments not possible in real experiments.
- To simulate a variety of IP networks.

NS-2 is available for both Windows and Linux platform. Under Linux, following components can be installed.

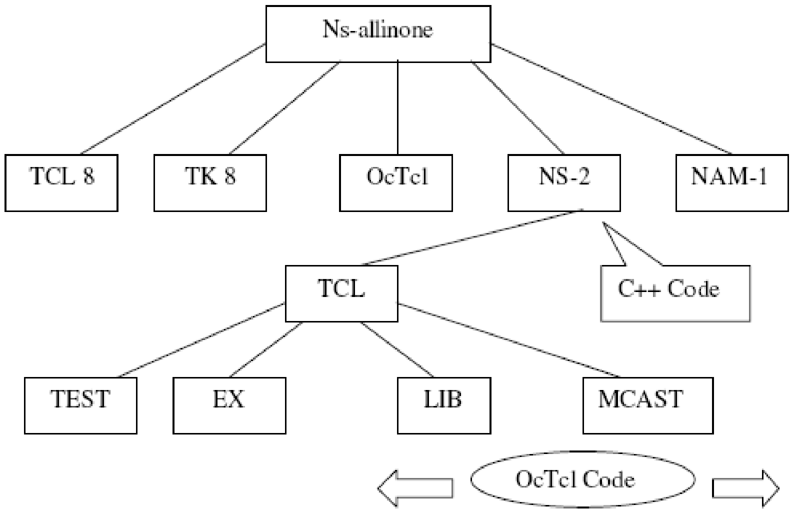


Figure 3.1: Directory structure of a NS-2

3.1.1 Architecture of NS-2

As shown in the simplified user's view of Figure 3.2, NS is an Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network set-up (plumbing) module libraries.

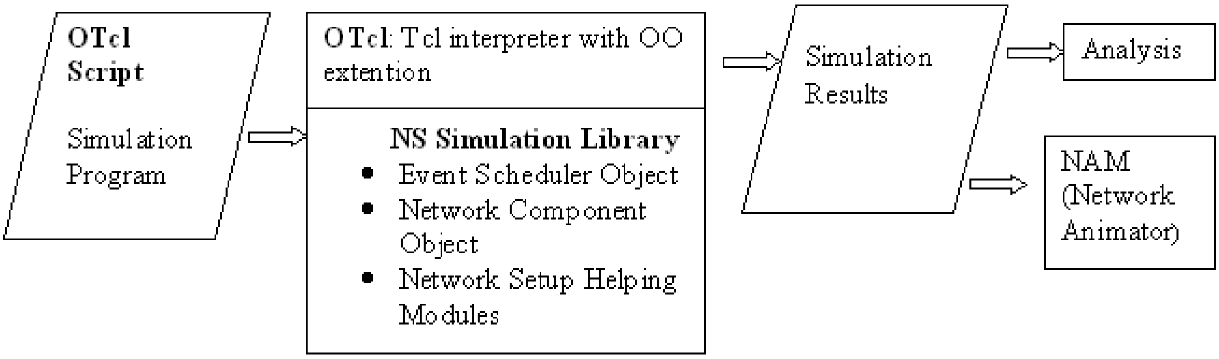


Figure 3.2: User's view of NS-2

To use NS-2, a user programs in the OTcl script language. An OTcl script will do the following.

- Initiates an event scheduler.
- Sets up the network topology using the network objects.
- Tells traffic sources when to start/stop transmitting packets through the event scheduler.

A user can add OTcl modules to NS-2 by writing a new object class in OTcl. These then have to be compiled together with the original source code.

Another major component of NS besides network objects is the event scheduler. An event in NS is a packet ID that is unique for a packet with scheduled time and the pointer to an object that handles the event. The event scheduler in NS-2 performs the following tasks:

- Organizes the simulation timer.
- Fires events in the event queue.
- Invokes network components in the simulation.

Depending on the users purpose for an OTcl simulation script, following simulation results are stored as trace files, which can be loaded for analysis by an external application:

- A NAM trace file (file.nam) for use with the Network Animator Tool
- A Trace file (file.tr) for use with XGraph or GnuPlot.

3.1.2 C++/OTcl linkage

NS2 is written in C++ with OTcl interpreter as a front end. For efficiency reason, NS separates the data path implementation from control path implementations. Languages used with NS-2 are:

a. Split-Language programming

   - Scripting Language (Tcl - Tool Command Language and pronounced as tickle)
   - System Programming Language (C/C++)

b. NS is a Tcl interpreter to run Tcl Scripts:
   TclCL is the language used to provide a linkage between C++ and OTcl. Toolkit Command Language (Tcl/OTcl) scripts are written to set up/configure network topologies. TclCL provides linkage for class hierarchy, object instantiation, variable binding and command dispatching. OTcl is used for periodic or triggered events.

   The following is written and compiled with C++

   - Event Scheduler
   - Basic network component objects.

   These compiled objects are made available to the OTcl interpreter through an OTcl linkage that creates a matching OTcl object for each of the C++ objects and makes the control functions and the configurable variables specified by the C++ object act as member functions and member variables of the corresponding

OTcl object. It is also possible to add member functions and variables to a C++ linked OTcl object.

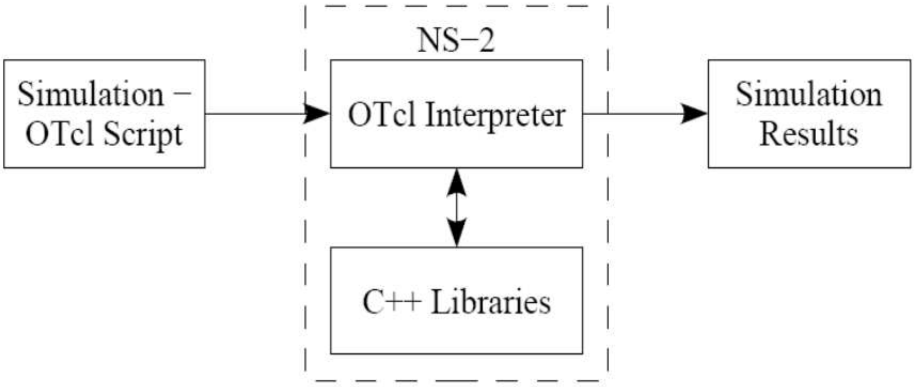


Figure 3.3: Architecture of NS-2

3.1.3 Network AniMation (NAM) Trace

NAM trace is records simulation detail in a text file, and uses the text file to play back the simulation using animation. NAM trace is activated by the command $ns namtrace-all $file, where ns is the Simulator handle and file is a handle associated with the file (e.g., out.nam in the above example) which stores the NAM trace information. After obtaining a NAM trace file, the animation can be initiated directly at the command prompt through the following command:

```
>> nam filename.nam
```

Many visualization features are available in NAM. These features are for example animating colored packet flows, dragging and dropping nodes (positioning), labeling nodes at a specified instant, shaping the nodes, coloring a specific link, and monitoring a queue

3.2 Programming languages

This topic covers the basic elements of the programming languages, which are essential for developing NS2 simulation programs. These include Tcl/OTcl which is the basic building block of NS2 and AWK which can be used for post simulation analysis [16].

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.

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.

AWK Programming Structure:

The general form of an AWK program is shown below:

```
BEGIN <initialization>
<pattern1> <actions>
<pattern2> <actions>
. . .
END <final actions>
```

Prior to processing an input text file, AWK performs <initialization> specified in the curly braces located after the reserved word BEGIN. Then, for each record, it performs actions if the records match with the corresponding pattern. After processing the entire file, it performs <final actions> specified in the curly braces located after the reserved word END.

Chapter 4

Implementation of Attacks

From literature survey of possible attacks on ad hoc network, three most prominent attacks are choosen to be implemented in ns-2. As ns-2 simulator does not provide any package for testing intrusion detection system,first of all testbed is prepared to analyze the effect of attacks on ad hoc network's performance. The implemented attacks will be used for the comparison and for testing of detection system.

For implementation purpose new behaviors are added in AODV protocol. Whole network works on AODV but the node specified "attacker" in tcl script will exhibit the malicious behavior assigned to it. Three different attacks blackhole attack, flooding attack and dropping routing traffic attacks are implemented and results are analyzed.

4.1 Implementation of Black hole Attack

To implement black hole attack (BHAODV), in tcl file one node was assigned name "attacker". Then changes were made in "aodv.cc" file such that when packet comes to the attacker node, it will perform malicious activity assigned to it. The main concept of blackhole attack is that the malicious node replies to all RREQ packets it receives regardless of whether it has route to the specific destination or not. Malicious node tries to attract maximum traffic towards itself and drops all the data packets it receives if they are not destined to it. Black hole attack cause maximum perfomance

degradation of the network.
The methods of AODV protocol that are modified to achieve black hole attack are the recvRequest, sendReply and recv.

For implementing black hole attack recvRequest function was modified in such a way that it will first check that the request is sent by him or has recently heard that request then will simply discard the request. If the condition if not true then will check if he is the destination of the request, if yes then will send route reply with sequence and hop count incremented by one else will on add some random number to the sequence number and will send reply and add that source in the list for future reference.

Next some changes are made to the sendReply function such that it will not generate any kind of error. Finally changes are made in recv function. When malicious node receive data packet it will check destination address, if it is destined to him it will accept otherwise it will discard all other packets. Random number will be generated and added to the sequence number and included in the header of the route request packet which makes attack more realistic.

4.2 Implementation of Dropping Routing Traffic Attack

For implementing dropping routing traffic attack (DROPAODV) same procedure is followed.
The main characteristic of the attack is that it will participate in normal route identification procedure, it will check received packet, check if he is not the destination of the packet and will drop routing packet. It will work normally when he has to send any data. To implement above attack, methods "recvRequest" and "recvReply" are modified such that a malicious node acts selfishly and drops all routing traffic that it

is not destined for itself. Thus, upon of a RREQ packet it checks if the destination of the route discovery is itself and if this holds then it further processes the packet and sends a RREP . When it receives a RREP packet it checks if it has sent the original request for this route and if this holds it adds the new route to its routing table. The RERR packets are processed normally in all cases. In any other cases it drops the packets without processing them further.

4.3 Implementation of Route Request Flooding Attack

Same procedure was followed for implementing Route Request Flooding Attack (DOSAODV). Changes are made in aodv.cc file for making the node behaving maliciously.

The only modification that had to be made was a loop that sends frequent unnecessary routing traffic. Method modified to implementing attack was sendRequest method. The destinations that are used in the unnecessary RREQ and RERR packets are nodes that do not exist in the network. In order to see that these packets are not to be discarded automatically by the protocol implementation, the destination nodes should be different each time. Hence, a function that returns pseudo random addresses was developed to realize this task.

4.4 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 4.1 consider both the accuracy and the efficiency of the simulation.

Table 4.1: Simulation Parameters

Simulator	ns-2.34
Simulation duration	1000 sec
Simulation area	1000 * 1000 m
Number of Nodes	30
Transmission range	250 m
Movement model	Random waypoint
Maximum speed	5–20 m/sec
Traffic type	CBR (UDP)
Data payload	512 bytes
Number of malicious nodes	5%
Number of active connection	3–29

4.5 Performance Metrics

Following performance metrics are used for analyzing the effect of attacks on ad hoc network.

- Packet Delivery Fraction: 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.

$$PDR = \frac{\Sigma \ Packets \ \ received \ \ by \ \ destination}{\Sigma \ Packet \ \ sent \ \ by \ \ source} \times 100 \tag{4.1}$$

- Normalized Routing Load: NRL routing load is the number of routing packets transmitted per data packet sent to the destination. Also each forwarded packet is counted as one transmission. This metric is also highly correlated with the number of route changes occurred in the simulation. It should be lower for efficient network.

$$NRL = \frac{\Sigma \ Sent \ \ received \ \ and \ \ forwarded \ \ routing \ \ packets}{\Sigma \ Sent \ \ received \ \ forwarded \ \ data \ \ packets} \tag{4.2}$$

- Packet loss ratio: PLR is the ratio of data packets lost over number of data packet sent during simulation. The metric should have lower value for the efficient network.

$$PLR = \frac{\Sigma \ Dropped \ \ data \ \ packets}{\Sigma \ Sent \ \ data \ \ packets} \times 100 \tag{4.3}$$

4.6 Evaluation of Black Hole Attack

Two metrics that were used in the evaluation of the black hole attack detection system are the delivery fraction and the packet loss ratio. All the metrics are plotted against:

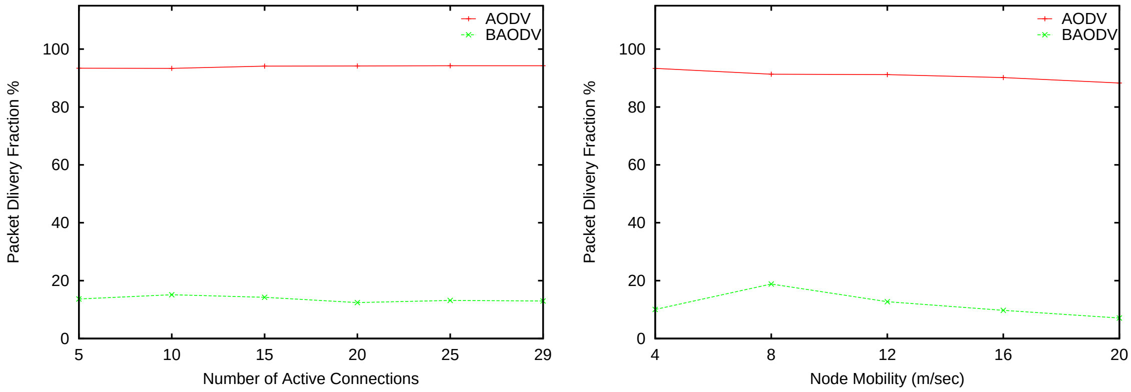
- Number of active connections
- Node mobility

Attacks are implemented in such a way that it decreases packet delivery fraction to a great extend as attacker node tries to attract maximum traffic towards itself. Packet loss ratio increases as all attracted traffic is dropped if not destined to attacker node. Table 4.2 shows the simulation results of packet delivery fraction for 30 nodes with varying number of active connections and varying node mobility.

Table 4.2: Packet Delivery Fraction

Parameter	**Values**	**AODV**	**BHAODV**
Number of active connections	5	93.41	13.67
	10	93.34	15.12
	15	94.12	14.24
	20	94.18	12.42
	25	94.26	13.17
	29	94.25	12.97
Node mobility (m/s)	4	93.34	10.04
	8	91.33	18.84
	12	91.19	12.73
	16	90.16	9.74
	20	88.29	7.09

Figure 4.1(a) shows the packet delivery fraction with respect to varying number of active connections and 4.1(b) shows the packet delivery fraction with respect to varying node mobility. Average packet delivery fraction of AODV with respect of number of active connection is 93.92% and with varying node mobility it is 90.86%. Due to



(a) PDF Vs number of Active connections

(b) PDF Vs node mobility

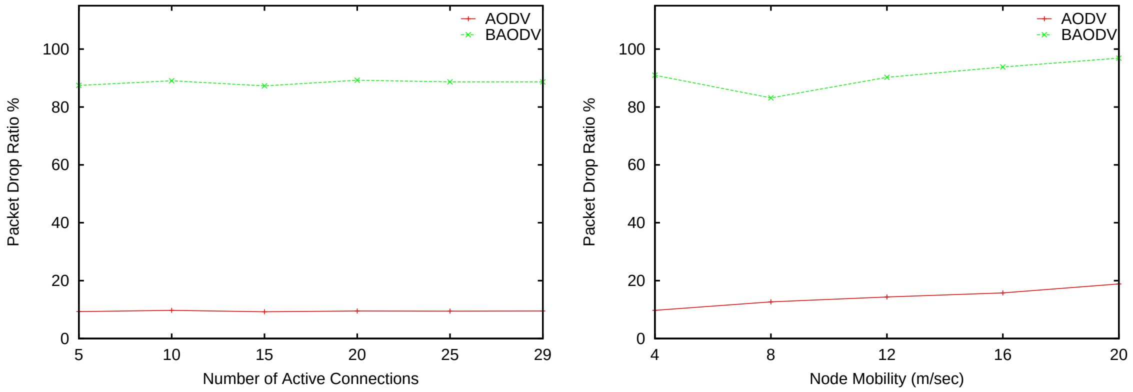
Figure 4.1: Packet Delivery Fraction

the effect of blackhole attack, PDF is decreased by 80.33% with respect to number of active connection and 79.17% with respect to node mobility.

Table 4.3 shows the simulation results of packet loss ratio for 30 nodes with respect to varying number of active connections and varying node mobility. Figure 4.2(a) shows the packet loss ratio with respect to varying number of active connections and 4.2(b) shows the packet loss ratio with respect to varying node mobility. Average

Table 4.3: Packet Loss Ratio

Parameter	**Values**	**AODV**	**BHAODV**
Number of active connections	5	9.29	87.45
	10	9.72	89.07
	15	9.23	87.3
	20	9.5	89.25
	25	9.47	88.68
	29	9.52	88.68
Node mobility (m/s)	4	9.72	90.98
	8	12.66	83.14
	12	14.35	90.26
	16	15.75	93.81
	20	18.87	96.89



(a) PLR Vs number of Active connections

(b) PLR Vs node mobility

Figure 4.2: Packet Loss Ratio

packet loss ratio of AODV with respect of number of active connection is 9.5% and with respect to varying node mobility is 14.3%. Due to the effect of black hole attack, PLR is increased by 78.95 % with respect to number of active connection and 76.75% with respect to node mobility.

4.7 Evaluation of Flooding Attack

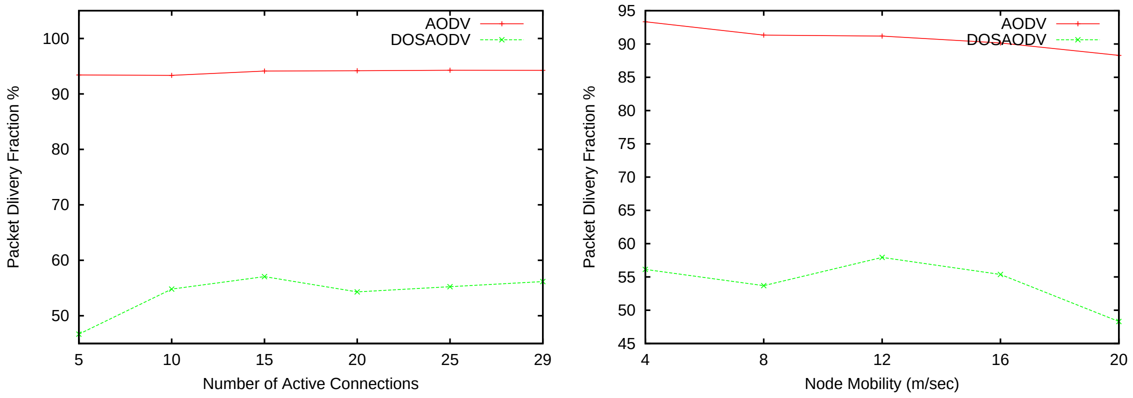
Flooding attack is designed to flood routing packets over the network for the node which does not exist in the network. As destination node is not in network no one is able to reply for the request and within some time, network gets flooded with unwanted routing traffic. Flooding attack mainly affects packet delivery fraction, packet loss ratio and normalized routing load. All the metrics are plotted against:

- Number of active connections
- Node mobility

Table 4.4 shows the simulation results of packet delivery fraction for 30 nodes with respect to varying number of active connections and varying node mobility. Figure 4.3(a) shows the packet delivery fraction with respect to varying number of active connections and 4.3(b) shows the packet delivery fraction with respect to varying node mobility. Average packet delivery fraction of AODV with respect of number of active connections is 93.92% and with respect to varying node mobility it is 90.86%.

Table 4.4: Packet Delivery Fraction

Parameter	**Values**	**AODV**	**DOSAODV**
Number of active connections	5	93.41	46.69
	10	93.34	54.81
	15	94.12	57.06
	20	94.18	54.31
	25	94.26	55.25
	29	94.25	56.16
Node mobility (m/s)	4	93.34	56.13
	8	91.33	53.69
	12	91.19	57.93
	16	90.16	55.37
	20	88.29	48.31



(a) PDF Vs number of Active connections

(b) PDF Vs node mobility

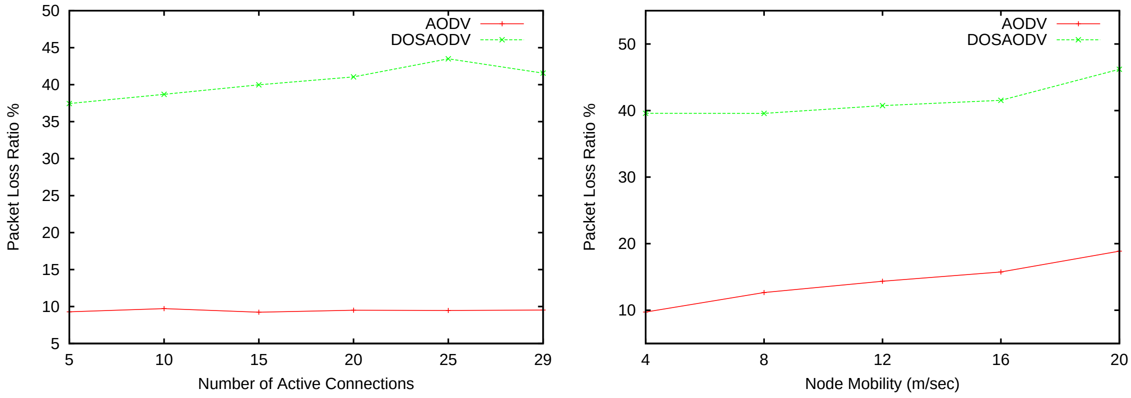
Figure 4.3: Packet Delivery Fraction

Due to the effect of flooding attack PDF is decreased by 39.8 % with respect to number of active connection and 36.6% with respect to node mobility.

Table 4.5 shows the simulation results of packet loss ratio for 30 nodes with varying number of active connections and varying node mobility. Figure 4.4(a) shows the packet loss ratio with respect to varying number of active connections and 4.4(b) shows the packet loss ratio with respect to varying node mobility. Average packet

Table 4.5: Packet Loss Ratio

Parameter	**Values**	**AODV**	**DOSAODV**
Number of active connections	5	9.29	37.45
	10	9.72	38.69
	15	9.23	39.98
	20	9.50	41.05
	25	9.47	43.50
	29	9.52	41.56
Node mobility (m/s)	4	9.72	39.59
	8	12.66	39.58
	12	14.35	40.74
	16	15.75	41.53
	20	18.87	46.20



(a) PLR Vs number of Active connections

(b) PLR Vs node mobility

Figure 4.4: Packet Loss Ratio

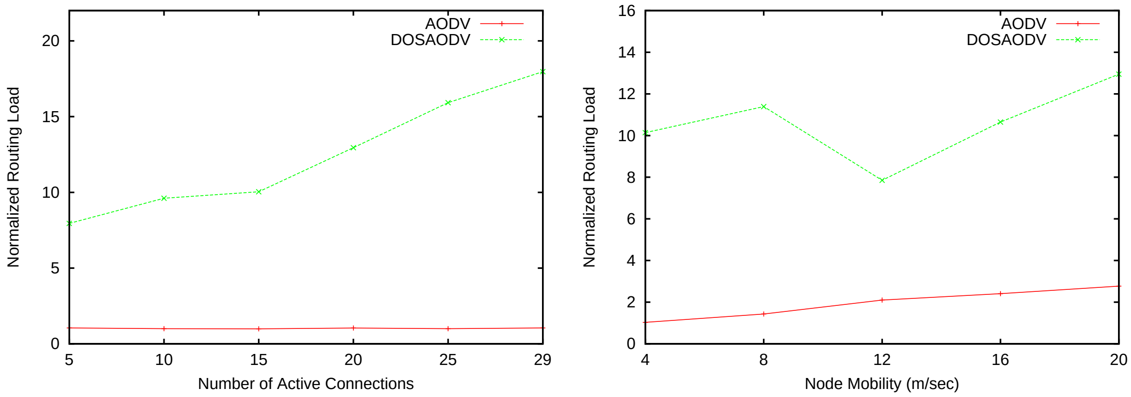
loss ratio of AODV with respect to the number of active connection is 9.5% and with respect to varying node mobility is 14.3%. Due to the effect of flooding attack, PLR is increased by 30.9 % with respect to number of active connection and 41.6% with respect to node mobility.

Table 4.6 shows the simulation results of normalized routing load for 30 nodes with varying number of active connections and varying node mobility. Figure 4.5(a) shows

Table 4.6: Normalized Routing Load

Parameter	Values	AODV	DOSAODV
Number of active connections	5	1.05	7.95
	10	1.0	9.61
	15	0.99	10.04
	20	1.04	12.95
	25	1.0	15.92
	29	1.05	17.97
Node mobility (m/s)	4	1.03	10.14
	8	1.43	11.39
	12	2.1	7.85
	16	2.41	10.65
	20	2.77	12.95

the normalized routing load with respect to varying number of active connections and 4.5(b) shows the normalized routing load with respect to varying node mobility. Average normalized routing load of AODV with respect of number of active connection is 1.02 and with respect to varying node mobility is 1.9. NRL due to effect of flooding attack is increased up to 12.4 with respect to number of active connection and 10.6 with respect to node mobility.



(a) NRL Vs number of Active connections

(b) NRL Vs node mobility

Figure 4.5: Normalized Routing Load

4.8 Evaluation of Routing Traffic Dropping Attack

Dropping routing traffic attack affects packet delivery fraction and packet drop ratio only if attacker node comes in route. Hence it is difficult to analyze effect of such attack.

Two metrics that were used in the evaluation of the flooding attack detection system are the packet delivery fraction and packet loss ratio.

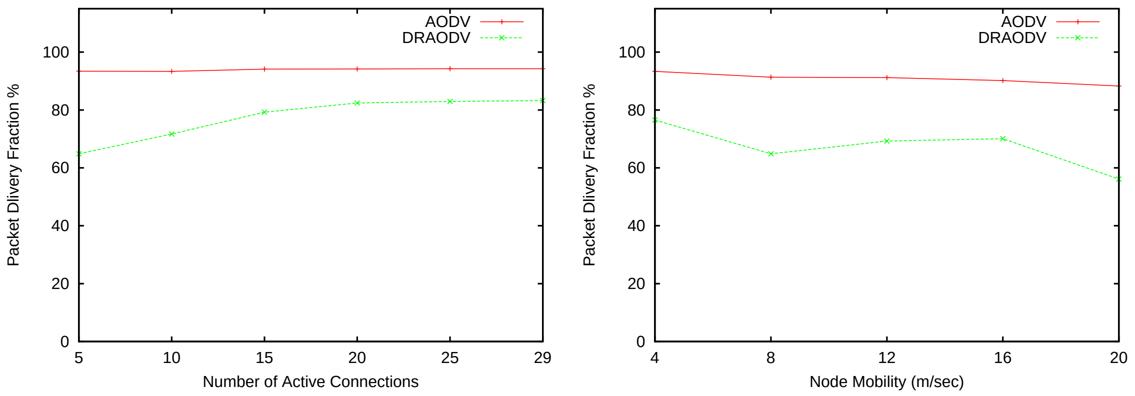
All the metrics are plotted against:

- Number of active connections
- Node mobility

Table 4.7 shows the simulation results of packet delivery fraction for 30 nodes with varying number of active connections and varying node mobility. Figure 4.6(a) shows the packet delivery fraction with respect to varying number of active connections and 4.6(b) shows the packet delivery fraction with respect to varying node mobility.

Table 4.7: Packet Delivery Fraction

Parameter	**Values**	**AODV**	**DOSAODV**
Number of active connections	5	93.41	64.87
	10	93.34	71.71
	15	94.12	79.25
	20	94.18	82.43
	25	94.26	82.96
	29	94.25	83.27
Node mobility (m/s)	4	93.34	76.54
	8	91.33	64.87
	12	91.19	69.27
	16	90.16	70.09
	20	88.29	56.13



(a) PDF Vs number of Active connections (b) PDF Vs node mobility

Figure 4.6: Packet Delivery Fraction

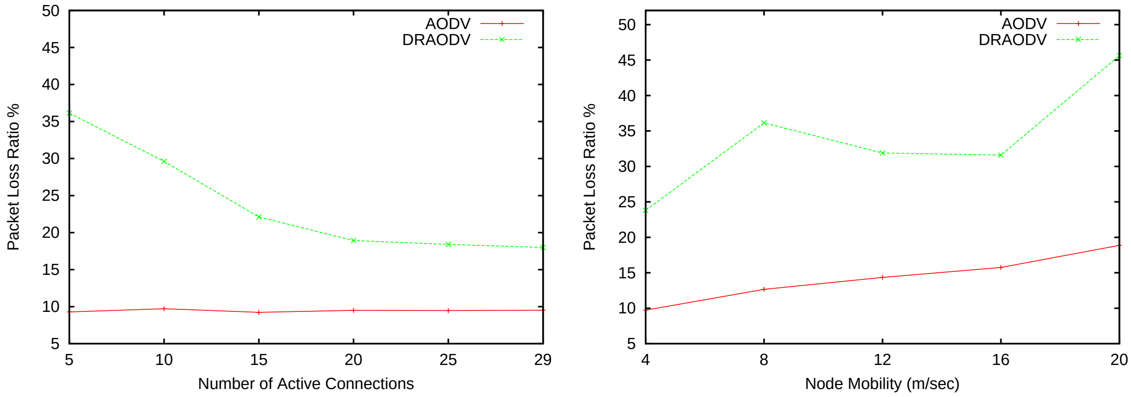
Average packet delivery fraction of AODV with respect to number of active connections is 93.92% and with respect to varying node mobility is 90.86%. Due to effect of dropping routing traffic attack, PDF is decreased by 16.5 % with respect to number of active connection and 23.5% with respect to node mobility.

Table 4.8 shows the simulation results of packet loss ratio for 30 nodes with respect to varying number of active connections and varying node mobility. Figure 4.7(a) shows

Table 4.8: Packet Loss Ratio

Parameter	**Values**	**AODV**	**DOSAODV**
Number of active connections	5	9.29	36.14
	10	9.72	29.61
	15	9.23	22.12
	20	9.5	18.93
	25	9.47	18.41
	29	9.52	17.99
Node mobility (m/s)	4	9.72	23.8
	8	12.66	36.14
	12	14.35	31.9
	16	15.75	31.6
	20	18.87	45.62

the packet loss ratio with respect to varying number of active connections and 4.7(b) shows the packet loss ratio with respect to varying node mobility. Average packet loss ratio of AODV with respect of number of active connections is 9.5% and with respect to varying node mobility, it is 14.3%. Due to effect of flooding attack,PLR is increased by 14.4 % with respect to number of active connection and 19.5% with respect to node mobility.



(a) PLR Vs number of Active connections

(b) PLR Vs node mobility

Figure 4.7: Packet Loss Ratio

4.9 Summary

The chapter presented the implementation of most prominent attacks in ad hoc network. Simulation was done using ns-2 simulator and results were than analyzed to show implications of attacks on performance of ad hoc network.

Three attacks, black hole, dropping routing packets and RREQ flooding are implemented. Results shows that individually the attacks have more or less effect on network with increase in active connections and node mobility. These attacks can lead to 20% to 80% decrease in packet delivery ratio. Black hole and dropping routing packet attacks lead to 40% to 70% packet loss while RREQ flooding attack leads to higher routing load, which highly affect the network. Above work was done to analyze degradation of performance of network due to attack, which will be further used to analyze the improvement in performance due to the detection system implemented.

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 approach proposed by Gromov et al. [22] is followed to define a Timed Finite State Machine (TFSM).

Timed FSM is 7-tuple $S = \langle S, I, O, s_0, \lambda_S, \sigma_S, \Delta_S \rangle$ where S is a finite nonempty set of states with the initial state s_0, I and O are finite disjoint input and output alphabets,$\lambda_S \subseteq S \times I \times O \times S$ is transition relation, $\sigma_S : \lambda_S \to \mathbb{N}$ is a speed function, and $\Delta_S : H \subseteq S \to S \times \mathbb{N}$ is a delay function.

If $\langle s, i, o, s' \rangle \in \lambda_S$ and $\sigma_S\,(s, i, o, s') = t$, denoted as $s \overset{i/o(t)}{\to} s'$, 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 $\Delta_S(s) = \langle s', t\rangle$, denoted as $s \xrightarrow{t} s'$, 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.

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 [21] . Figure 5.1 shows the cycle followed by intrusion detection 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

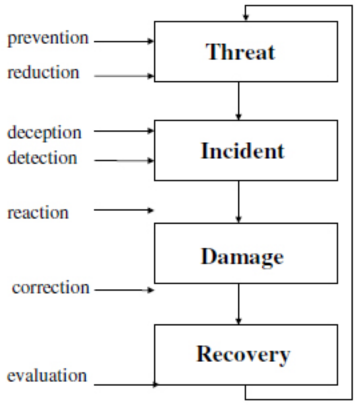


Figure 5.1: Security Incident Cycle [21]

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

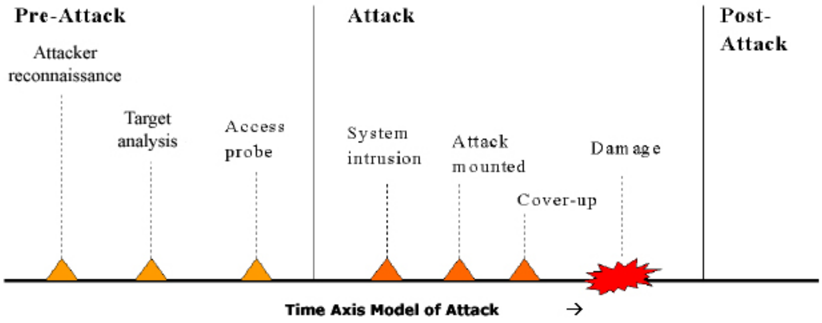


Figure 5.2: Time Axis Model of an Attack [21]

5.2 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.

5.4 Research achievements

- Stamouli et al. [9], in 2005 proposed a real time intrusion detction system which was based on TFSM. It was a knowledge based, host based intrusion detection. But due to lack of cooperation in nodes some attacks couldn't be detected.

- Hong Ding et al. [20], in 2006 has modified the above method. They have used central monitoring and cooperation function to reduce the computation process done by every node. But security of monitoring node and how to preventing a malicious node from becoming a monitor is not specified.

- Ping et al. [14] in 2006 proposed an FSM based distributed IDS for DSR but they didn't given any details about practical implementation and efficiency of the system. Ping et al. [18], in 2007 modified same algorithm but again details about practical implementation and results are not given.

- Xia Ye et al. [19], in 2009 have implemented an FSM based algorithm for AODV protocol in which to reduce the overhead network is divided into one hop region, periodically monitor node is choosen to run IDS, which will monitor the network traffic. Due to mobility the election method was to be done frequently and it increased overhead and the security of monitoring node was not considered.

5.5 Summary

In this chapter, details about Timed Finite State Machine was explained. Also the need of real time intrusion detection system was specified. Finally research achievements in the field of FSM based intrusion detection system were presented.

Chapter 6

The Proposed Algorithm

6.1 Proposed TFSM Based DIDS

Design and implementation of the system is inspired by the system proposed by Stamouli et al. [9] and Zhang et al. [17] . The system is fully distributed in nature, as I have concluded in my literature survey of part-I that distributed IDS is the best for the intrusion detection in Ad hoc networks. As in adhoc network all nodes cooperate with each other for communication and same leads to the malicious behavior. Here we will use the cooperation of nodes for detecting malicious node. Architecture of proposed algorithm is as shown in Figure 6.1 has Local traffic analysis module which will be activated on each node. Each node will capture incoming traffic. Event generation module will abstract essential information required for the local detection module to detect attack. Local detection module will trigger cooperation module for verification of its detection. After verification if it is found that the particular node is malicious response module will take appropriate action to maintain the network performance in acceptable performance measures.

Event generation module, local detection module and cooperation module are TFSM based. IDS component operates between the network traffic and routing pro-

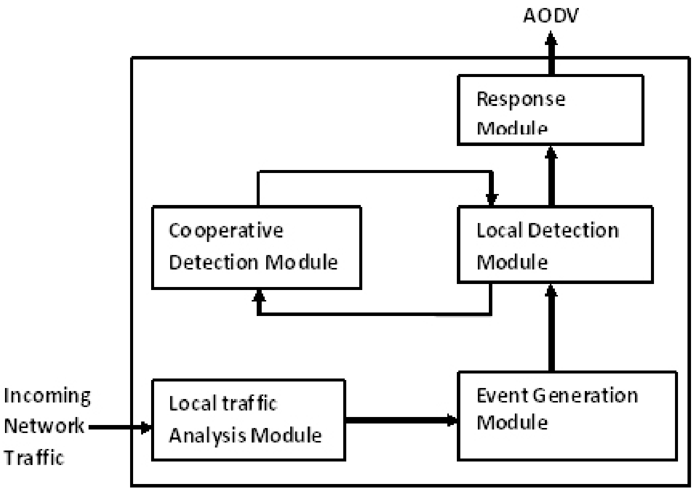


Figure 6.1: Architecture of IDS.

tocol hence no modification in underlying protocol is required. Assumptions on which our system relies:

- All links between the nodes are bidirectional.
- Interfaces have a promiscuous mode to monitor traffic of neighboring nodes.
- All nodes in network have activated IDS system except malicious node.

6.2 Working of TDIDS

As shown in Figure 6.2, TDIDS is designed to detect three different types of attacks. The figure shows the which node on which condition initiates which TFSM. To avoid looping of TFSMs, care is taken so that a node will not get into infinite loop. Sender node on receiving RREP for the packet sent by itself will initiate TFSM1 which is for black hole detection. All sender nodes will be responsible for detecting malicious nodes exhibiting black hole attack.

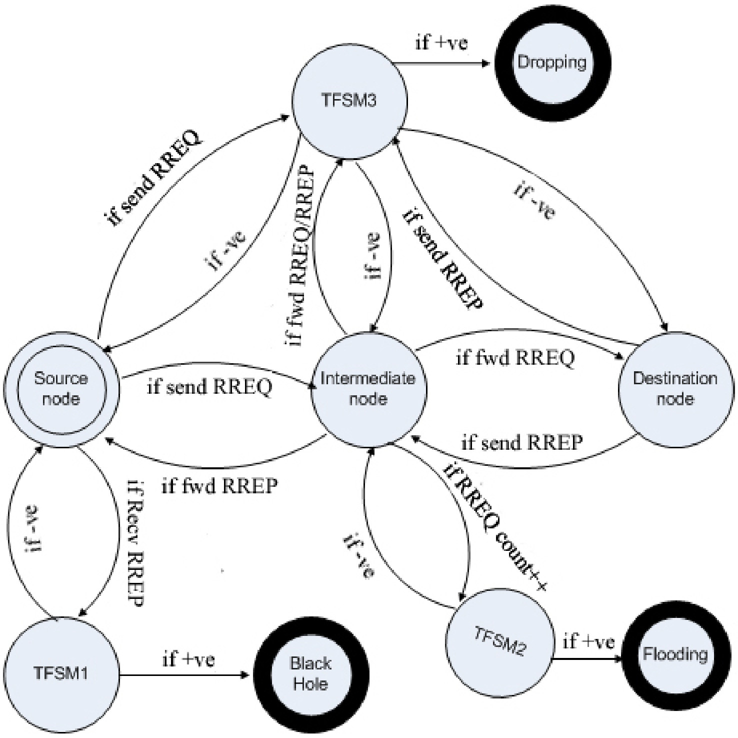


Figure 6.2: Detection Process

All intermediate node who receives RREQ from neighbors will be responsible for detecting flooding attack. As soon as the node receives any RREQ it will initiate TFSM2, which leads to Flooding attack detection.

All nodes who will send or forward routing traffic will initiate TFSM3 which leads to the detection of dropping routing traffic attack.

6.2.1 Implementation of TDIDS for Black Hole Detection

Step 1: TFSM is triggered whenever a node initiates a route discovery process and moves from init_0 state to state 1.

Step 2: If a RREP message does not arrive within the NET_TRAVERSAL_TIME defined internally in the AODV implementation the FSM goes to Time-out RESET (Tout_reset) and resets to its initial state (init_0).

Step 3: Upon the reception of the first RREP FSM moves to state 2 where it checks if the RREP_desti_seq_number is much higher than the original_seq_number included in the RREQ.

Step 4: If it is suspiciously higher it goes directly in the global detection state .

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 3 and checks for the validity of the destination sequence number and similarly decides whether to move to a global detection state.

Step 7: 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 knows that the information in the RREP is forged. Now node will decide the prevention measures i.e if node does not receive reply it will not update the routing table with the invalid routing information.

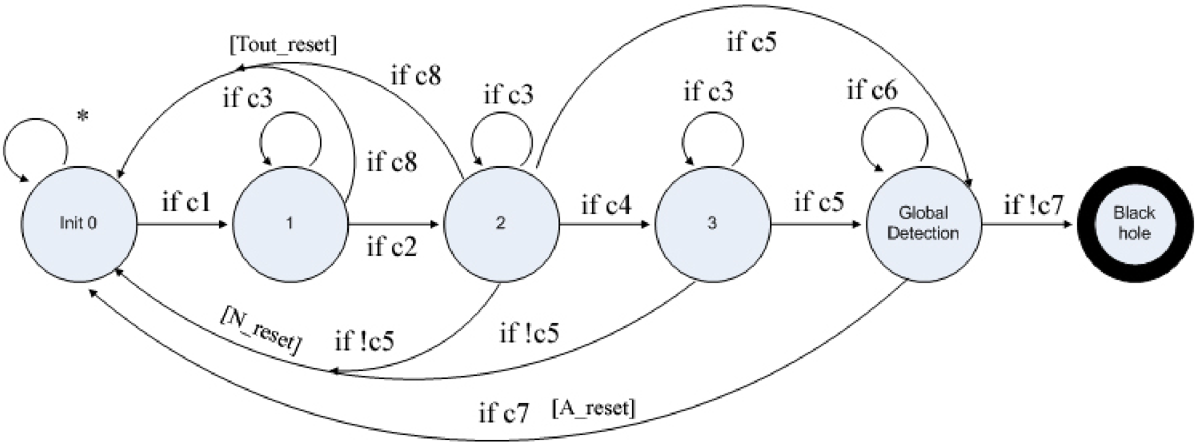


Figure 6.3: TFSM for Blackhole Detection

Condition	Value
c1	if send Request
c2	if receive Reply
c3	if T_recv > CURRENT_TIME
c4	if receive 2..n Reply
c5	if RREP_dest_seqno > orig_dest_seqno
c6	if T_alarm > CURRENT_TIME
c7	if receive prealarm Reply
c8	if T > NET_TRAVERSAL_TIME

Table 6.1: Conditions

6.2.2 Flooding Attack Detection System

TFSM used for flooding attack detection is shown in Figure 6.4. Initially the FSM is at state init_0, as soon as node receives RREQ, FSM moves to state 1. Here we maintain a table which contain the node_id and analysis_time of the packet. So as soon as node receive RREQ it adds the senders id and time at which node received RREQ. A RREQ counter will be maintained for each sender node till time interval *t*. Before analysis_time gets expired if count become greater than threshold which is equal to RREQ_RATELIMIT then the node will be detected as malicious one.

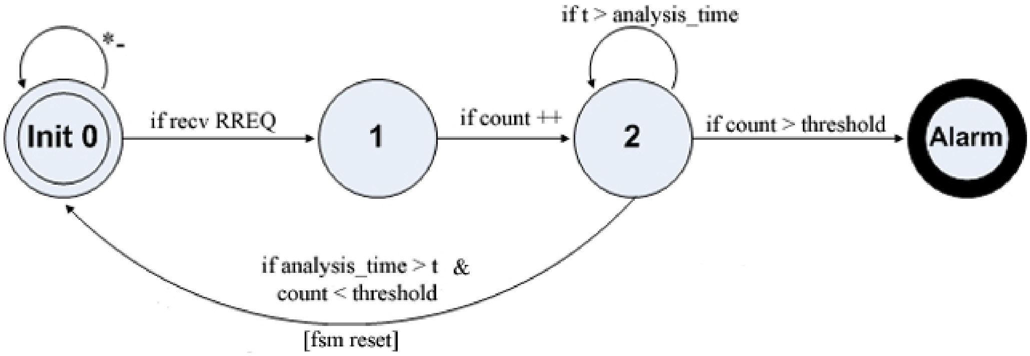


Figure 6.4: TFSM of Flooding Attack Detection System

Now the prevention module is activated. We use node suppression method for preventing flooding attack. After detection of malicious node, all neighbor nodes will not accept any kind of traffic from that particular node. It means a malicious node will be isolated from the network.

But it may happened that the node is falsely identified as a malicious node hence we have taken care for that condition also. The malicious node will be ignored for specific interval of time so that effect of false detection can be minimized.

6.2.3 Dropping Routing Traffic Attack Detection system

In In chapter 4, effect of dropping routing traffic was analyzed. TFSM for dropping routing traffic attack detection system is shown in Figure 6.5. For detection system, it is assumed that AODV is in promicious mode and each node can overhear the traffic of neighboring node. At initial position FSM is at init_0 state. As soon as the node sends or forwards any routing traffic (RREQ or RREP) it will add all recipients who are not the destinations of the packet to the analysis_table. Now node will wait till t_fwd time, if time expires and node has not forwarded that packet or sent reply the FSM moves to global detection state and set pre alarm true. Here detecting node will send Alarm_inquiry_request packet and wait till t_reply time. If detecting node receives pre_alarmreply, FSM gets pre alarm reset and goes to initial position else node is detected as malicious.

Now prevention system will send RERR message to all neighboring node and the malicious node will be isolated from the network for some specific time interval.

Number of times the node is detected as malicious, isolation time will be incremented accordingly.

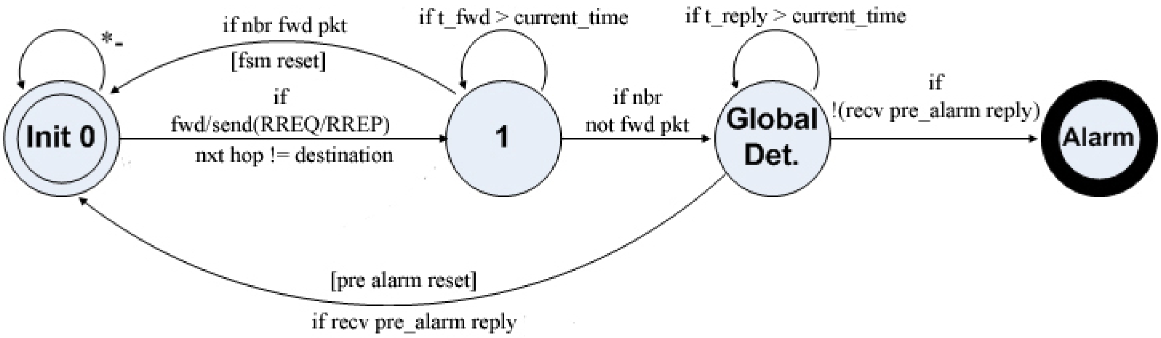


Figure 6.5: TFSM of Dropping Routing Traffic Attack Detection system

6.3 Summary

Based on the previous chapter a timed finite sate machine based distributed intrusion detection system is proposed and description of individual FSM is presented in this chapter.

Chapter 7

Testing and 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.

Table 7.1: Simulation Parameters

Simulator	ns-2.34
Simulation duration	1000 sec
Simulation area	1000 * 1000 m
Number of Nodes	30
Transmission range	250 m
Movement model	Random waypoint
Maximum speed	5–20 m/sec
Traffic type	CBR (UDP)
Data payload	512 bytes
Number of malicious nodes	5%
Number of active connection	3–29

7.2 Evaluation of Black Hole Attack Detection

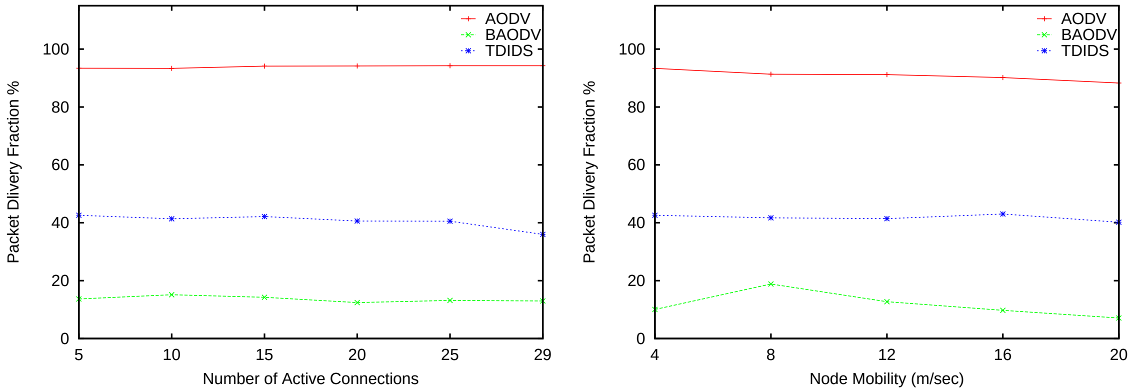
Two metrics that were used in the evaluation of the black hole attack detection system are the packet delivery fraction and packet loss ratio. Both the metrics are plotted against number of active connections and node mobility.

Table 7.2 shows the simulation results of packet delivery fraction for 30 nodes with varying number of active connections and varying node mobility.

Table 7.2: Packet Delivery Fraction

Parameter	**Values**	**AODV**	**BHAODV**	**TDIDS**
Number of active connections	5	93.41	13.67	42.58
	10	93.34	15.12	41.37
	15	94.12	14.24	42.14
	20	94.18	12.42	40.59
	25	94.26	13.17	40.53
	29	94.25	12.97	35.97
Node mobility (m/s)	4	93.34	10.04	42.58
	8	91.33	18.84	41.71
	12	91.19	12.73	41.43
	16	90.16	9.74	43.03
	20	88.29	7.09	40.18

Figure 7.1(a) shows the packet delivery fraction with respect to varying number of active connections and 7.1(b) shows the packet delivery fraction with respect to varying node mobility. Average packet delivery fraction of AODV with respect of number of active connection is 93.92% and with varying node mobility is 90.86%. Due to effect of blackhole attack, PDF is decreased by 80.33% with respect to number of active connection and 79.17% with respect to node mobility. Where as detection system gives 91.7% true positive results and prevention system is able to improve PDF upto 26.93% with respect to number of active connections and give 86% true positive results with 30.1% improvement with respect to node mobility.



(a) PDF Vs number of Active connections

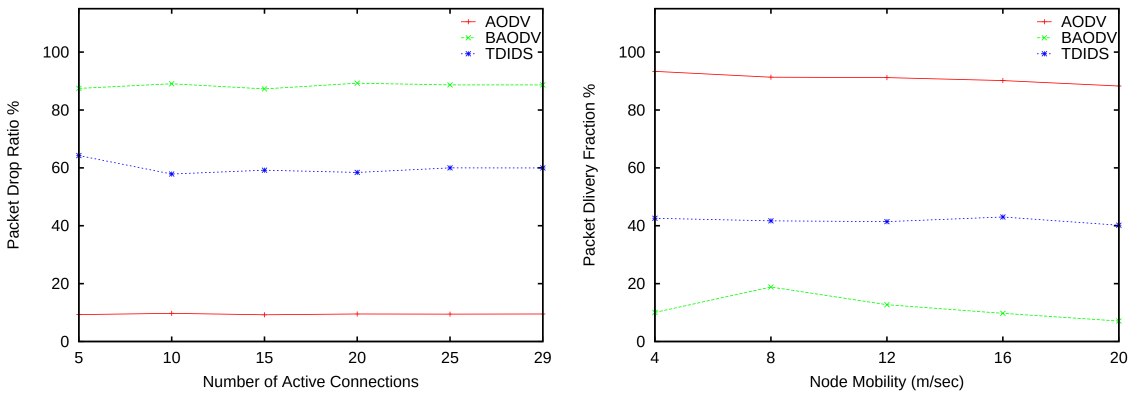
(b) PDF Vs node mobility

Figure 7.1: Packet Delivery Fraction

Table 7.3 shows the simulation results of packet loss ratio for 30 nodes with varying number of active connections and varying node mobility. Figure 7.2(a) shows the packet loss ratio with respect to varying number of active connections and 7.2(b) shows the packet loss ratio with respect to varying node mobility. Average packet loss ratio of AODV with respect of number of active connection is 9.5% and with varying node mobility is 14.3%. Due to effect of blackhole attack, PLR is increased

Table 7.3: Packet Loss Ratio

Parameter	**Values**	**AODV**	**BHAODV**	**TDIDS**
Number of active connections	5	9.29	87.45	64.27
	10	9.72	89.07	57.86
	15	9.23	87.3	59.2
	20	9.5	89.25	58.44
	25	9.47	88.68	60.01
	29	9.52	88.68	59.98
Node mobility (m/s)	4	9.72	90.98	57.86
	8	12.66	83.14	59.01
	12	14.35	90.26	59.72
	16	15.75	93.81	58.3
	20	18.87	96.89	61.71



(a) PLR Vs number of Active connections

(b) PLR Vs node mobility

Figure 7.2: Packet Loss Ratio

by 78.95 % with respect to number of active connections and 76.75% with respect to node mobility. Here the detection and prevention system is able to reduce PLR upto 28.45% with respect to number of active connection and 31.7% reduction with respect to node mobility.

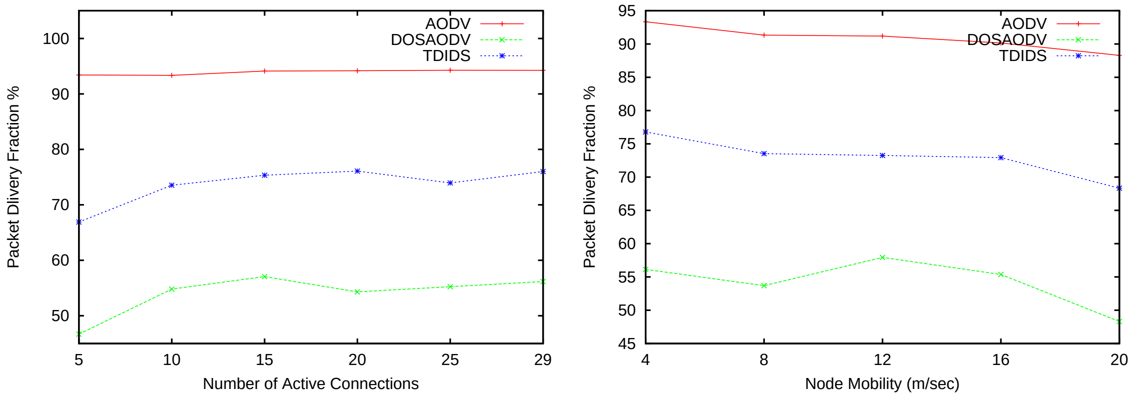
7.3 Evaluation of Flooding Attack Detection System

Three metrics that were used in the evaluation of the flooding attack detection system are the packet delivery fraction, packet loss ratio and normalized routing load. All metrics are plotted against the number of active connections and node mobility.

Table 7.4 shows the simulation results of packet delivery fraction for 30 nodes with varying number of active connections and varying node mobility. Figure 7.3(a) shows the packet delivery fraction with respect to varying number of active connections and 7.3(b) shows the packet delivery fraction with respect to varying node mobility. Average packet delivery fraction of AODV with respect of number of active connection is 93.92% and with varying node mobility is 90.86%. Due to effect of flooding attack, PDF is decreased by 39.8 % with respect to number of active connection and 36.6% with respect to node mobility. The detection system gives 92.3% true positive results and prevention system is able to improve PDF upto 19.6% with respect to number of active connections and gives 86.6% true positive results with 18.7% improvement

Table 7.4: Packet Delivery Fraction

Parameter	**Values**	**AODV**	**DOSAODV**	**TDIDS**
Number of active connections	5	93.41	46.69	66.9
	10	93.34	54.81	73.53
	15	94.12	57.06	75.34
	20	94.18	54.31	76.09
	25	94.26	55.25	73.96
	29	94.25	56.16	76
Node mobility (m/s)	4	93.34	56.13	76.79
	8	91.33	53.69	73.53
	12	91.19	57.93	73.25
	16	90.16	55.37	72.93
	20	88.29	48.31	68.33



(a) PDF Vs number of Active connections

(b) PDF Vs node mobility

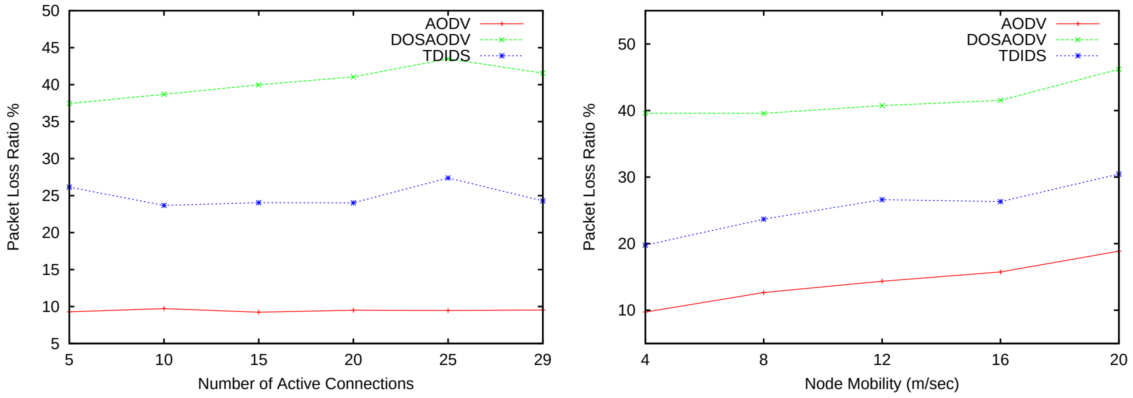
Figure 7.3: Packet Delivery Fraction

with respect to node mobility.

Table 7.5 shows the simulation results of packet loss ratio for 30 nodes with varying number of active connections and varying node mobility. Figure 7.4(a) shows the packet loss ratio with respect to varying number of active connections and 7.4(b) shows the packet loss ratio with respect to varying node mobility. Average packet loss ratio of AODV with respect of number of active connection is 9.5% and with

Table 7.5: Packet Loss Ratio

Parameter	**Values**	**AODV**	**DOSAODV**	**TDIDS**
Number of active connections	5	9.29	37.45	26.14
	10	9.72	38.69	23.68
	15	9.23	39.98	24.04
	20	9.50	41.05	24.02
	25	9.47	43.50	27.39
	29	9.52	41.56	24.29
Node mobility (m/s)	4	9.72	39.59	19.77
	8	12.66	39.58	23.68
	12	14.35	40.74	26.62
	16	15.75	41.53	26.32
	20	18.87	46.20	30.45



(a) PLR Vs number of Active connections

(b) PLR Vs node mobility

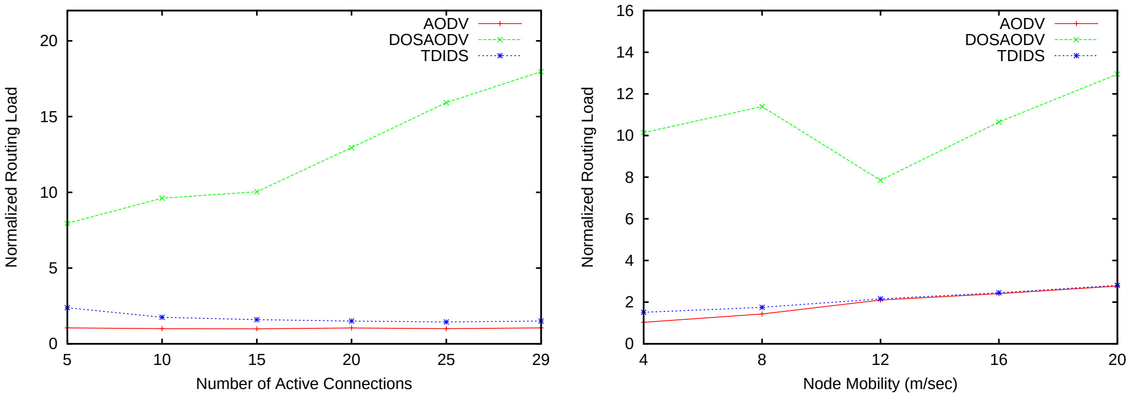
Figure 7.4: Packet Loss Ratio

varying node mobility is 14.3%. Due to effect of flooding attack, PLR is increased by 30.9 % with respect to number of active connection and 41.6% with respect to node mobility. The detection and prevention system is able to reduce PLR upto 15.5% with respect to number of active connection as well as 16.2% reduction with respect to node mobility.

Table 7.6 shows the simulation results of normalized routing load for 30 nodes with varying number of active connections and varying node mobility. Figure 7.5(a) shows the normalize routing load with respect to varying number of active connections and 7.5(b) shows the normalized routing load with respect to varying node mobility. Average normalized routing load of AODV with respect of number of active connection is 1.02 and with varying node mobility is 1.9 . Due to effect of flooding attack, NRL is increased up to 12.4 with respect to number of active connection and 10.6 with respect to node mobility. The detection and prevention system is able to maintain NRL upto 1.7 with respect to number of active connection and 2.1 with respect to node mobility which is slightly higher than AODV.

Table 7.6: Normalized Routing Load

Parameter	**Values**	**AODV**	**DOSAODV**	**TDIDS**
Number of active connections	5	1.05	7.95	2.38
	10	1.0	9.61	1.75
	15	0.99	10.04	1.59
	20	1.04	12.95	1.5
	25	1.0	15.92	1.44
	29	1.05	17.97	1.5
Node mobility (m/s)	4	1.03	10.14	1.51
	8	1.43	11.39	1.75
	12	2.1	7.85	2.16
	16	2.41	10.65	2.45
	20	2.77	12.95	2.81



(a) NRL Vs number of Active connections

(b) NRL Vs node mobility

Figure 7.5: Normalized Routing Load

7.4 Evaluation of Dropping Routing Traffic Attack Detection System

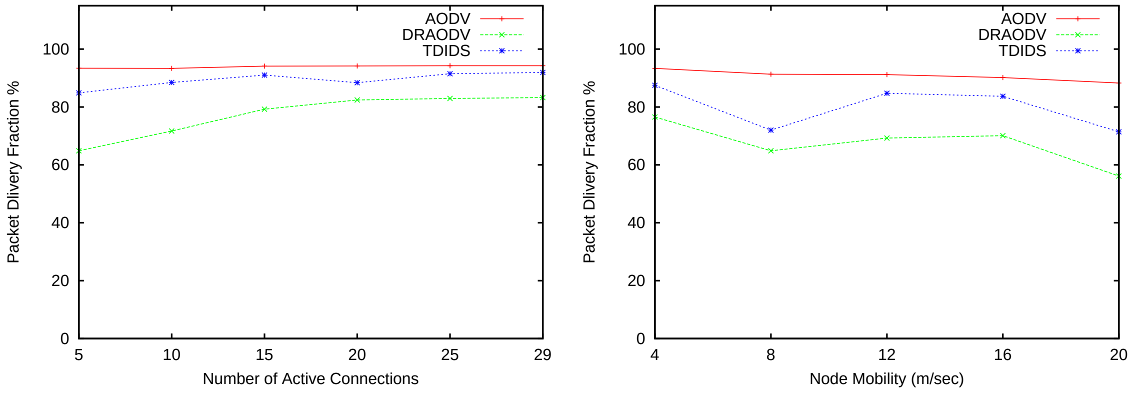
Two metrics that were used in the evaluation of the flooding attack detection system are the packet delivery fraction and packet loss ratio. Both the metrics are plotted against number of active connections and node mobility.

Table 7.7 shows the simulation results of packet delivery fraction for 30 nodes with varying number of active connections and varying node mobility.

Table 7.7: Packet Delivery Fraction

Parameter	**Values**	**AODV**	**DOSAODV**	**TDIDS**
Number of active connections	5	93.41	64.87	84.88
	10	93.34	71.71	88.49
	15	94.12	79.25	91.03
	20	94.18	82.43	88.41
	25	94.26	82.96	91.49
	29	94.25	83.27	91.94
Node mobility (m/s)	4	93.34	76.54	87.49
	8	91.33	64.87	72.01
	12	91.19	69.27	84.76
	16	90.16	70.09	83.71
	20	88.29	56.13	71.43

Figure 7.6(a) shows the packet delivery fraction with Routing Traffic varying number of active connections and 7.6(b) shows the packet delivery fraction with Routing Traffic varying node mobility. Average packet delivery fraction of AODV with respect to number of active connection is 93.92% and with varying node mobility is 90.86%. Due to effect of dropping routing traffic attack, PDF is decreased by 16.5 % with respect to number of active connection and 23.5% with respect to node mobility. The detection system gives 80.1% true positive results and prevention system is able to improve PDF upto 12% with respect to number of active connections and 77.3% true



(a) PDF Vs number of Active connections

(b) PDF Vs node mobility

Figure 7.6: Packet Delivery Fraction

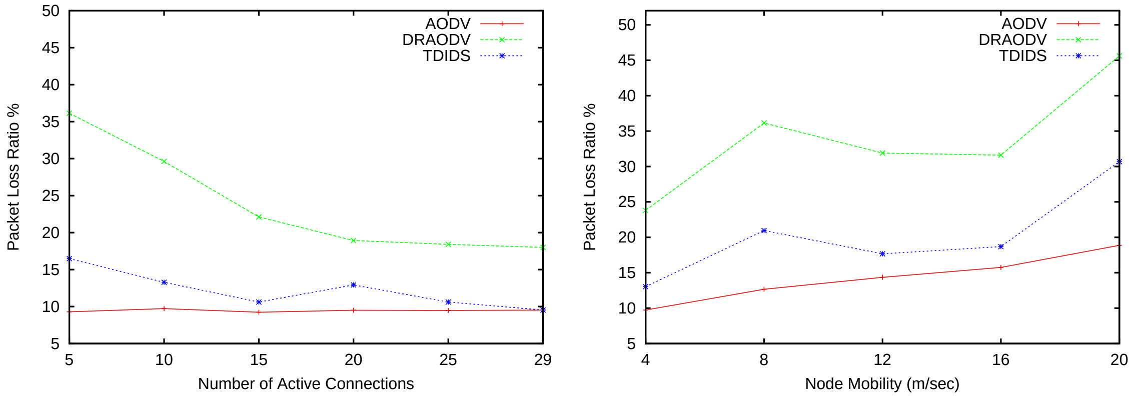
positive results with 12.5% improvement with respect to node mobility.

Table 7.8 shows the simulation results of packet loss ratio for 30 nodes with varying number of active connections and varying node mobility.

Table 7.8: Packet Loss Ratio

Parameter	Values	AODV	DOSAODV	TDIDS
Number of active connections	5	9.29	36.14	16.48
	10	9.72	29.61	13.28
	15	9.23	22.12	10.61
	20	9.5	18.93	12.92
	25	9.47	18.41	10.61
	29	9.52	17.99	9.52
Node mobility (m/s)	4	9.72	23.8	13.03
	8	12.66	36.14	20.96
	12	14.35	31.9	17.67
	16	15.75	31.6	18.68
	20	18.87	45.62	30.69

Figure 7.7(a) shows the packet loss ratio with Routing Traffic varying number of active connections and 7.7(b) shows the packet loss ratio with Routing Traffic varying node mobility. Average packet loss ratio of AODV with respect of number of active



(a) PLR Vs number of Active connections

(b) PLR Vs node mobility

Figure 7.7: Packet Loss Ratio

connection is 9.5% and with varying node mobility is 14.3%. Due to effect of flooding attack, PLR is increased by 14.4 % with respect to number of active connection and 19.5% with respect to node mobility. The detection and prevention system is able to reduce PLR upto 11.6% with respect to number of active connection and 12% reduction with respect to node mobility.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

TDIDS is designed for the detection of three different types of attacks in adhoc network. TDIDS analyzes real time traffic instead of analyzing logs. Its prevention system is able to minimize the damage done by the attacks by active detection of attacks and maintenance of network performance in acceptable range. The detection system has high accuracy of detection.

- The proposed system gives average 27% improvement in packet delivery ratio with variable number of active connections and 25.08% improvement with respect to node mobility when network is attacked by black hole attack. Due to distributed detection system, it gives 91.7 % true positive results with variable number of active connections and 86 % true positive results with variable mobility.

- Detection system gives 92.3% true positive results and prevention system is able to improve PDF upto 19.6% with respect to number of active connections and 86.6% true positive results with 18.7% improvement with respect to node mobility against flooding attacks. Also the system is able to maintain routing load near about AODV.

- Detection system gives 80.1% true positive results and prevention system is able to improve PDF upto 12% with respect to number of active connection as well as give 77.3% true positive results with 12.5% improvement with respect to node mobility against dropping routing traffic attack.

8.2 Future Work

- To extend the TDIDS to provide security from more attacks.
- To provide some cryptographical mechanism for preventing impersonation of node.
- To test TDIDS system with variable bit rate traffic to analyze effect of system with real time environment.

Appendix A

List of publication

- Sumitra Menaria, Sharada Valiveti, K. Kotecha, "Comparative study of Distributed Intrusion Detection in Ad-hoc Networks", *International Journal of Computer Applications (ISSN no: 0975-8887), 8(9):1116, October 2010.* Available at `http://www.ijcaonline.org/archives/volume8/number9/1237-1699`

- Sumitra Menaria, Sharada Valiveti, K. Kotecha, "Attack Generation and its Implications on Ad Hoc Networks", *International Conference on Emerging Trends in Networks and Computer Communications -(ETNCC-2011), April 2011*, (IEEE Record Number 18690)

Web References

[1] http://www.winlab.rutgers.edu/~zhibinwu/html/network_simulator_2.html

[2] http://www.isi.edu/nsnam/ns/tutorial/

[3] http://csis.bits-pilani.ac.in/faculty/murali/resources/tutorials/ns2.htm

[4] http://nslab.ee.ntu.edu.tw/courses/ns-tutorial/labs/lab5.html

[5] http://nile.wpi.edu/NS/

[6] http://tools.ietf.org/html/rfc3561

[7] http://www.awktutorial.com/

References

[1] 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 & Communication Technology, IEEE,*2010.

[2] Kwan-Wu Chin, “The Behavior of MANET Routing Protocols in Realistic Environments”, *Asia-Pacific Conference on Communications, Perth, Western Australia, @IEEE* October, 2005.

[3] Ali Ghaffari, “Vulnerability and Security of Mobile Ad hoc Networks”, *Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization, Lisbon, Portugal,* September, 2006.

[4] Karan Singh, R. S. Yadav, Ranvijay, “A Review Paper On Ad Hoc Networks Security”, *International Journal of Computer Science and Security, Volume (1)*

[5] Adam Burg, “Ad hoc network specific attacks”, *IEEE workshop on Security and Assurance in Ad-Hoc Networks,* 2003.

[6] Giovanni Vigna, Sumit Gwalani, Kavitha Srinivasan, Elizabeth M., Belding-Royer Richard, A. Kemmerer, “An Intrusion Detection Tool for AODV-based Ad hoc Wireless Networks ”, *IEEE InternationalConference on Computer and Information Technology,* 2000.

[7] Tiranuch Anantvalee, Jie Wu, “A Survey on Intrusion Detection in Mobile Ad Hoc Networks,” *Springer,* 2006.

[8] Satria Mandala, Md. Asri Ngadi, A.Hanan Abdullah, “A Survey on MANET Intrusion Detection”, *in: International Journal of Computer Science and Security, Volume (2) : Issue (1),*

[9] 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.

[10] Yian Huang, Wenke Lee, "A 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

[11] O. Kachirski and R. Guha, "Effective Intrusion Detection Using Multi- ple Sensors in Wireless Ad Hoc Networks", *Proceedings of the 36th An- nual Hawaii International Conference on System Sciences,* January 2003.

[12] AikateriniMitrokotsa, Rosa Mavropodi, Christos Douligeris, "Intrusion Detection of Packet Dropping Attacks inMobile Ad Hoc Network", *in Proceedings TAyia Napa, Cyprus,* 2006.

[13] R. Nakkeeran, T. Aruldoss Albert and R.Ezumalai, "Agent Based Eficient Anomaly Intrusion Detection System in Adhoc networks", *IACSIT International Journal of En- gineering and Technology Vol. 2, No.1, February,* 2010.

[14] Ping Yi, Yichuan Jiang, Yiping Zhong, Shiyong Zhang, "Distributed Intrusion Detection for Mobile Ad Hoc Networks", *In Proceeding IEEE Symposium on Applications and the Internet Workshops SAINT-W05,* 2005.

[15] Stuart Kurkowski, Tracy Camp, Michael Colagrosso, "MANET Simulation Studies: The Incredibles", *Mobile Computing and Communications Review, Volume 9, Number 4.*

[16] Sampathkumar Veeraraghavan, S. Bose, K. Anand and A. Kannan, "Intelligent Agent Based Approach for Intrusion Detection and Prevention in Adhoc Networks", *IEEE - ICSCN 2007, MIT Campus, AnnaUniversity, Chennai, India. Feb. 22-24, 2007.*

[17] Yongguang Zhan, Wenke Lee, Yi-An Huang, "Intrusion Detection Techniques for Mobile Wireless Networks", *ACM WINET journal 2002*

[18] Ping YI, Vue WU ,Jianhua LI, "Malicious Node Detection in Ad Hoc Networks Using Timed Automata", *Proceedings of IEEE Symposium on Wireless, Mobile and Sensor Networks, 2007. (CCWMSN07)*

[19] Xia Ye, Junshan Li, "An FSM-based Automatic Detection in AODV for Ad Hoc network", *Proceedings of IEEE Symposium on Computer Network and Multimedia Technology, 2009.*

[20] Hong Ding, Xiaomei Xu, "Real-time Cooperation Intrusion Detection System for MANets", *Proceedings of IEEE International conferance on Wireless, Mobile and Multimedia Networks, 2006*

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

[22] M.Gromov, D.Popov, N.Yevtushenko, “Deriving Test Suites for Timed Finite State Machines”, *Proceedings of IEEE EastWest Design & Test Symposium (EWDTS’08)*

Index