**INTRUSION DETECTION TECHNIQUE IN**

**MOBILE ADHOC NETWORKS**

***PROJECT REPORT***

*Submitted by*

**C.DINESH(810015104304)**

**S.IRULAPPAN(810015104305)**

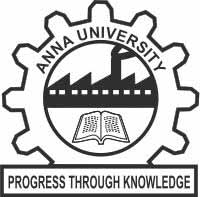
***In partial fulfilment for the award of the degree***

***Of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**UNIVERSITY COLLEGE OF ENGINEERING AND TECHNOLOGY,**

**BIT CAMPUS**

**TIRUCHIRAPPALLI-620 024**

**ANNA UNIVERSITY: CHENNAI 600025**

**APRIL 2019**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**INTRUSION DETECTION TECHNIQUE IN MOBILE ADHOC NETWORKS***”*, is a bonafide work of **Mr. C.DINESH (810015104304)** and **Mr. S.IRULAPPAN (810015104305)** whose carried out the project work under my supervision.

SIGNATURESIGNATURE

**Mr. D. VENKATESAN** **Mr .M. KRISHNAKUMAR**

HEAD OF THE DEPARTMENTDepartment Of CSE/IT

University College Of Engineering, BIT Campus,Trichy-600024.

Submitted for the Project Viva-Voice Examination held on …………….

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**DECLARATION**

We hereby declare that the work entitled **“INTRUSION DETECTION TECHNIQUE IN MOBILE ADHOC NETWORKS”** is submitted in partial fulfilment of the requirement for the award of the degree in B.E,University College Of Engineering(BIT Campus), Tiruchirappalli, is a record of the our original work carried out by us during the academic year 2018-2019. Under the supervision and guidance of Mr.M.Krishnakumar, Teaching Fellow, University College Of Engineering(BIT Campus) , Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other Degree, either in this or any other university.

Signature Of The Candidates

C.DINESH S.IRULAPPAN

810015104304 810015104305

I certify that the declaration made above by the candidate is true.

Signature of the Guide

Mr.M.KRISHNAKUMAR

Teching Fellow

Department of CSE/IT

Anna University

Tiruchirappalli.

**ACKNOWLEDGEMENT**

First and foremost we would like to thank the God for giving us the power to believe in our self and pursue our dreams.

I would like to convey my heartfelt thanks to our honorable Dean **Dr. T. SENTHILKUMAR,** Associate Professor for having provided me with all required facilities to complete my project without hurdles.

I would like to express my sincere thanks and deep sense of gratitude to guide **Mr. D. VENKATESAN,** Assistant Professor and Head, Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of this project work.

It is our responsibility to thank our project coordinator **Mr. C. SANKAR RAM,** Assistant Professor, Department of Computer science and Engineering for his constant inspiration that he has all through the project period.

I would like to thank **Mr. C. SURESH KUMAR,** Teaching Fellow, Department of Computer Science and Engineering, for his encouragement for this work.We wish to express our profound thanks with gratitude to **Mr M. Krishnakumar**, Teaching Fellow, Department of CSE/IT, for providing us the opportunity to do this project.

We wish to express our gratitude to the officials and other staff members of our department who rendered their help during the period of our project work.

**ABSTRACT**

Ad-hoc network is a temporary self-organizing network that needs no fixed infrastructure. So it has been applied extensively in many areas requesting temporary communication such as military field, emergency disaster relief and road traffic. While, due to the feature of self-organization and wireless communication channels, ad-hoc network is more vulnerable to various attacks compared to the traditional network. Mobile Ad hoc Networks use cooperative routing algorithms such as Ad hoc On demand Distance Vector (AODV) routing protocol that have been designed to efficiently reroute traffic when confronted with network congestion. However the design of these routing protocol of MANETs assume that there is no malicious intruder node in the network. But MANETs are vulnerable to various types of attacks at all layers. In this paper, we introduce a prediction algorithm which we name as Trust Level prediction algorithm for the prediction of malicious node in the network. This algorithm uses a mathematical prediction technique called linear prediction Algorithm to identify the malicious node in the network. A comparison of proposed protection mechanism will be included in our paper. Finally, we identify areas where further research could focus.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **TABLE OF CONTENTS** | |  |
| **CHAPTER NO** | **TITLE** | | **PAGE NO** |
|  | **ABSTRACT** | | 5 |
|  | **LIST OF FIGURES** | | 9 |
|  | **LIST OF ABBREVATIONS** | | 10 |
| **1** | **INTRODUCTION** | | **11** |
|  | **1.1** | OVERVIEW | 11 |
|  | **1.2** | CHARACTERISTICS OF NETWORK | 13 |
|  | **1.3** | MOBILE ADHOC NETWORK | 17 |
|  | **1.4** | ATTACKS IN MANETS | 18 |
|  | **1.5** | INTRUSION DETECTION SYSTEM APPROACHES | 21 |
|  | **1.6** | METHODS OF INTRUSION DETECTION SYSTEM | 23 |
| **2** | **LITERATURE SUREVEY** | | **24** |
|  | **2.1** | INTRUSION DETECTION IN WIRELESS AD HOC NETWORKS | **24** |
|  | **2.2** | ADAPTIVE INTRUSION DETECTION AND PREVENTION OF DENIAL OF SERVICE ATTACKS | **24** |
|  | **2.3** | A SURVEY OF MANET INTRUSION DETECTION AND PREVENTION APPROACHES FOR NETWORK LAYER ATTACKS | **25** |
|  | **2.4** | LIGHT WEIGHT SYBIL ATTACK DETECTION IN MANETS | **26** |
| **3** | **SYSTEM REQUIREMENTS** | | **27** |
|  | **3.1** | HARDWARE REQUIREMENTS | 27 |
|  | **3.2** | SOFTWARE REQUIREMENTS | 27 |
|  | **3.3** | SOFTWARE ENVIRONMENT | 28 |
| **4** | **SYSTEM ANALYSIS** | | **30** |
|  | **4.1** | EXISTING SYSTEM | 30 |
|  | **4.2** | PROPOSED SYSTEM | 31 |
|  | **4.3** | PROPOSED SYSTEM ARCHITECTURE | 34 |
| **5** |  | **IMPLEMENTATION** | **35** |
|  | **5.1** | MODULES | 35 |
|  | **5.2** | MODULE DESCRIPTION | 35 |
|  | **5.2.1** | MANET ENVIRONMENT ESTABLISHMENT | 35 |
|  | **5.2.2** | PACKET DROP SCENARIO IMPLEMENTATION | 35 |
|  | **5.2.3** | IMPLEMENTATION OF TRUST LEVEL PREDICTION ALGORITHM | 35 |
|  | **5.2.4** | AUDIT DATA MAINTENANCE | 36 | |
| **6** |  | **SOFTWARE DESCRIPTION** | **37** |
| **7** |  | **IMPLEMENTATION AND RESULTS** | **42** |
|  | **7.1** | SCREEN SHOTS | 43 |
| **8** |  | **CONCLUSION** | **51** |
|  |  | **APPENDIX** | **52** |
|  |  | **REFERENCES** | **61** |

**LIST OF FIGURES**

**FIGURE NO NAME OF FIGURES PAGE NO**

3.2.1 PROPOSED ARCHITECTURE 18

5.2.1 NS2 ARCHITECTURE 25

6.1.1 TOPOLOGY FORMATION 29

6.1.2 PACKET TRANMISSION 30

6.1.3 PACKET DROPING 31

6.1.4 MALICIOUS NODES 32

6.1.5 TRACE FILE 33

6.1.6 EXCEL SHEET - THRESHOLD 34

6.1.7 SYSTEM OUTPUT 35

**LIST OF ABBREVITIONS**

**MANET** Mobile Adhoc NETwork

**IDS**  Intrusion detection System

**ABID**  Anomaly Based Intrusion Detection

**KBID**  Knowledge Based Intrusion Detection

**SBID** Specification Based Intrusion Detection

**GIDP** Generalized Intrusion Detection and Prevention

**DSR** Dynamic Source Routing

**RREQ** Route Request

**RREP**  Route Replay

**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

**What is networking?**

Networking, also known as computer networking, is the practice of transporting and exchanging data between nodes over a shared medium in an information system. Networking comprises not only the design, construction and use of a network, but also the management, maintenance and operation of the network infrastructure, software and policies.

**How networking works?**

**General Network Techniques** - When computers communicate on a network, they send out data packets without knowing if anyone is listening. Computers in a network all have a connection to the network and that is called to be connected to a network bus. What one computer sends out will reach all the other computers on the local network.

 A computer network is simply a collection of computer equipment that's connected with wires, [optical fibers](https://www.explainthatstuff.com/fiberoptics.html), or wireless links so the various separate devices (known as **nodes**) can "talk" to one another and swap **data** (computerized information).

For the different computers to be able to distinguish between each other, every computer has a unique ID called MAC-address (Media Access Control Address). This address is not only unique on your network but unique for all devices that can be hooked up to a network. The MAC-address is tied to the hardware and has nothing to do with IP-addresses. Since all computers on the network receives everything that is sent out from all other computers the MAC-addresses is primarily used by the computers to filter out incoming network traffic that is addressed to the individual computer.

When a computer communicates with another computer on the network, it sends out both the other computers MAC-address and the MAC-address of its own. In that way the receiving computer will not only recognize that this packet is for me but also, who sent this data packet so a return response can be sent to the sender.

**On an Ethernet network** as described here, all computers hear all network traffic since they are connected to the same bus. This network structure is called multi-drop.

One problem with this network structure is that when you have, let say ten (10) computers on a network and they communicate frequently and due to that they sends out there data packets randomly, collisions occur when two or more computers sends data at the same time. When that happens data gets corrupted and has to be resent. On a network that is heavy loaded even the resent packets collide with other packets and have to be resent again. In reality this soon becomes a bandwidth problem. If several computers communicate with each other at high speed they may not be able to utilize more than 25% of the total network bandwidth since the rest of the bandwidth is used for resending previously corrupted packets. The way to minimize this problem is to use network switches.

**1.2 CHARACTERISTICS OF NETWORK**

The following characteristics should be considered in network design and ongoing maintenance:

* **Availability** is typically measured in a percentage based on the number of minutes that exist in a year. Therefore, uptime would be the number of minutes the network is available divided by the number of minutes in a year.
* **Cost**includes the cost of the network components, their installation, and their ongoing maintenance.
* **Reliability**defines the reliability of the network components and the connectivity between them. Mean time between failures (MTBF) is commonly used to measure reliability.
* **Security** includes the protection of the network components and the data they contain and/or the data transmitted between them.
* **Speed**includes how fast data is transmitted between network end points (the data rate).
* **Scalability**defines how well the network can adapt to new growth, including new users, applications, and network components.
* **Topology**describes the physical cabling layout and the logical way data moves between components.

### Types of Networks:

Organizations of different structures, sizes, and budgets need different types of networks. Networks can be divided into one of two categories:

* peer-to-peer
* server-based networks
* Ad hoc Networks

### Peer-to-Peer Network:

A peer-to-peer network has no dedicated servers; instead, a number of workstations are connected together for the purpose of sharing information or devices. Peer-to-peer networks are designed to satisfy the networking needs of home networks or of small companies that do not want to spend a lot of money on a dedicated server but still want to have the capability to share information or devices like in school, college, cyber cafe

### Server-Based Networks:

In server-based network data files that will be used by all of the users are stored on the one server. With a server-based network, the network server stores a list of users who may use network resources and usually holds the resources as well. This will help by giving you a central point to set up permissions on the data files, and it will give you a central point from which to back up all of the data in case data loss should occur.

1. **Ad hoc Networks:**

Mobile Ad hoc Networks present a number of unique problems for intrusion detection system.Network traffic can be monitored on a wired network segment, but ad hoc nodes can only monitor network traffic within their observable radio range. Since MANETs have no predefined structure, traditional wired routing algorithm are not suitable for MANETs.On the other hand, presence of malicious node in the network may depreciate the performance of entire network.

**Network security:**

Network security consists of the provisions and policies adopted by a network Administrator to prevent and monitor unauthorized access, misuse modification or denial of a computer network and network accessible resources.The process of taking physical and software preventative measures to protect the underlying networking infrastructures from unauthorized access user and program to perform their permitted critical functions within a secure environment. A node that sends a false routing information can be considered as a compromised node and the node that drops packets or does not faithfully execute the routing algorithm can be considered as a malicious node.Researchers have proposed a number of Intrusion Detection System (IDS) such anomaly based, signature based, network based and host based intrusion detection systems.Though these IDS came into existence, the malicious nodes are still taking part in the routing activities.

**Advantages of Networking:**

1. **Easy Communication:**

It is very easy to communicate through a network. People can communicate efficiently using a network with a group of people. They can enjoy the benefit of emails, instant messaging, telephony, video conferencing, chat rooms, etc.

1. **Ability to Share Files, Data and Information:**

This is one of the major advantages of networking computers. People can find and share information and data because of networking. This is beneficial for large organizations to maintain their data in an organized manner and facilitate access for desired people.

1. **Sharing Hardware:**

Another important advantage of networking is the ability to share hardware. For an example, a printer can be shared among the users in a network so that there’s no need to have individual printers for each and every computer in the company. This will significantly reduce the cost of purchasing hardware.

1. **Sharing Software:**

Users can share software within the network easily. Networkable versions of software are available at considerable savings compared to individually licensed version of the same software. Therefore large companies can reduce the cost of buying software by networking their computers

**Advantages of network security:**

1. Network security helps in protecting personal data of clients and existing on network.
2. Network security facilitates protection of information that is shared between computers on network.
3. Hacking attempts or virus/spyware attacks from the internet will not be able to harmphysical computers. External possible attacks are prevented.
4. Network security provides different levels of access. If there are various computers attached to a network, there may be some computers that may have greater access to information than others.
5. Private networks can be provided protection from external attacks by closing them off from internet. Network Security makes them safe from virus attacks, etc.

**1.3 MOBILE ADHOC NETWORK:**

Mobile Adhoc NETwork (MANET) plays a vital role in various application due to its significant features are its reliability and scalability. In MANETs each node acts as both transmitter and receiver. It does not have any fixed architecture. Thus all nodes are free to move randomly. In Wireless network transmission is limited that is communication is possible with in a range. In MANET there is a direct communication within a range whereas for those beyond the radio range, nodes depend on their neighbour nodes to transmit [1]. Intruders can compromise the network operation by attacking the MAC or network layers. The network layer, especially the routing algorithm is more vulnerable because of the use of cooperative routing algorithms.

Due to its open medium MANETs can be easily attacked by malicious nodes; therefore it is necessary to provide security against attacks. Because of distributed architecture and changing topology centralized monitoring technique will not be effective. So that it is necessary to develop an intrusion detection system (IDS). Intrusion is an action aimed to violate security and compromise network components. Network layer attacks can be classified into two categories namely active and massive attacks [3]. Passive attack does not affect routing but it simply analyze network traffic and access some valuable information such as node identity, location and topology of network. Some of passive attacks are Eavesdropping, Traffic analysis and location disclosure. The disturbing activities such as modifying, injecting, forging, fabricating or dropping data or routing packets in networks are termed as active attacks.

Active attacks are classified into two types. They are malicious packet dropping and routing attacks. Routing attack is further categorized into sleep deprivation, black hole, grey hole, rushing and Sybil. Considering all attacks a general intrusion detection system may provide better solution to MANETs. Intrusion Detection System can be divided into three types based on its mechanism namely anomaly-based intrusion detection (ABID), knowledge-based intrusion detection (KBID) and specification-based intrusion detection (SBID)

The functions of intrusion detection system are monitoring network activities,

Configuring systems for generation of vulnerability reports, assessing integrity, recognizing attacks, analyzing regular activity and policy violations. Many approaches have been proposed concentrating on MANETs security.

**1.4 ATTACKS IN MANNET:**

An attack is any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset. Various types of attacks in MANET.

**INTERNAL AND EXTERNAL ATTACK**

Internal attacks are directly leads to the attacks on nodes presents in network and links interface between them. This type of attacks may broadcast wrong type of routing information to other nodes [10]. Internal attacks are sometimes more difficult to handle as compare to external attacks, because internal attacks arise from more trusted nodes. The wrong routing information generated by compromised nodes or malicious nodes are difficult to identify. This difficulty occurs due to the compromised nodes, which are able to generate the valid signature using their private keys.

External attacks are attacks launched by adversaries who are not initially authorized to participate in the network operations. These attacks usually aim to cause network congestion, denying access to specific network function or to disrupt the whole network operations. Bogus packets injection, denial of service, and impersonation are some of the attacks that are usually initiated by the external attackers.

**ACTIVE AND PASSIVE ATTACK**

MANETs are more susceptible to passive attacks. A passive attack does not alter the data transmitted within the network. But it includes the unauthorized “listening” to the network traffic or accumulates data from it. Passive attacker does not disrupt the operation of a routing protocol but attempts to discover the important information from routed traffic. Detection of such type of attacks is difficult since the operation of network itself doesn’t get affected. In order to overcome this type of attacks powerful encryption algorithms are used to encrypt the data being transmitted.

Active attacks are very severe attacks on the network that prevent message flow between the nodes. However active attacks can be internal or external. Active external attacks can be carried out by outside sources that do not belong to the network. Internal attacks are from malicious nodes which are part of the network, internal attacks are more severe and hard to detect than external attacks. These attacks generate unauthorized access to network that helps the attacker to make changes such as modification of packets, DoS, congestion etc. The active attacks are generally launched by compromised nodes or malicious nodes. Malicious nodes change the routing information by advertising itself as having shortest path to the destination.

**MOBILE AND WIRED ATTACK**

Mobile attacks are done by mobile attackers that have the same capabilities as the other nodes in the ad hoc networks. Since they have the same resources limitations, their capabilities to harm the networks operations are also limited. For instance, with the limited transmitting capabilities and battery powers, mobile attackers could only jam the wireless links within its vicinity. They are not capable to launch the network jamming attacks to disrupt the whole networks operations.

Wired attacks are done by wired attackers that are capable of gaining access to the external resources such as the electricity. Since they have more resources, they could launch more severe attacks in the networks, such as jamming the whole networks or breaking expensive cryptography algorithms. Existence of the wired attackers in the ad hoc networks (especially in the open environment networks) is always possible as long as the wired attackers are able to locate themselves in the communication range and have access to the wired infrastructures.

**SINGLE AND MULTIPLE ATTACK**

Attackers might choose to launch attacks against the ad hoc networks independently or by colluding with the other attackers. Single attackers usually generate a moderate traffic load as long as they are not capable to reach any wired facilities. Since they also have similar abilities to the other nodes in the networks, their limited resources become the weak points to them. For instance, complex cryptography algorithms could be used to help in defending the authentication, integrity, and the confidentiality services from a single attacker. As it becomes very expensive for the single attackers to break the encrypted messages, nodes in the networks could share the expensive cryptography workloads with each other by exploiting the distributed operations and the multiple connections they had among them.

However, if several attackers are colluding to launch attacks, defending the ad hoc networks against them will be much harder. Colluding attackers could easily shut down any single node in the network and be capable to degrading the effectiveness of network’s distributed operations including the security mechanisms. Adding to the severity, colluding attackers could be widely distributed or reside at the certain area where they presumed high communication rate in the networks exist. If no suitable security measures employed, nodes in that targeted area are susceptible to any kind of denial of service (DoS) attacks that could be launched by the colluding attackers.

**1.5 INTRUSION DETECTION SYSTEM APPROACHES:**

Intrusion Detection System can be divided into three types based on its mechanism namely anomaly-based intrusion detection (ABID), knowledge-based intrusion detection (KBID) and specification-based intrusion detection (SBID). The functions of intrusion detection system are monitoring network activities, configuring systems for generation of vulnerability reports, assessing integrity, recognizing attacks, analyzing regular activity and policy violations.

**Anomaly-Based Intrusion Detection (ABID)**

It is also known as behavior-based intrusion detection. In ABID the normalbehaviour of the network is extracted and compared to the current behaviour of thenetwork to identify malicious activities. ABID includes two operations namely trainingand testing. In training phase, the expected behaviour of the network are collected and used as a profile for user or network behavior.

Testing phase is the process of comparing the normal behaviour with the current behavior. This detection technique involves statistical or mathematical approaches to flag any significant deviation between the two models.

**Knowledge-Based Intrusion Detection (KBID)**

KBID is also termed as misuse detection. KBID system maintains a knowledge base which consists of patterns of well-known attacks. KBID uses expert system to maintain the knowledge of known attacks and an inference engine to detect intrusions using rules in knowledge base. Rule-based approach is used to frame the rules from the facts of knowledge base. The process of gathering information and updating it is more complex task.

**Specification-Based Intrusion Detection (SBID)**

SBID defines the operations of a network or protocol of specifications first and then it monitors the network or protocol operations using the specifications to Identify Intruders. The network actions are monitored and the action that shows deviation from the specifications is considered as intrusions. There is also another intrusion detection approach named hybrid detection approach which is a combined approach of anomaly based and knowledge approaches.

**1.6 METHODS OF INTRUSION DETECTION SYSTEM (IDS):**

Intrusion Detection methods are classified into two types based on the place of fixing the Intrusion Detection System.

**Network Based IDS**

Networks based IDS monitors’ network traffic for a particular network segment or device and analyze the network and application of protocol activity to identify suspicious activity. Network based IDS are best suited for alert generation of intrusion from outside world. It is inserted at different points of LAIN to monitor the packets. The information is split into packets and sent through LAN. If the IDS is placed outside the firewall it will be useful to detect the intruder.

**Host Based IDS**

A host-based intrusion detection system (HIDS) is a system that monitors a computer system on which it is installed to detect an intrusion and/or misuse, and responds by logging the activity and notifying the designated authority. A HIDS can be thought of as an agent that monitors and analyzes whether anything or anyone, whether internal or external, has circumvented the system’s security policy.

Host based IDS monitors the characteristics of single host and the events occur in that host for suspicious activity. Host based IDS placed on network resources monitor the audit logs. If any intruder access the resources it will immediately trigger an alarm. Host based IDS overcome the problems occur in the Network based IDS.

Many methods have been proposed for intrusion detection in MANETs. But most of the methods are acknowledgement based systems. These methods detect malicious nodes depending only upon the acknowledgements.

**CHAPTER 2**

**LITERATURE SURVEY**

1. **Intrusion Detection in Wireless Ad-Hoc Networks**

**Authors: Zhang & Lee**

As the recent denial-of-service attacks on several major Internet sites have shown us, no open computer network is immune from intrusions. The wireless ad-hoc network is particularly vulnerable due to its features of open medium, dynamic changing topology, cooperative algorithms, lack of centralized monitoring and management point, and lack of a clear line of defense. Many of the intrusion detection techniques developed on a fixed wired network are not applicable in this new environment. How to do it differently and effectively is a challenging research problem. In this paper, we first examine the vulnerabilities of a wireless ad-hoc network, the reason why we need intrusion detection, and the reason why the current methods cannot be applied directly. We then describe the new intrusion detection and response mechanisms that we are developing for wireless ad-hoc networks.

1. **Adaptive Intrusion Detection & Prevention of Denial of Service Attacks in MANETs**

**Authors: Nadeem & Howrath**

Mobile ad-hoc networks (MANETs) are well known to be vulnerable to various attacks, due to features such as lack of centralized control, dynamic topology, limited physical security and energy constrained operations. In this paper we focus on preventing denial-of-service (DoS) attacks. As an example, we consider intruders that can cause DoS by exploiting the route discovery procedure of reactive routing protocols. We show the unsuitability of tools such as control chart, used in statistical process control (SPC), to detect DoS and propose an anomaly-based intrusion detection system that uses a combination of chi-square test & control chart to first detect intrusion and then identify an intruder. When the intruder is isolated from the network we show reduced overhead and increased throughput. Simulation results show that AIDP performs well at an affordable processing overhead over the range of scenarios tested.

1. **A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks**

**Authors: Nadeem and Howrath**

In the last decade, mobile ad hoc networks (MANETs) have emerged as a major next generation wireless networking technology. However, MANETs are vulnerable to various attacks at all layers, including in particular the network layer, because the design of most MANET routing protocols assumes that there is no malicious intruder node in the network. In this paper, we present a survey of the main types of attack at the network layer, and we then review intrusion detection and protection mechanisms that have been proposed in the literature. We classify these mechanisms as either point detection algorithms that deal with a single type of attack, or as intrusion detection systems (IDSs) that can deal with a range of attacks. A comparison of the proposed protection mechanisms is also included in this paper. Finally, we identify areas where further research could focus.

1. **Lightweight Sybil Attack Detection in MANETs**

**Author: *Abbas, Merabti, Llewellyn, Kifayat***

Fully self-organized mobile ad hoc networks (MANETs) represent complex distributed systems that may also be part of a huge complex system, such as a complex system-of-systems used for crisis management operations. Due to the complex nature of MANETs and its resource constraint nodes, there has always been a need to develop lightweight security solutions. Since MANETs require a unique, distinct, and persistent identity per node in order for their security protocols to be viable, Sybil attacks pose a serious threat to such networks. A Sybil attacker can either create more than one identity on a single physical device in order to launch a coordinated attack on the network or can switch identities in order to weaken the detection process, thereby promoting lack of accountability in the network. In this research, we propose a lightweight scheme to detect the new identities of Sybil attackers without using centralized trusted third party or any extra hardware, such as directional antennae or a geographical positioning system. Through the help of extensive simulations and real-world testbed experiments, we are able to demonstrate that our proposed scheme detects Sybil identities with good accuracy even in the presence of mobility

**CHAPTER 3**

**SYSTEM REQUIREMENTS**

**3.1 HARDWARE REQUIREMENTS:**

System : Intel(R) Pentium(R) CPU B950@ 2.10 GHz.

Hard Disk : 200 GB.

Input/output Peripheral : Mouse and Keyboard.

Monitor : 15” Colour.

Mouse : Two button Mouse.

Ram : 2.00 GB.

**3.2 SOFTWARE REQUIREMENTS:**

Operating system : Fedora 20.

Coding Language : TCL Script, C programming

IDE : Turbo C

**3.3 SOFTWARE ENVIRONMENT**

**Network Simulator 2:**

Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

Ns began as a variant of the [REAL network simulator](http://www.cs.cornell.edu/home/skeshav/real/overview.html) in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the [VINT project](http://www.isi.edu/nsnam/vint/index.html) at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is supported through DARPA with [SAMAN](http://www.isi.edu/saman/index.html) and through NSF with [CONSER](http://www.isi.edu/conser/index.html), both in collaboration with other researchers including [ACIRI](http://www.aciri.org/). Ns has always included substantial contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems.

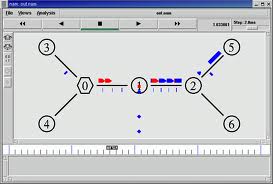
1996-97, ns version 2 (ns-2) was initiated based on a refactoring by Steve McCanne. Use of Tcl was replaced by MIT's [Object Tcl (OTcl)](http://en.wikipedia.org/wiki/OTcl), an [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming) dialect [Tcl](http://en.wikipedia.org/wiki/Tcl" \o "Tcl). The core of ns-2 is also written in C++, but the C++ simulation objects are linked to shadow objects in OTcl and variables can be linked between both language realms. Simulation scripts are written in the OTcl language, an extension of the Tcl scripting language.

Presently, ns-2 consists of over 300,000 lines of source code, and there is probably a comparable amount of contributed code that is not integrated directly into the main distribution (many [forks](http://en.wikipedia.org/wiki/Fork_(software_development)) of ns-2 exist, both maintained and unmaintained). It runs on [GNU/Linux](http://en.wikipedia.org/wiki/GNU/Linux), [FreeBSD](http://en.wikipedia.org/wiki/FreeBSD), [Solaris](http://en.wikipedia.org/wiki/Solaris_(operating_system)), [Mac OS X](http://en.wikipedia.org/wiki/Mac_OS_X) and Windows versions that support [Cygwin](http://en.wikipedia.org/wiki/Cygwin). It is licensed for use under [version 2](http://en.wikipedia.org/wiki/GNU_General_Public_License) of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License).

The general process of creating a simulation can be divided into several steps:

1. *Topology definition*: to ease the creation of basic facilities and define their interrelationships, ns-3 has a system of containers and helpers that facilitates this process.
2. *Model development*: models are added to simulation (for example, UDP, IPv4, point-to-point devices and links, applications); most of the time this is done using helpers.
3. *Node and link configuration*: models set their default values (for example, the size of packets sent by an application or MTU of a point-to-point link); most of the time this is done using the attribute system.
4. *Execution*: simulation facilities generate events, data requested by the user is logged.
5. *Performance analysis*: after the simulation is finished and data is available as a time-stamped event trace. This data can then be statistically analysed with tools like [R](http://en.wikipedia.org/wiki/R_(programming_language)) to draw conclusions.
6. *Graphical Visualization*: raw or processed data collected in a simulation can be graphed using tools like [Gnuplot](http://en.wikipedia.org/wiki/Gnuplot" \o "Gnuplot), [matplotlib](http://en.wikipedia.org/wiki/Matplotlib) or [XGRAPH](http://en.wikipedia.org/w/index.php?title=XGRAPH&action=edit&redlink=1).

*Fig: 3.1 Working with NS2*



**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 EXISTING SYSTEM:**

Routing protocols in MANET having limited transmission power and it can communicate only with the nodes within the communication range. Due to this limitation nodes in MANETs assume that other nodes always cooperate with each other to relay data. Thisassumption leaves the attackers with the opportunities to achieve significant impact on the network with just one or two compromised nodes. To address this problem, IDS should be added to enhance the security level of MANETs. A number of secure IDS schemes have been brought forward to intrusion in MANETs.

**SYBIL ATTACK:**

Each node in a MANET requires a unique address toparticipate in routing, through which nodes are identified. However, in a MANET there is no central authority to verify these identities. An attacker can exploit this property and send control packets, for example RREQ or RREP, using different identities; this is known as a sybil attack. This is an impersonation attack where the intruder could use either random identities or the identity of another node to create confusion in the routing process, or to establish bases for some other severe attack. In summary, we note that the motivation of intruders behind launching either packet dropping or routing attacks is to achieve a certain goal such as denial of service (i.e. making certain resources or services, such as applications, web access, printing, or routing, unavailable to the intended users). In addition, other goals of intruders might includepartitioning the network, creating routing loops, discoveringvaluable information, or theft of resources.

**LIGHT WEIGHT ARCHITECTURE:**

Fully self-organized mobile ad hoc networks(MANETs) represent complex distributed systems that may also be part of a huge complex system, such as a complex system-of-systems used for crisis management operations. Due to the complex nature of MANETs and its resource constraint nodes, there has always been a need to develop lightweight security solutions. Since MANETs require a unique, distinct, and persistent identity per node in order for their security protocols to be viable, Sybil attacks pose a serious threat to such networks. A

Sybil attacker can either create more than one identity on a single physical device in order to launch a coordinated attack on the network or can switch identities in order to weaken the detection process, thereby promoting lack of accountability in the network. In this research, we propose a lightweight scheme to detect the new identities of Sybil attackers without using centralized trusted third party or any extra hardware, such as directional antennae or a geographical positioning system. Through the help of extensive simulations and real-world testbed experiments, we are able to demonstrate that our proposed scheme detects Sybil identities with good accuracy even in the presence of mobility.

**4.2 PROPOSED SYSTEM:**

Mobile Ad hoc Networks use cooperative routing algorithms such as Ad hoc On demand Distance Vector (AODV) routing protocol that have been designed to efficiently reroute traffic when confronted with network congestion. However the design of these routing protocol of MANETs assume that there is no malicious intruder node in the network. But MANETs are vulnerable to various types of attacks at all layers. In this paper, we introduce a prediction algorithm which we name as Trust Level prediction algorithm for the prediction of malicious node in the network. This algorithm uses a mathematical prediction technique called linear prediction Algorithm to identify the malicious node in the network.

A Trust level Prediction Algorithm is implemented that monitors and records the activity of each and every node that take part in the packet routing in MANETs.This audit data is used for evaluating the faithfulness of each node based on a mathematical prediction technique called Linear Prediction Algorithm.Computational time will be reduced because of the implementation of mathematical based prediction technique.

**LINEAR PREDICTION ALGORITHM:**

**Linear prediction** is a mathematical operation where future values of a [discrete-time](http://en.wikipedia.org/wiki/Discrete_time) [signal](http://en.wikipedia.org/wiki/Signal_processing) are estimated as a [linear function](http://en.wikipedia.org/wiki/Linear_transformation) of previous samples.

In [digital signal processing](http://en.wikipedia.org/wiki/Digital_signal_processing), linear prediction is often called [linear predictive coding](http://en.wikipedia.org/wiki/Linear_predictive_coding) (LPC) and can thus be viewed as a subset of [filter theory](http://en.wikipedia.org/wiki/Filter_theory). In [system analysis](http://en.wikipedia.org/wiki/System_analysis) (a subfield of[mathematics](http://en.wikipedia.org/wiki/Mathematics)), linear prediction can be viewed as a part of [mathematical modelling](http://en.wikipedia.org/wiki/Mathematical_model) or [optimization](http://en.wikipedia.org/wiki/Optimization_(mathematics)).

**X(n) = ∑ai X(n-i)**

Where,

i = number of samples

X(n) = Present value

X(n-i) = Past value

ai = Predictor coefficient

**PREDICTOR COEFFICIENT:**

The predictor coefficient can be calculated as the root mean square of the samples. i.e.

It is given as the square root of the mean of the sum of the squares of the samples.

Example:

M = S12 + S22 + S32 + ……. + Sn2

**n**

R = √M

Where M = Mean of the sum of the squares of the samples.

R = Square Root of Mean

**ADVANTAGES:**

Improves the performance of the overall MANETs by increasing the packet delivery ratio, routing overhead, network overhead.

**APPLICATIONS:**

Linear prediction are mainly used in speech processing techniques. For the first time we use it in Network Security concept to predict whether a node will be malicious or not in the future.

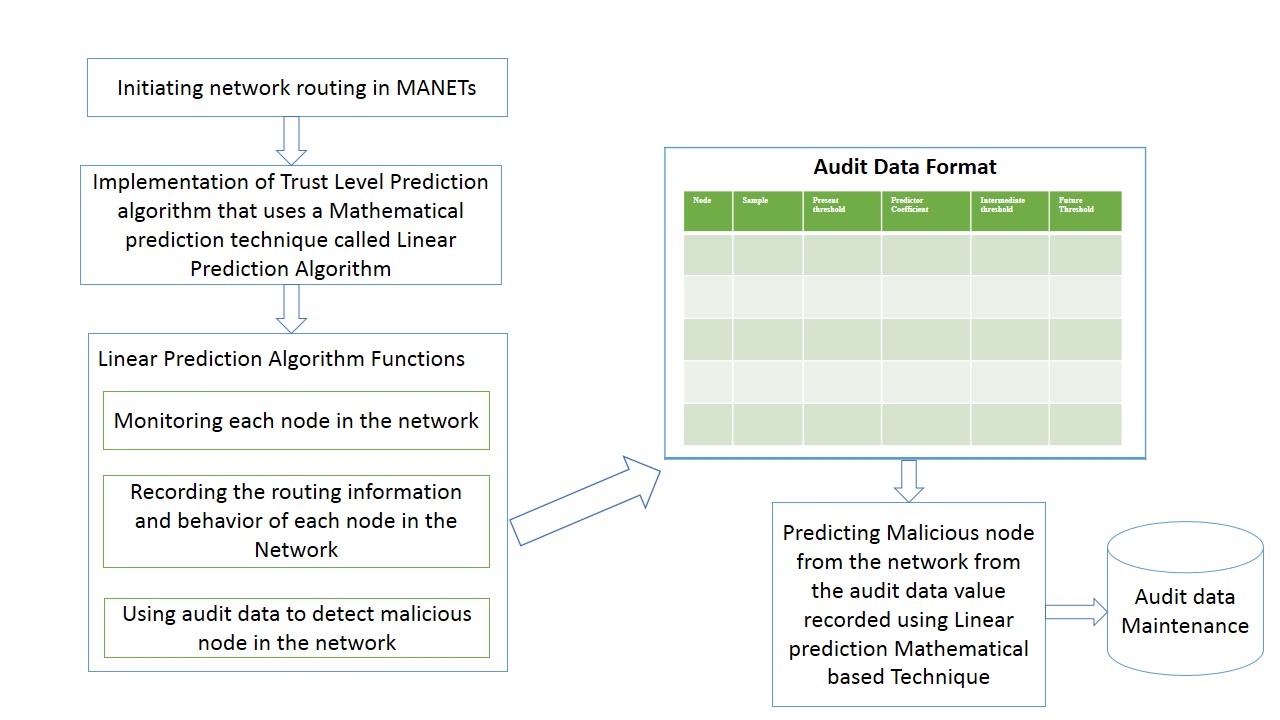
**4.3 PROPOSED SYSTEM ARCHITECTURE:**

Fig 4.1: System Architecture

**CHAPTER 5**

**MODULE DESCRIPTION**

* 1. **MODULES:**

Various modules involved in our system are,

1. MANETs Establishment and Implementation of Packet drop scenario in MANETs.
2. Implementation of Trust Level Routing Algorithm and mathematical prediction Technique.
3. Implementation of Node monitoring and malicious node detection mechanism and Audit Data maintenance.
   1. **MODULE DESCRIPTION:**

**MANETs Establishment and Implementation of Packet drop scenario in MANETs.**

The environment with mobile nodes that send, receive and drop packet data is established using TCL language in Network Simulator 2.35 in Fedora Operating System Version 20.

**Implementation of Trust Level Prediction Algorithm and mathematical prediction Technique.**

The Algorithm is implemented based on previous sample input and outputs obtained in other papers using Digital Signal Processing principles. First the trace file is interpreted and the necessary details for say three samples of each node is extracted and recorded in a separate excel sheet. Then the linear prediction algorithm here we name it as trust level prediction algorithm is implemented and a future threshold is calculated. This threshold defines whether a node is malicious or not.

**Implementation of Node monitoring and malicious node detection mechanism and Audit Data maintenance.**

In this module the mobile nodes are monitored and this data is recordedwhich is implemented using C programming language. Once the malicious node is detected it is reported to the administration which we consider as the areas for future research.

**CHAPTER 6**

**SOFTWARE DESCRIPTION**

**6.1 NETWORK SIMULATOR-2:**

Network Simulator is an Object-oriented, discrete event driven network simulator developed at UC Berkeley that simulates variety of IP networks. It implements network protocols such as TCP and UPD, traffic source behaviour such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra. NS also implements multicasting and some of the MAC layer protocols for LAN simulation. Currently, NS (version 2) written in C++ and OTcl (Tcl script language with Object-oriented extensions) is available.

**NS-2:**

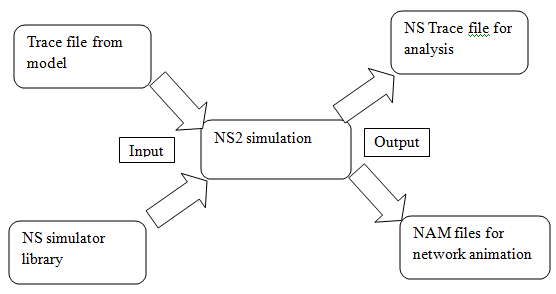
NS-2 is a discrete network simulator that provides significant simulation of transport, routing, and multicast over-wired and wireless networks. Ns-2 code is written using C++ and OTCL and is kept in a separate file that is executed by OTCL interpreter, thereby generating a communication trace file and a NAM (Network animator) file as its output. The output trace file describes the network topologies and log events that exhibit the output of the nodes communicating with each other and the NAM file animates the traces derived from the simulation and analyse the events to understand the network behaviour.

Ns-2 may be installed over either LINUX or any UNIX-based operating systems or over Microsoft Windows operative systems using *Cygwin*. In our case NS-2 (ns-allinone-2.34) installed in LINUX platform

Once installed, program files needs to be written in OTCL script language and execute NS with this file as an input. This OTCL file incorporates traffic generator trace file from Vanet MobiSim and other communication parameters like routing protocol, CBR (constant bit rate) etc. for using them in the simulation.

**NS-2 Trace Analysis:**

After NS-2 output trace file and network animator NAM file generated, the next phase of task lies in animating the NAM file to visualize node movement behaviour and more importantly, extracting information from trace data file and appropriately plot and study them for performance evaluation

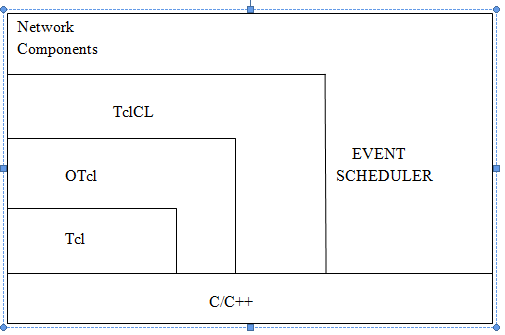


**Fig.6.1 Input/output flow diagram of NS2 simulation**

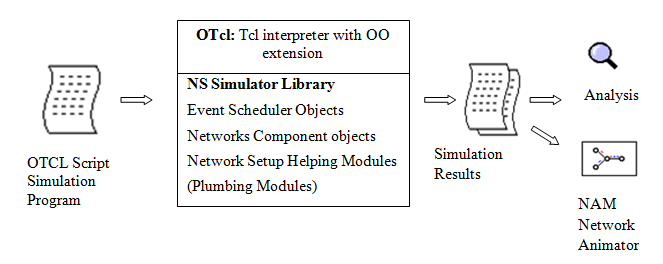
**6.2 NS2 ARCHITECTURE:**

NS2 consists mainly of two languages: C++ and OTcl

* OTcl is a language for creating a network so that you don’t have to recompile the codes every time you make changes to your simulation scenarios. It is also used to connect blocks (e.g., classifier, links, and agent) within each NS2 components (e.g., node).OTcl for control,
  + - Simulation setup, configuration, occasional actions
    - Compromise between speed and abstraction level(s) offered to the user

****

**Fig.6.2 NS2 Architecture**

**Fig.6.2 (b) Interpreting Otcl and C++**

* C++ defines internal mechanism of each block. It does things like packet forwarding, scheduling events, collecting statistic. C++ for packet processing,
  + - Fast execution, detailed, full control over execution
    - To make simulator scalable, packet processing must be done at C++ level
  1. **TCL**

**TCL (**Tool Command Language) is a scripting language created by John Ousterhout. A special simulator object is defined in NS2 that perform process-based simulation on the network objects. Tcl gained acceptance on its own. It is commonly used for rapid prototyping, scripted applications, GUIs and testing. Tcl is used on embedded systems platforms. Tcl's features include

* All operations are commands, including language structures. They are written in prefix notation. Commands are commonly variadic.
* Everything can be dynamically redefined and overridden. All data types can be manipulated as strings, including source code.
* Fully dynamic, class-based object system, TclOO, including advanced features such as meta-classes, filters, and mixins.
* Event-driven interface to sockets and files. Time-based and user-defined events are also possible.
* Variable visibility restricted to lexical (static) scope by default, but allowing process to interact with the enclosing functions' scopes.
* All commands defined by Tcl itself generate error messages on incorrect usage.
* Extensibility, via C, C++, Java, and Tcl Interpreted language.

**CHAPTER 7**

**IMPLEMENTATION AND RESULTS**

System is performed with the network simulator (NS) 2.35 environment on a platform Linux. The system is running on a laptop with Core 2 Duo T 7250 CPU and 2-GB RAM. Both the physical layer and MAC 802.11 are included in the wireless extension ofNS2. The network environment with fourty wireless nodes is created. The both physical layer and MAC 802.11 are included.

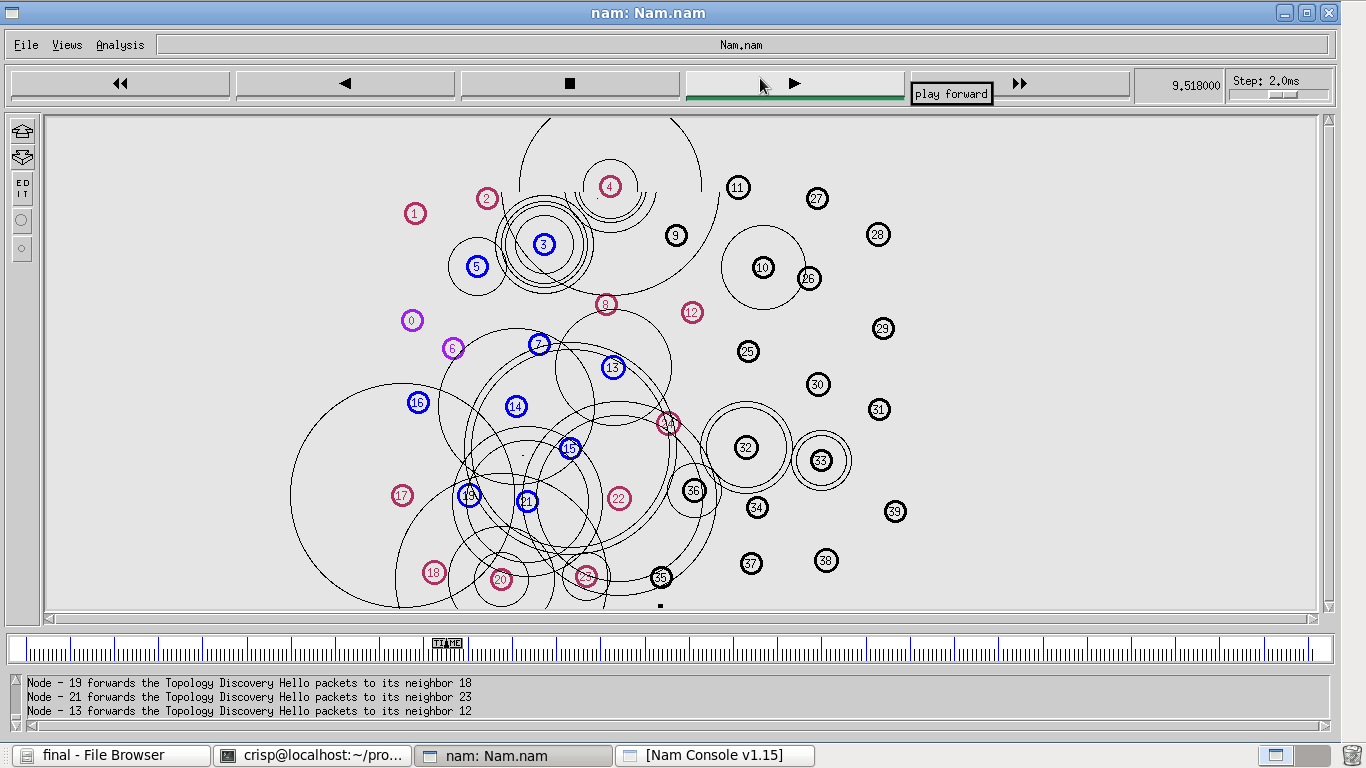
The packet transmission between the nodes is transmitted using UDP protocol. The initializing nodes are specified and time period to initialize is fixed. The channel type, propagation model, maximum number of packets, routing protocol, dimensions of topology and the simulation end time are declared. The nodes are configured to ad hoc routing and the connections between the nodes are given. The location to store the created nam and trace file is provided and the time period to be evaluated is specified to generate the performance graph. The network formed is simulated.

The animator shows the mobility of nodes and also the packet transmission between the nodes (the process of sending packets and receiving acknowledgements). Once the simulation finishes, the trace file will be generated and stored in the location specified.

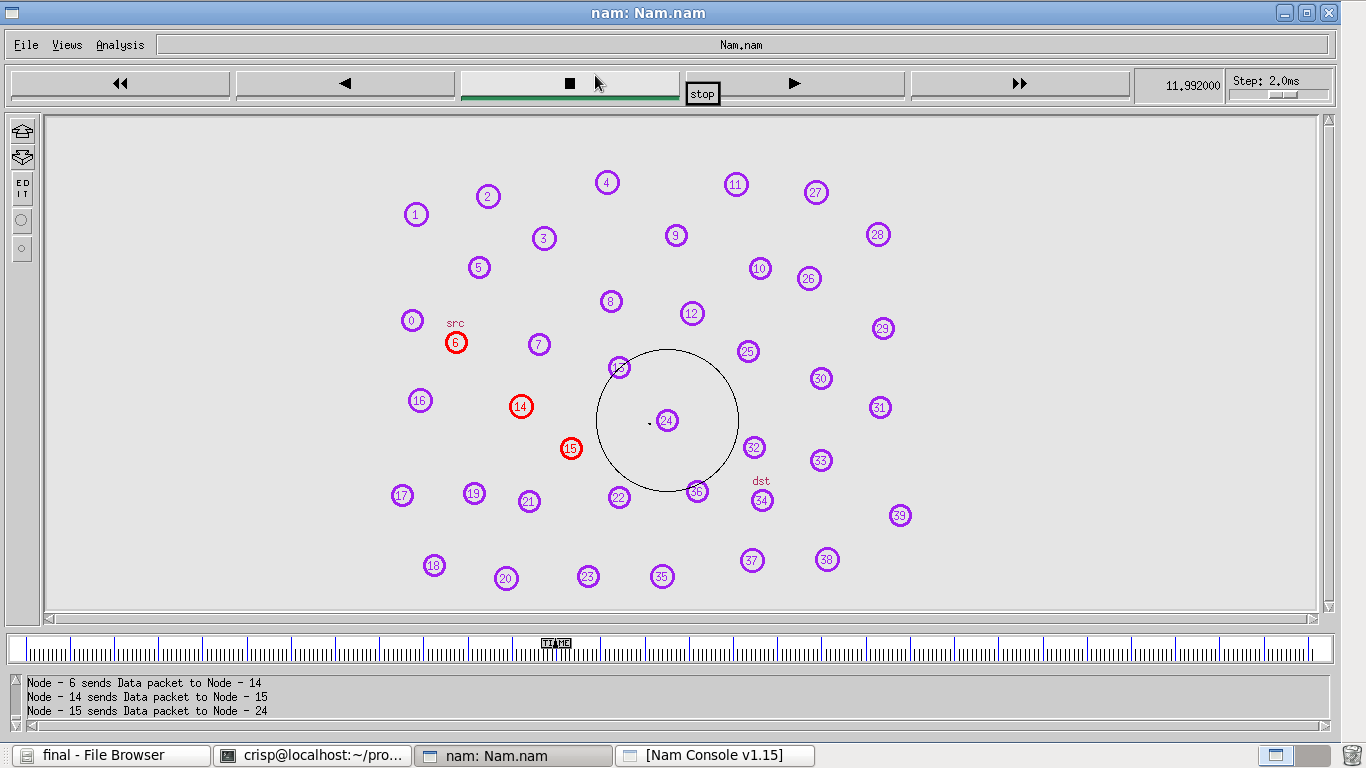
This trace file is interpreted and the necessary details for say three samples of each node is extracted and recorded in a separate excel sheet. Then the linear prediction algorithm here we name it as trust level prediction algorithm is implemented and a future threshold is calculated. This threshold defines whether a node is malicious or not.

**7.1 SCREENSHOTS**

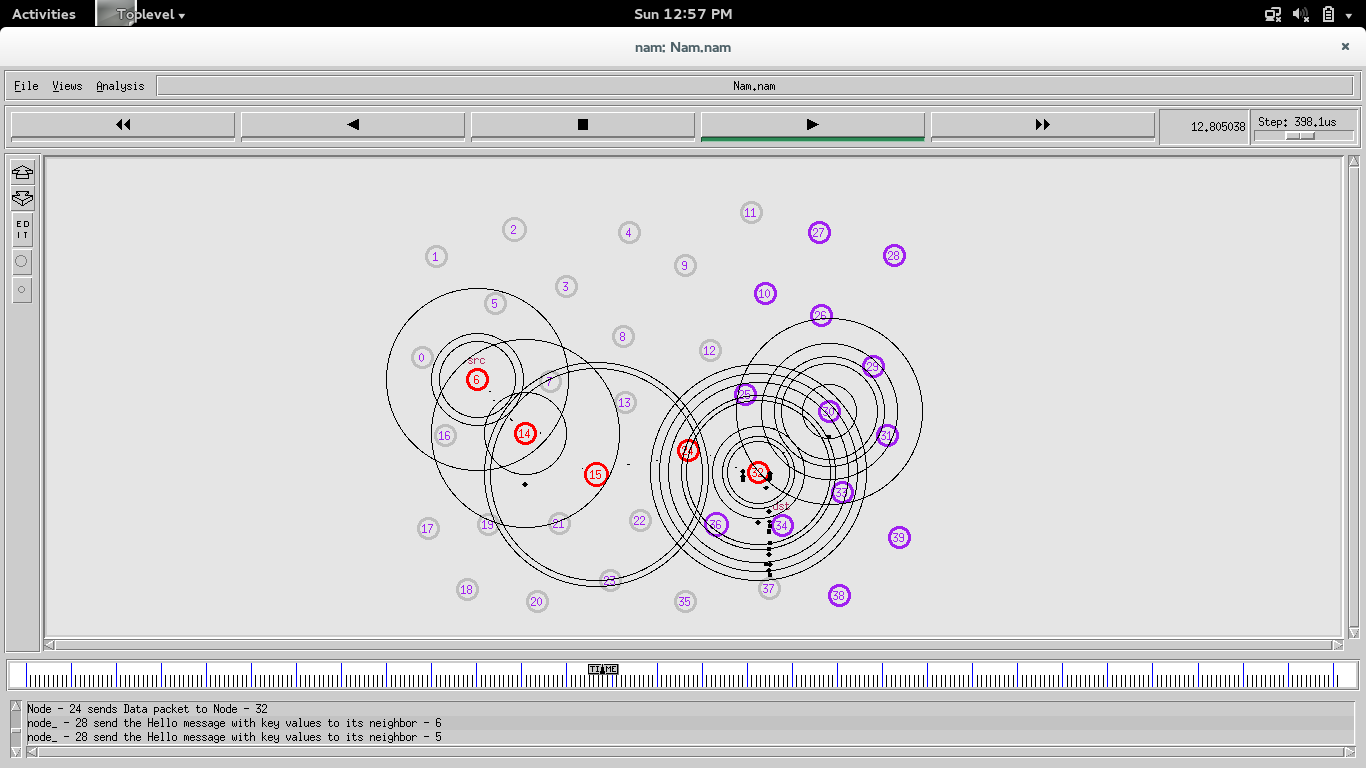
**7.1.1 TOPOLOGY FORMATION:**

****

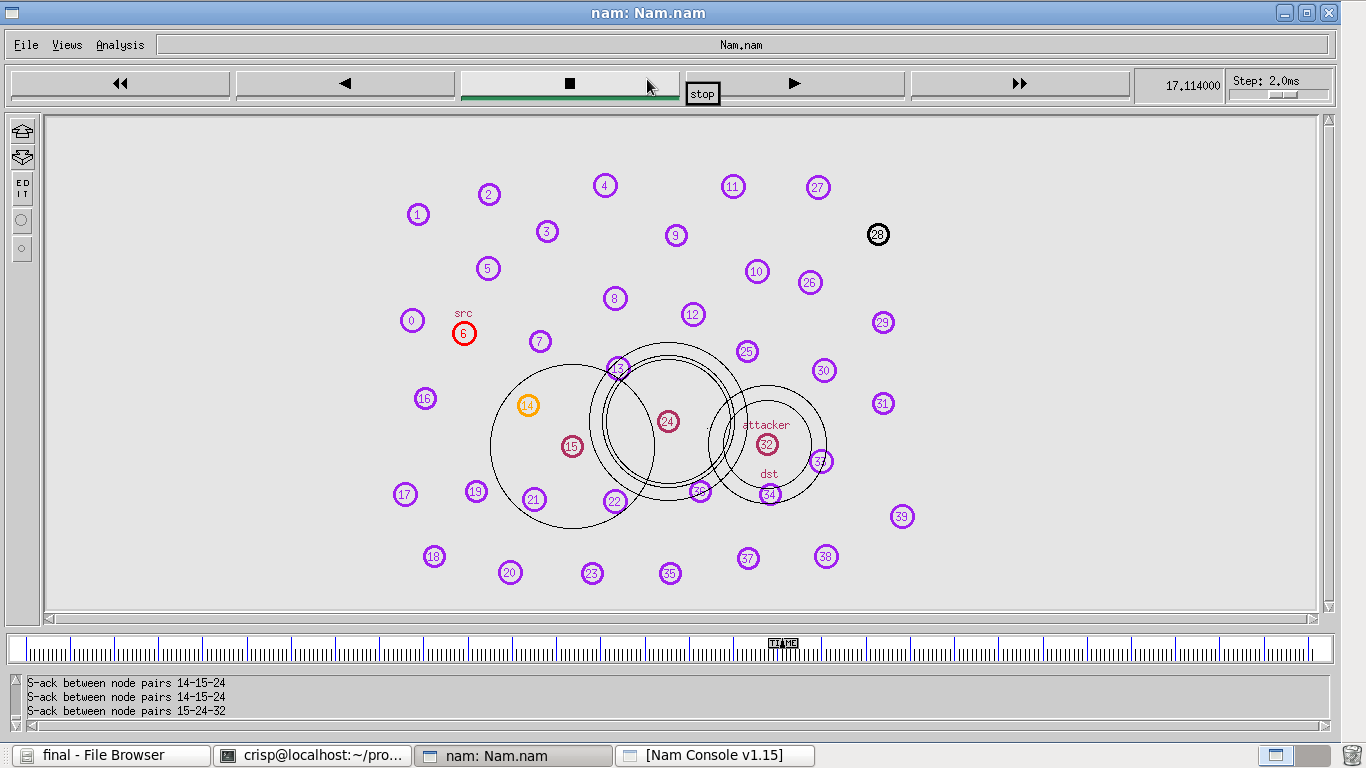
**7.1.2 PACKET TRANSMISSION:**

****

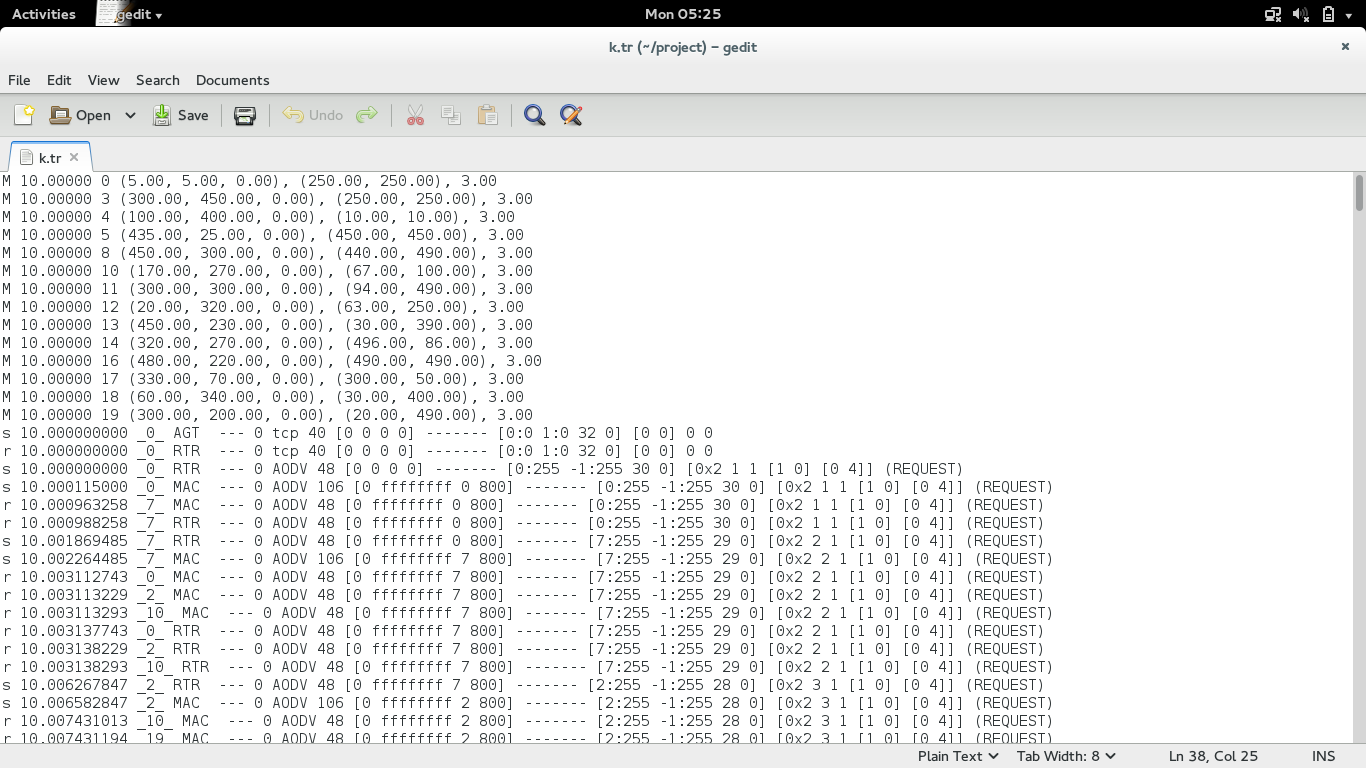
**7.1.3 PACKET DROPING:**



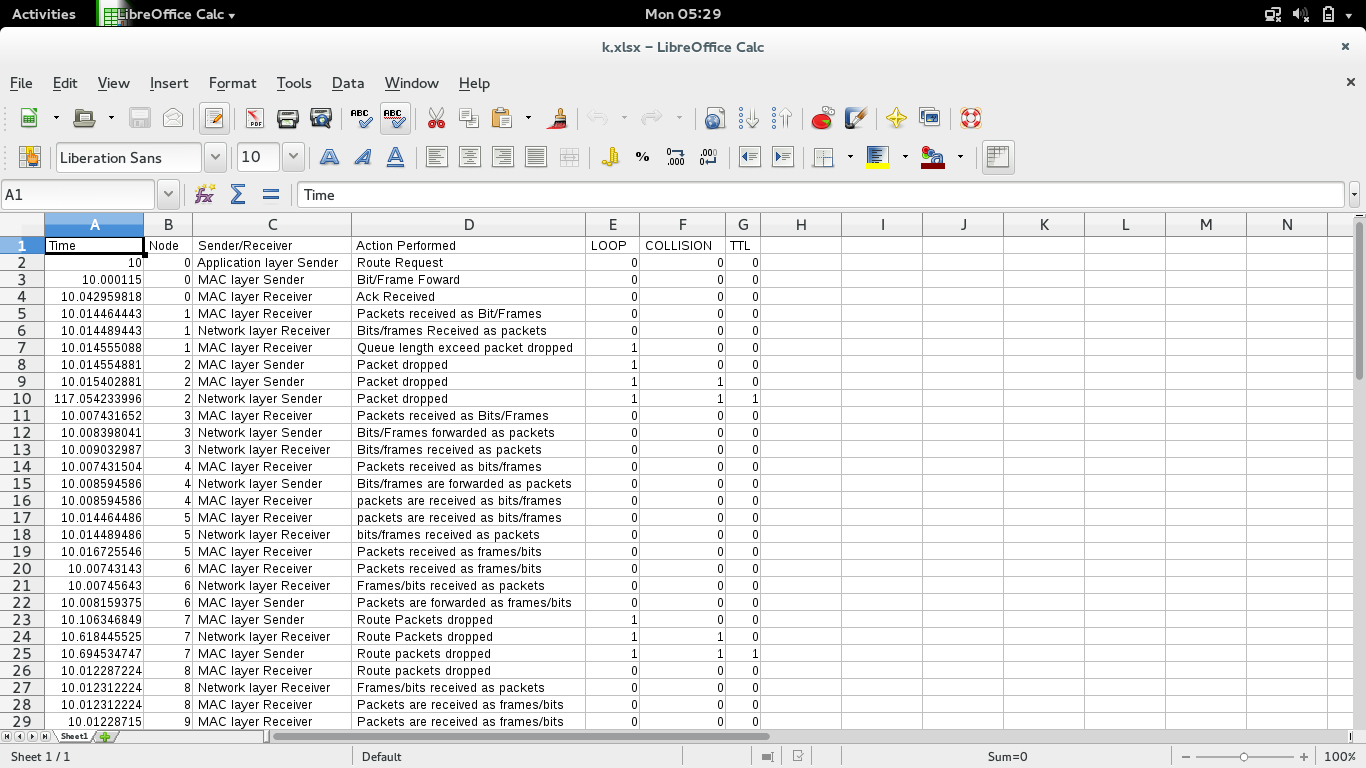
**7.1.4 MALICIOUS NODES:**

****

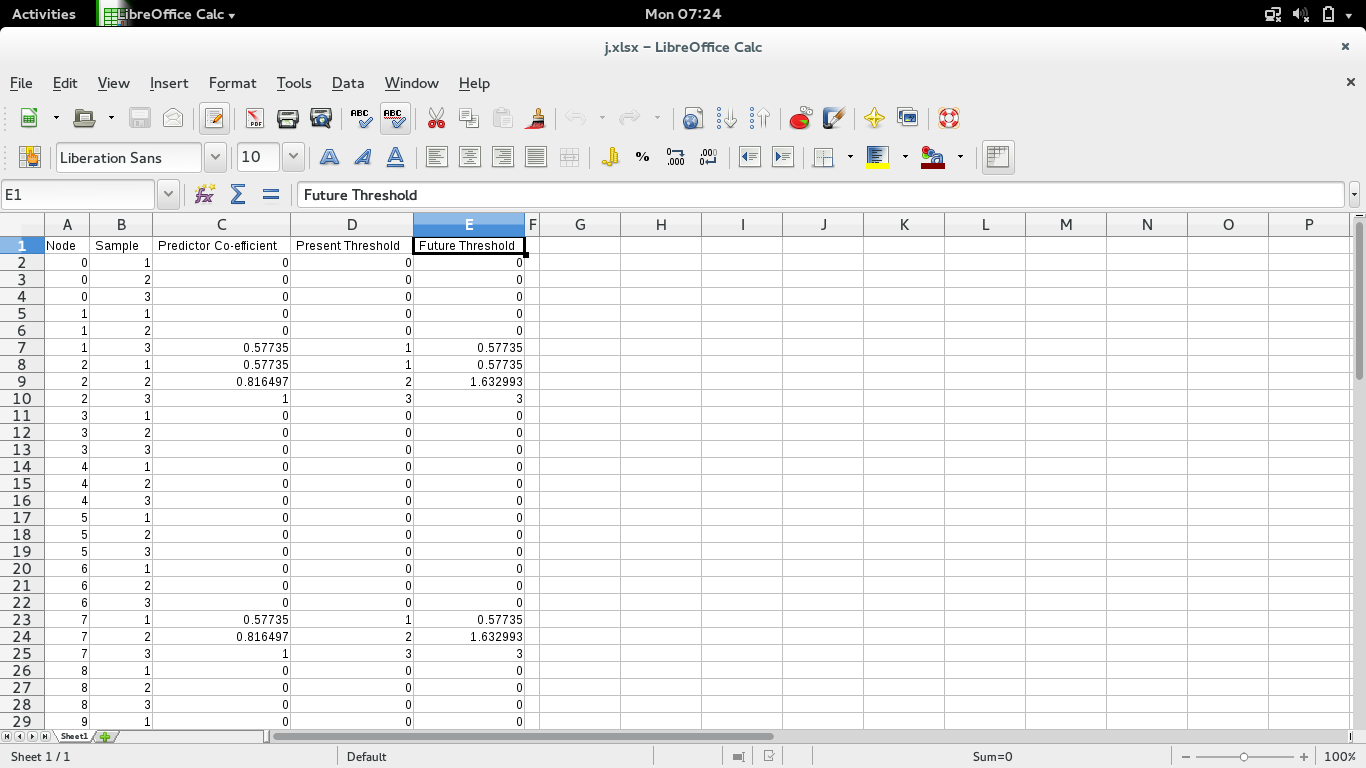
**7.1.5 TRACE FILE:**



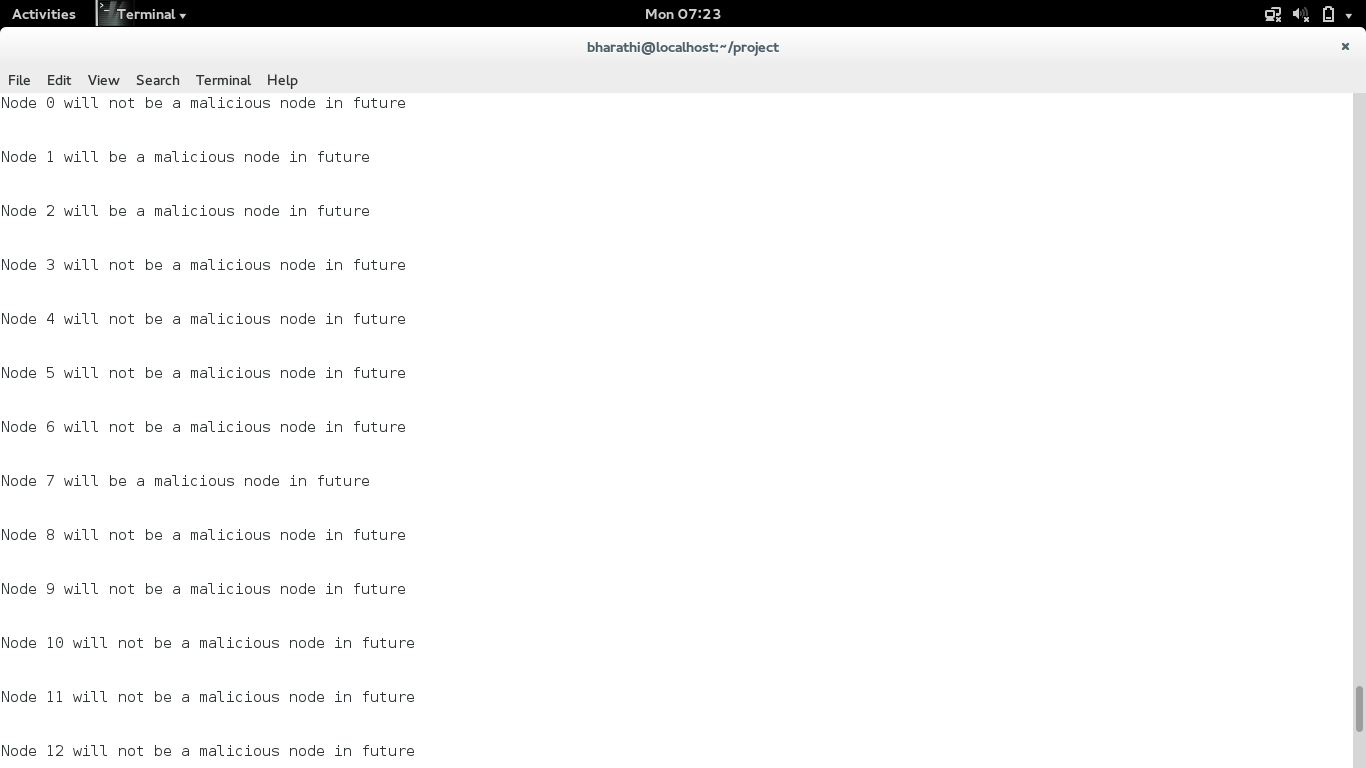
**7.1.6 NECESSARY DETAILS EXTRACTED FROM TRACE FILE IN EXCEL SHEET**

****

**7.1.7 THRESHOLD CALCULATION IN EXCEL SHEET:**



**SYSTEM OUTPUT:**

****

**CHAPTER-8**

**CONCLUSION**

Packet-dropping attack has always been a major threat to the security in MANETs. In this research paper, we have proposed a prediction algorithm that predicts whether a node will be malicious or not in the future. The results demonstrated positive performances against various IDS in the cases of receiver collision, limited transmission power, and false misbehaviour report. Since mathematical based prediction technique called linear prediction algorithm is used to predict malicious node in the network, the computational time has been largely reduced.The results obtained show good performance of proposed technique.

**APPENDIX**

**SOURCE CODE FOR MANNET ENVIRONMENT WITH PACKET DROP SCENARIO TCL SCRIPT**

set val(chan) Channel/WirelessChannel ;

set val(prop) Propagation/TwoRayGround ;

set val(netif) Phy/WirelessPhy ;

set val(mac) Mac/802\_11;

set val(ll) LL ;

set val(ant) Antenna/OmniAntenna ;

set val(ifq) Queue/DropTail/PriQueue ;

set val(ll) LL ;

set val(ant) Antenna/OmniAntenna ;

set val(ifqlen) 500 ;

set val(nn) 20 ;

set val(rp) AODV ;

set val(x) 500 ;

set val(y) 500 ;

set val(stop) 150 ;

set ns [new Simulator]

set tracefd [open k.tr w]

$ns trace-all $tracefd

set WindowVsTime [open win.tr w]

set namtrace [open k.nam w]

$ns namtrace-all $namtrace

$ns namtrace-all-wireless $namtrace $val(x) $val(y)

set topo [new Topography]

$topo load\_flatgrid $val(x) $val(y)

create-god $val(nn)

$ns node-config -adhocRouting $val(rp) \

-llType $val(ll) \

-macType $val(mac) \

-ifqType $val(ifq) \

-ifqLen $val(ifqlen) \

-antType $val(ant) \

-propType $val(prop) \

-phyType $val(netif) \

-channelType $val(chan) \

-topoInstance $topo \

-agentTrave ON \

-routerTrace ON \

-macTrace ON \

-movementTrace ON

for {set i 0} {$i < $val(nn)} {incr i} {

set n($i) [$ns node]

}

$n(0) set X\_ 5.0

$n(0) set Y\_ 5.0

$n(0) set Z\_ 0.0

$n(1) set X\_ 490.0

$n(1) set Y\_ 285.0

$n(1) set Z\_ 0.0

$n(2) set X\_ 200.0

$n(2) set Y\_ 230.0

$n(2) set Z\_ 0.0

$n(3) set X\_ 300.0

$n(3) set Y\_ 450.0

$n(3) set Z\_ 0.0

$n(4) set X\_ 100.0

$n(4) set Y\_ 400.0

$n(4) set Z\_ 0.0

$n(5) set X\_ 435.0

$n(5) set Y\_ 25.0

$n(5) set Z\_ 0.0

$n(6) set X\_ 127.0

$n(6) set Y\_ 389.0

$n(6) set Z\_ 0.0

$n(7) set X\_ 68.0

$n(7) set Y\_ 50.0

$n(7) set Z\_ 0.0

$n(8) set X\_ 450.0

$n(8) set Y\_ 300.0

$n(8) set Z\_ 0.0

$n(9) set X\_ 480.0

$n(9) set Y\_ 390.0

$n(9) set Z\_ 0.0

$n(10) set X\_ 170.0

$n(10) set Y\_ 270.0

$n(10) set Z\_ 0.0

$n(11) set X\_ 300.0

$n(11) set Y\_ 300.0

$n(11) set Z\_ 0.0

$n(12) set X\_ 20.0

$n(12) set Y\_ 320.0

$n(12) set Z\_ 0.0

$n(13) set X\_ 450.0

$n(13) set Y\_ 230.0

$n(13) set Z\_ 0.0

$n(14) set X\_ 320.0

$n(14) set Y\_ 270.0

$n(14) set Z\_ 0.0

$n(15) set X\_ 50.0

$n(15) set Y\_ 380.0

$n(15) set Z\_ 0.0

$n(16) set X\_ 480.0

$n(16) set Y\_ 220.0

$n(16) set Z\_ 0.0

$n(17) set X\_ 330.0

$n(17) set Y\_ 70.0

$n(17) set Z\_ 0.0

$n(18) set X\_ 60.0

$n(18) set Y\_ 340.0

$n(18) set Z\_ 0.0

$n(19) set X\_ 300.0

$n(19) set Y\_ 200.0

$n(19) set Z\_ 0.0

$ns at 10.0 "$n(0) setdest 250.0 250.0 3.0"

$ns at 15.0 "$n(1) setdest 45.0 285.0 5.0"

$ns at 40.0 "$n(0) setdest 480.0 300.0 5.0"

$ns at 10.0 "$n(3) setdest 250.0 250.0 3.0"

$ns at 10.0 "$n(4) setdest 10.0 10.0 3.0"

$ns at 10.0 "$n(5) setdest 450.0 450.0 3.0"

$ns at 10.0 "$n(8) setdest 440.0 490.0 3.0"

#$ns at 10.0 "$n(9) setdest 400.0 400.0 3.0"

$ns at 10.0 "$n(10) setdest 67.0 100.0 3.0"

$ns at 10.0 "$n(11) setdest 94.0 490.0 3.0"

$ns at 10.0 "$n(12) setdest 63.0 250.0 3.0"

$ns at 10.0 "$n(13) setdest 30.0 390.0 3.0"

$ns at 10.0 "$n(14) setdest 496.0 86.0 3.0"

$ns at 10.0 "$n(16) setdest 490.0 490.0 3.0"

$ns at 10.0 "$n(17) setdest 300.0 50.0 3.0"

$ns at 10.0 "$n(18) setdest 30.0 400.0 3.0"

$ns at 10.0 "$n(19) setdest 20.0 490.0 3.0"

set tcp [new Agent/TCP/Newreno]

$tcp set class\_ 2

$ns attach-agent $n(0) $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n(1) $sink

$ns connect $tcp $sink

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ns at 10.0 "$ftp start"

$ns at 140.0 "$ftp stop"

$ns at 10.0 "$n(0) label source"

$ns at 10.0 "$n(1) label destination"

$n(0) color Blue

$ns at 10.0 "$n(0) color Blue"

$n(1) color Green

$ns at 10.0 "$n(1) color Green"

$ns at 80.0 "[$n(7) set ragent\_] malicious"

#$ns at 80.0 "$n(7) add-mark n(7) red circle"

$ns at 20.0 "[$n(6) set ragent\_] malicious"

#$ns at 20.0 "$n(6) add-mark n(6) red circle"

$ns at 97.0 "[$n(13) set ragent\_] malicious"

#$ns at 97.0 "$n(13) add-mark n(3) red circle"

$ns at 80.0 "[$n(2) set ragent\_] malicious"

#$ns at 60.0 "$n(2) add-mark n(2) red circle"

$ns at 135.0 "[$n(19) set ragent\_] malicious"

$ns at 95.0 "[$n(3) set ragent\_] malicious"

#proc plotWindow {tcpSource file} {

#global ns

#set time 0.01

#set now [$ns now]

#set cwnd [$tcpSource set cwnd\_]

#puts $file "$now $cwnd"

#$ns at [expr $now+$time] "plotWindow $tcpSource $file"

#}

#$ns at 1.1 "plotWindow $tcp $WindowVsTime"

for {set i 0} {$i < $val(nn)} {incr i} {

$ns initial\_node\_pos $n($i) 30

}

for {set i 0} {$i < $val(nn)} {incr i} {

$ns at $val(stop) "$n($i) reset"

}

$ns at val(stop) "$ns nam-end-wireless $val(stop)"

$ns at $val(stop) "stop"

$ns at 150.01 "puts \" End Simulation \" ; $ns halt"

proc stop {} {

global ns tracefd namtrace

$ns flush-trace

close $tracefd

close $namtrace

}

$ns run

**REFERENCES**

1. Sohail Abbas, Madjid Merabti, David Llewellyn-Jones, and Kashif Kifayat, “Lightweight Sybil Attack Detection in MANETs”, IEEE SYSTEMS JOURNAL, VOL. 7, NO. 2, JUNE 2013.
2. I. Chlamtac, M. Conti, and J. J.-N. Liu, “Mobile ad hoc networking: Imperatives and challenges”, *Ad Hoc Newt.*, vol. 1, no. 1, pp. 13–64, 2003.
3. J. R. Douceur, “The Sybil attack”, presented at the Revised Papers from the First Int. Workshop on Peer-to-Peer Systems, pp. 251–260,2002.
4. J. Newsome, E. Shi, D. Song, and A. Perrig, “The Sybil attack in sensor networks: Analysis and defences”, presented at the 3rd Int. Symp. Information Processing in Sensor Networks (IPSN), pp. 259–268, 2004.