2011

Detection of malicious nodes in mobile ad hoc networks

Ratna Sireesha Singamsetty
The University of Toledo

Follow this and additional works at: http://utdr.utoledo.edu/theses-dissertations

Recommended Citation
http://utdr.utoledo.edu/theses-dissertations/722

This Thesis is brought to you for free and open access by The University of Toledo Digital Repository. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of The University of Toledo Digital Repository. For more information, please see the repository's About page.
A Thesis

entitled

Detection of Malicious Nodes in Mobile Adhoc Networks

by

Ratna Sireesha Singamsetty

Submitted to the Graduate Faculty as partial fulfillment of the requirements for The Master of Science Degree in Engineering

_________________________________
Dr. Mansoor Alam, Committee Chair

_________________________________
Dr. Weiqing Sun, Committee Co-Chair

_________________________________
Dr. Mohammad Niamat, Committee Member

_________________________________
Dr. Patricia Komuniecki, Dean
College of Graduate Studies

The University of Toledo
December 2011
An Abstract of

Detection of Malicious Nodes in Mobile Adhoc Networks

by

Ratna Sireesha Singamsetty

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Engineering

The University of Toledo

December 2011

In recent years, the rapid proliferation of wireless networks, usage of wireless devices and deployment of many mobile computing devices and applications has changed the shape of network security. One field which needs more security is the mobile ad hoc network (MANET). The term “ad hoc” means self-organized nodes that do not have a central entity to govern them. Network security plays a crucial role in this MANET and the traditional way of protecting the networks through firewalls and encryption software is no longer effective and sufficient. In order to provide additional security to the MANET, intrusion detection mechanisms should be added. In this research, a quantitative method for detecting malicious nodes in the Mobile ad-hoc network is proposed. The proposed method is a behavior anomaly based system which makes it scalable, robust, configurable, and dynamic. Voting process is used in this proposed method to confirm whether a node is malicious or not. The proposed method is verified by running simulations with mobile nodes using the Ad-hoc on-demand distance vector (AODV) routing.
Beloved Father                  (Late)Madhava Rao Singamsetty
Beloved Mother                  Sita Maha Lakshmi Singamsetty
Ever Supportive and loving Husband Narasimha Gunda
Loving Brother                  Ajay Ram Singamsetty
Supportive Uncle                VishnuVardhanRao Koneru
Well Wisher                     Richard Tabb Schreder

Their unconditional love, encouragement and blessings have always been the
top reason of my success. Without these people I wouldn’t have achieved this.
Acknowledgements

The completion of thesis is obviously not possible without the support of numerous people. I wish to acknowledge the support and camaraderie of the following people: My primary advisors, Dr. Mansoor Alam and Dr. Weiqing Sun, whose insightful nature, vision and encouragement have been a pillar of support throughout my Masters. I also appreciate Dr. Niamat and his advice on experimental approaches which helped me develop as student. My sincere thanks go out to Mr. Daniel Solarek and Mr. Allen Rioux for supporting me financially. I would also like to thank my friends for giving me a laptop to continue my research especially to the initiators Dr. Clinton Longenecker and Cindy Longenecker. I would also like to thank Tabb Schreder and his family for their love and support. I would like to thank my mother and brother for her unconditional love, faith and encouragement. I would also like to thank my uncle Vishnu Vardan RaoKoneru who has been always with me whenever I needed help. Apart from earning a Master’s degree, I am fortunate to have my soul-mate and husband. I could not have completed this long journey without my husband, Narasimha Gunda. I would like to thank Kavitha and Omar for helping me in answering some questions/doubts related to my research. I would also like to thank my friends Teja Lanka, Chandrika Kota, Ramya Gullapalli, Lakshmi Yedaruri, Tejaswy Potluri, Indira Priya Samayam, Noureen Faizee and Christie Hennen for the encouragement and advice. I would like to thank God for giving me the capacity to achieve this milestone and for all the blessings.
# Table of Contents

Abstract ........................................................................................................................... iii

Acknowledgements ......................................................................................................... v

Table of Contents ........................................................................................................... vi

List of Tables .................................................................................................................. xiii

List of Figures ................................................................................................................ xiv

List of Abbreviations ..................................................................................................... xvi

1 Introduction .................................................................................................................. 1

1.1 Discussion about features of MANET ................................................................. 2

1.1.1 Bandwidth ........................................................................................................ 2

1.1.2 Connectivity .................................................................................................... 3

1.1.2.1 Routing ....................................................................................................... 3

1.1.2.1.1 Classification of Routing Protocols ......................................................... 4

1.1.2.1.2 Table driven routing protocol ................................................................. 4

1.1.2.1.3 On-demand routing protocol ................................................................. 5

1.1.3 Security ........................................................................................................... 5

1.2 Real world components of MANET ................................................................. 7

1.2.1 Nodes ............................................................................................................. 7

1.2.2 Architecture of the node .............................................................................. 8
1.2.3 Communication Architecture.................................................................8
1.2.4 Network.................................................................................................8
1.2.5 Traffic ....................................................................................................9
1.2.6 Mobility..................................................................................................9
1.3 Conclusion...............................................................................................9

2 Problem Statement ..................................................................................11

2.1 Background on Intrusion Detection.......................................................12
2.2 Classification of IDS...............................................................................13
   2.2.1 Anomaly Detection System.................................................................13
   2.2.2 Specification Based Detection...............................................................15
   2.2.3 Misuse Detection System....................................................................15
   2.2.4 Compound Detection..........................................................................16
2.3 Related work ............................................................................................16
2.4 Problem Definition...................................................................................17
2.5 Specific needs and challenges.................................................................18
2.6 Conclusion...............................................................................................18

3 Simulator ....................................................................................................18

3.1 Introduction...............................................................................................19
3.2 Omnet ++ ..................................................................................................20
   3.2.1 Topology Description .........................................................................24
   3.2.2 Parameters..........................................................................................24
   3.2.3 Building and running simulations.......................................................24
   3.2.4 NED overview...................................................................................25
3.2.5 User interface .........................................................................................................27
3.2.6 Description of Components .....................................................................................29
3.2.7 Description of a component behavior ......................................................................30
3.2.8 Running the simulation .............................................................................................32
  3.2.8.1 Coding .................................................................................................................32
  3.2.8.2 Building ...............................................................................................................32
  3.2.8.3 Simulation ..........................................................................................................33
    3.2.8.3.1 One step..........................................................................................................33
    3.2.8.3.2 Slow execution .................................................................................................33
    3.2.8.3.3 Run ...............................................................................................................33
    3.2.8.3.4 Fast run ..........................................................................................................34
    3.2.8.3.5 Express run .....................................................................................................34
    3.2.8.3.6 Run until ........................................................................................................34
    3.2.8.3.7 Cmdenv ..........................................................................................................34
3.2.9 Result Recording in the simulator .............................................................................34
  3.2.9.1 Warm-up period ..................................................................................................35
  3.2.9.2 Output vectors ....................................................................................................35
  3.2.9.3 Scave Tool .........................................................................................................35
3.3 Conclusion .....................................................................................................................36

4 Research Approach .....................................................................................................37
  4.1 Research Solution Proposed ......................................................................................37
  4.2 Intrusion Detection .....................................................................................................37
    4.2.1 Identifying the malicious nodes ............................................................................38
5.2.3 Attack models

5.3 Conclusion

6 Discussion of Results

6.1 Detection of malicious nodes by IDS:

6.1.1 Changes by Mobility of the nodes:

6.1.1.1 Varying speed

6.1.1.2 Discussion

6.1.2 Changes by malicious node count

6.1.2.1 Varying the number of malicious nodes:

6.1.2.2 Discussion

6.1.3 Importance of voting process

6.1.3.1 Turning the voting process on and off

6.1.3.2 Discussion

6.1.4 Performance of IDS with different attacks

6.2 Effect of different parameters while varying the speed

6.2.1 Reception of packets

6.2.1.1 Discussion

6.2.2 Performance of Network Delay

6.2.2.1 Discussion

6.3 Impact of IDS

6.3.1.1 Discussion

6.4 Conclusion
7 Conclusion ............................................................................................................... 84

7.1 Conclusion ......................................................................................................... 84

7.2 Scope for future work ....................................................................................... 85

References ............................................................................................................... 86
List of Tables

1.1 Security issues at various OSI layers[18] ................................................................. 7
3.1 Comparison of Simulators [5].................................................................................. 22
4.1 Comparison among driven routing protocols[58]....................................................... 43
4.2 Comparison among on-demand routing protocols[58]............................................ 44
4.3 Comparison among On-demand and driven[58]...................................................... 45
4.4 Comparison among DSDV and AODV[59]............................................................ 46
4.5 Forwarding of MH4 [13] .......................................................................................... 49
4.6 Advertised route of MH4 [13] ................................................................................ 49
4.7 Forwarding of MH4 (updated) [13].......................................................................... 50
4.8 Route at MH4 (updated) [13]................................................................................ 50
6.1 Detection performance of IDS with different attacks .............................................. 77
List of Figures

1-1 Classification of Routing Protocols[10] ........................................................................ 4
3-1 GUI representation of a NED file .................................................................................. 25
3-2 Textual representation of a NED file ............................................................................. 26
3-3 GUI representation of .ini file ....................................................................................... 26
3-4 Textual representation of .ini file .................................................................................. 27
3-5 Graphical representation of execution environment ....................................................... 28
3-6 Textual representation of execution environment .......................................................... 28
3-7 Simple module definition ............................................................................................ 29
3-8 Compound module definition ...................................................................................... 29
3-9 Definition of code for a simple module in C++ ............................................................ 31
3-10 Graphical output of hop count ..................................................................................... 36
4-1 Mobile host movement in ad hoc networks[13] ............................................................ 48
4-2 Process of route finding in AODV Routing Protocol ..................................................... 55
5-2 Voting process .............................................................................................................. 60
6-1 Detection of malicious nodes with varying speed ......................................................... 72
6-2 Detection of malicious nodes by varying malicious node count .................................. 74
6-3 Effect of voting process ............................................................................................... 75
6-4 Transmission of packets .............................................................................................. 79
6-5 Transmission of packets with higher number of packets ............................................. 79
6-6 Performance of Network Delay ................................................................. 80
6-7 Performance of Network Delay with more simulation time .................. 81
6-10 Variation of Throughput with Delay ...................................................... 82
List of Abbreviations

AODV .........................Ad hoc On-demand Distance Vector
DSDV .........................Dynamic Sequenced Distance Vector Routing
DSR .........................Dynamic Source Routing
DTQ .........................Data transmission Quality
IDS .........................Intrusion Detection System
MAC .........................Medium Access Control
MANET ......................Mobile Ad-hoc NETwork
MPS .........................Meters per second
NED .........................Network Description
OSI .........................Open System Interconnection
STB .........................Stability Factor
Chapter 1

Introduction

An Ad-hoc network is a network with a group of mobile nodes/hosts which will communicate with each other via wireless links as long as they are within the radio range. The term “ad hoc” means, nodes that are self-organized which means that they do not have a central entity to govern them [1]. So, that’s how the name mobile ad hoc network (MANET) was formed. Unlike networks which are using dedicated nodes to support some of the basic functions like routing, packet forwarding and network management, in adhoc networks these are carried out by all nodes [2]. Nodes that present in an adhoc network move in all different directions with any speed but still they are connected to the network because of the wireless links. These ad-hoc networks do not have any kind of fixed infrastructure and are also called by the names MANET and adhoc networks.

Each node acts like both a host and as a router at the same time in order to do both transmission and reception in a network [3]. As the nodes keep moving in the network, the topology of the network changes frequently and it is not predictable. Whenever a
node wants to communicate with another node which is out of its radio range, the cooperation of other nodes in the network is needed which is known as the multi-hop communication. Because of this MANET possess a need to change their routing based on the presence of the nodes.

The authors in [4] have proposed a learning-based algorithm for automatically computing anomaly detection models which are based on the correlations among a large set of features.

Some of the nodes in the MANET are operated on battery and also their energy is very limited. So, running the normal operations of a MANET by conserving energy depends upon the design and a better implementation of the MANET. Along with this some other challenges for a MANET are bandwidth, quality of service, network topology and security.

There are a wide range of applications for the MANET, ranging from classroom/meeting room applications, data collection networks and even in military applications.

1.1 **Discussion about features of MANET**

The features can be broadly classified in terms of security, connectivity, bandwidth and battery lifetime etc. These features will be discussed further in the sections below.

1.1.1 **Bandwidth**

Bandwidth as always is one of the important factors of MANET, connectivity and data exchange in the network and maintenance of the network always depend on the
bandwidth of the network. If the data communication/other connection establishment activities in the network have consumed the available bandwidth for the network then it is tough to establish newer connections in the network or the existing connections may not be re-established when the mobile nodes are trying to relocate themselves [5]. The effective utilization of the bandwidth for a network can be optimized through the routing activity that takes place in the network. The performance of the network is always directly proportional to the bandwidth of the network.

1.1.2 Connectivity

The data transmission from one layer to another layer, one node to another node or any kind of transmission of data is based on the connectivity that is present in the network. In MANET, the connectivity is always a multifaceted problem because of the network topology. In wired networks, the connectivity is centered on the dedicated switches, routers and gateways [5]. Connectivity of the network also depends upon the various activities that take place in the OSI layers, especially between the communication layers like physical, data link and the network layers. The connectivity between two nodes in a MANET is completely different from a wired network. The neighboring nodes always keep changing in a MANET whereas in the wired network that remains constant. So, because of this reason the routing plays an important role for connectivity of the network. The following sections in this chapter will throw some light on routing in detail.

1.1.2.1 Routing

This section is to explore routing in detail. MANET, does not have any kind of fixed infrastructure for routing, so nodes just relay packets for each other. So, the nodes
will be able to communicate only if they participate and cooperate in routing and forwarding [6]. There are numerous people who are trying to standardize everything related to routing protocols in MANET. For example, it is discussed in detail in [7], that the working group is trying to evolve MANET routing specification(s) and then introduce them to the Internet Standards Track. There are also different studies in detail about the approaches to the same [8].

1.1.2.1.1 Classification of Routing Protocols

Classification of routing protocols can be done in many ways, but most of them are done basing upon the network strategy and network structure [9-11]. Based upon the routing strategy, the routing protocols may be generally categorized as follows [10] and also as shown in the Figure 1-1.

1. Table driven
2. Source-initiated (demand driven)

![Figure 1-1 Classification of Routing Protocols][10]

1.1.2.1.2 Table driven routing protocol

These protocols maintain the routing information even before it is needed so they are called as “Proactive routing protocols” [12, 13]. In a network every node maintains
routing information about every other node in the network. This information is usually kept in the routing tables and is always updated as soon as the topology changes in the network. Most of these routing protocols come from the link-state routing [9].

The routing protocols which come under this category differs from each other and the difference is depending upon how the routing information is being updated in their respective routing tables and they also maintain different routing tables. As this routing protocol need to maintain the node entries for each and every node in their respective routing tables, they are not suitable for larger networks. Maintaining of node entries in the routing tables causes more overhead to the routing tables which leads to consumption of more bandwidth [10].

1.1.2.1.3 On-demand routing protocol

This routing protocol is different from the above mentioned table driven routing protocol. This routing protocol creates the routes from the source node to the destination node only when they are desired, which means in the case if there is no communication between the nodes then there will be no routing information. These kind of routing protocols are also called as “Reactive routing protocols”. This protocol searches for the route only when there is a demand for it (which means only when a node wants to send a packet to another node), and then it establishes the connection for transmit and receive the packet [10, 14].

1.1.3 Security

The need for security in MANET is very high because there is no fixed infrastructure for the network and the nodes are mobile with open and dynamic structure.
The most important parameters that security depends on are authentication, integrity, confidentiality, availability and non-repudiation [15].

The wireless adhoc networks need more security because it is more vulnerable to attacks by design. The use of wireless links makes an adhoc network more susceptible to attacks ranging from passive eavesdropping to active interfering [16, 17]. Unlike in wired networks, where an attacker must gain physical access to the network wires or pass through the several lines of defense like firewalls and gateways. When compared to a wired network, it’s easier to attack a wireless network because of its structure and also the attack may come in any direction and any node can be attacked at any point of time.

So, that’s the reason each and every node in the network has to prepare for attacks at any point of time. And also as there is no central based controlling identity for the participating nodes; the attacks are much easier to launch in MANET.

Attacks in MANET can occur at different levels of the OSI layers and they can be given as follows [18] in the Table 1.1. For any communication network, usually the first line of defense is robust encryption [19]. Protecting the routing data has been studied and also many solutions have been proposed at the network level in [20-23]. Especially the authors of [20, 21], deal with strengthening the route discovery process or introducing new methods to select an efficient route from the available routes from the same source to the same destination [23]. The authors in [24] discussed about the data in link layer, physical layer, data protection and adhoc layers and also about protecting the physical layer protocols by addressing their features of the above mentioned layers which makes them vulnerable.
Table 1.1 Security issues at various OSI layers[18]

<table>
<thead>
<tr>
<th>Name of the layer</th>
<th>Security issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application layer</td>
<td>Detecting and preventing viruses, worms, malicious codes and application abuses.</td>
</tr>
<tr>
<td>Transport layer</td>
<td>Authenticating and securing end-to-end communication through data encryption</td>
</tr>
<tr>
<td>Network layer</td>
<td>Protecting the ad hoc routing and forwarding protocols</td>
</tr>
<tr>
<td>Link layer</td>
<td>Protecting the wireless MAC protocol and providing link-layer security support</td>
</tr>
<tr>
<td>Physical layer</td>
<td>Preventing signal jamming denial-of-service attacks.</td>
</tr>
</tbody>
</table>

All of the above mentioned approaches attempted to improve the capability of the network to fight against the malicious attacks by examining their features which makes them vulnerable. And most often they result in becoming an overhead process to the bandwidth consumption, usage of power and retrieving of data. In order to avoid this kind of overhead, the activities of the node can be monitored, the malicious intent has been detected as soon as it enters the network and finally it can be eliminated which results in avoiding the mentioned overhead problems. This is the place where “intrusion” comes in.

1.2 Real world components of MANET

This section gives us some details about the real world components of MANET and one should know them before choosing to model them in a simulator and the components of these networks can be broadly divided as follows:

1.2.1 Nodes

A component which has the capabilities of transmitting, processing and receiving of information is called as node or host. In real world, these are powered by on-board
batteries. They also contain sensors and navigation devices which perform different varieties of functions start like storing the data to intelligent aggregation of data.

1.2.2 Architecture of the node

Nodes/hosts are the main communication devices (i.e., radio devices) and their internal architecture is defined in terms of the OSI protocol stack and the protocol layers are physical, data link, medium access control (MAC), network, transport, session, presentation and application layer. Even the implementation of the networks with the simulators is similar to the real time counterparts.

1.2.3 Communication Architecture

As the term, “mobile ad hoc” means the nodes which are mobile and those doesn’t have a central entity to maintain/organize them. From this we know that the nodes are independent and they use wireless communication in order to communicate with other nodes in the network, but the nodes will be able to communicate only when the other nodes lie within the radio transmission range. The physical properties of a node are transmission and reception.

1.2.4 Network

This is the place where are the nodes/hosts are present and obviously only a group of them is called as a network. Always there will be physical boundaries for every network. For example, all the cell phone service providers will have their network coverage only limited to the country that they exist. Even if they offer service outside the physical territory of their country, they borrow of the service where the mobile user is
actually present. The count of the nodes will not be constant as they keep moving so nodes will be leaving and entering the network at any time.

1.2.5 Traffic

The process of moving data from one node to another node or transfer of information from one layer to another layer (according to OSI protocol layers) is termed as traffic.

1.2.6 Mobility

As the term MANET, has mobile in it which means the nodes are mobile and are free to move in any given direction. But the simulators follow certain types of mobility models such as linear mobility, restricted linear speed mobility and so on. In real world they don’t have any specific model for their mobility but for simulators we have to use certain mobility models which are almost similar to different mobility patterns of the real world.

1.3 Conclusion

In this chapter, we have given an introduction about the classification, features have been discussed and also the problems that MANET are facing to owe the characteristics have also been discussed. This thesis has been organized in such a way that, Chapter 2 talks about the Intrusion Detection, Intrusion Detection System (IDS), solutions that are already available and also defines our research problem. Chapter 3 deals about the simulators and also about the choice of our simulator. Chapter 4 discusses about our research approach. Chapter 5 gives the details of the solution. Chapter 6 discusses about the collection of simulation data and the discussions of results. Chapter 7
gives about the conclusion of this research and also discuss about the future scope of this work and then finally Chapter 8 is about the list of references.
Chapter 2

Problem Statement

Intrusion can be defined as, any kind of unwanted or unexpected activity happened in the network which is affecting the integrity, confidentiality or using a network resource without its prior permission. A system which is used to find out these abnormal behaviors in this network is called as “Intrusion Detection System (IDS)” and the way that it does the actions can be termed as “Intrusion Detection”.

There are many approaches for Intrusion Detection in MANET. Mainly there are two important classifications of IDS and are namely behavior based and authentication based. Both the classifications give a brief thought about them by their names and in detail the former one is completely based on the behavior of a node and its nodal activities whereas the later one is based on authenticating the identity of a node and the usage of encryption keys (public key and private key pairs) falls into this category [5, 16, 25, 26]. The former approach is behavioral based algorithms where intrusion is defined based upon its nodal activities instead of its identifier. According to us, this is a better approach because of the following reasons [5, 27]:

1. Behavior of a node is very tough to replicate.

2. No need of storage of identities.
3. As the node identifiers should be unique, the process of deployment will become expensive both in terms of time and cost.

Basing on the above reasons, our focus will be on behavior-based intrusion detection, since we think this is more efficient, lightweight and easily scalable to Intrusion Detection in MANET. Intrusion Detection, Intrusion Detection System (IDS) and its classification is further discussed in this chapter.

2.1 Background on Intrusion Detection

In terms of the authors at [25] Intrusion can be defined as a process or a set of actions which attempt to compromise the confidentiality, integrity and availability of a resource occurs. Then intrusion can be prevented with some techniques like encryption and authentication (e.g. encryption keys, passwords, and fingerprints) are the first line of defense. But just the prevention is not only sufficient and there should be a strong defensive system which will protect the network even it passes the above mentioned techniques.

In other words, IDS can be used as a second line of defense to protect the network systems. As always security is the primary and the important concern for any network and maintaining the security of the network will be even more complex in proportion to the growth in the size of the network. The necessity of security is even more in MANET because there is no fixed infrastructure, lack of centralized entity and dynamically changing network topology.

Security attacks exist mainly of two types and they are insider and outsider attacks. Prevention techniques such as authentication and encryption are mainly used for the outsider attacks and to deal with insider attacks, intrusion detection systems (IDS)
techniques have been developed [16]. Intrusion Detection can be mainly classified basing on the audit data as host based or network based. The former system uses the operating system and the logs for analysis whereas later system is dealt with capturing and analyzing packets from the network traffic [26, 28].

Huang and Lee in [29] proposed a cluster-based detection approach. The idea is to elect a node, the clusterhead and to perform the functions of IDS for all nodes which are within a cluster. They had presented cluster formation protocols that achieve fairness and security in that clusterhead election and they had achieved all this with two feature computational schemes.

2.2 Classification of IDS

Basing on the detection techniques, IDS can be classified into three important categories: anomaly detection system, misuse detection system and specification detection system [5, 17, 25, 26, 28, 30]. Further sections in this chapter will give some more detailed explanation about the above mentioned detection systems.

2.2.1 Anomaly Detection System

This detection defines a profile of normal behavior of nodes/users and stores them in a system. This is one of the earliest approaches to intrusion detection [31]. Any kind of deviation in the profile for a user can be detected as a possible intrusion. There are both pros and cons for this detection system. The pros of this approach are as follows:

1. Any kind of small deviation to the profile is detected as a possible intrusion.

2. It updates the possible intrusion to the system administrators.

3. It doesn’t require any knowledge about the possible intrusion.
There are even some cons about this detection system and they are:

1. It cannot describe about the attack clearly.
2. It may exhibit previously unknown attacks, but exhibits high rate of false positives.
3. If the user is not behaving as it should on its own, even that is considered as an intrusion, which is also called as false negatives.

For an example as mentioned in [32], the normal profile of a user consists of averaged frequencies of some system commands used in their login sessions. If a session is being monitored, and when the frequencies are significantly lower or higher, even then the alarm will be raised with no useful result at the end.

The authors in [17] had proposed a distributed and cooperative intrusion detection model in their work which is based on statistical anomaly detection techniques. In that system, every node runs and IDS agent runs which performs local data collection and detection, whereas global intrusion response and cooperative detection can be triggered only when a node report anomaly. They had considered two scenarios as attacks namely – detecting abnormal activities in layers other than the routing layer and sudden changes in the routing tables which formed the definition of the anomaly.

Anomalies in packet forwarding like dropping, modifying and/or delaying of packets by intermediate nodes have been observed by the IDS in the design of it mentioned by the authors in [33].
2.2.2  Specification Based Detection

This detection can be defined as a set of constraints that describe the correct operation of a program or protocol and monitor the execution of a program or protocol with respect to the defined constraints.

According to the authors in [30], the detector works by detecting the intrusion against the background of the normal traffic in the system as these detectors have a better chance of correctly detecting truly interesting events in the supervised system, since they both know the patterns of intrusive behavior and can relate them to the normal behavior of the system. At least they are able to support their decisions on detecting better than the rest.

The authors in [34] stated that, the specification based detection system has showed low rate of false positives. In the online evaluation they were able to detect the attacks without any false positives but whereas in the offline evaluation their system was able to detect only 80% of the attacks.

2.2.3  Misuse Detection System

In this misuse detection system, [26] the system keeps a record/track of all the signatures from their previous known attacks and uses them to compare with the present captured data [35]. If any kind of matched pattern is observed then it is treated as intrusion. Best example for this can be virus detection system, which cannot detect new kinds of attacks. In an architecture proposed by the authors at [36], suggested the implementation of a Local Intrusion Detection System (LIDS) agents on each node. Some examples of misuse detection systems in IDIOT [35], STAT[31], [17, 37] have
used patterns of well-known attacks or weak spots of a system to match and identify the intrusions.

According to the authors in [17], an example for a signature rule for the “guessing password attack” can be “if there are more than 4 failed login attempts within 2 minutes”. The main advantage of this misuse detection system is that it can efficiently and accurately detect the instances of all known attacks and the main disadvantage is that it doesn’t have the ability to detect the truly innovative (i.e., newly invented) attacks [17].

2.2.4 Compound Detection

Compound Detection is derived from the combination of misuse and anomaly detection systems but mostly a misuse inspired system which forms a compound decision in view of a model of both the intrusive behavior and the normal behavior of an intruder[30]. The intrusion which is occurring at the background of the normal traffic in the system is detected by operating a detector. These kinds of detectors play a good role in detecting truly interesting events in the supervised system, because they know the patterns of the intrusive behavior and they are also capable of relating them to the normal behavior [30].

The authors in [38] have proposed that an IDS that uses a quantitative method of anomaly definition which is based on the transmission characteristics, but factors in the historical transmission behavior of the node.

2.3 Related work

The authors in [39, 40] suggested a method to detect intrusion detection in MANET using the ensemble methods. In [39] they had used a three-level hierarchical
system to serve three different purposes namely data collection, processing and transmission. Each node in the MANET has attached to Local Intrusion Detection Systems (IDSs) which collects raw data of the network operation, and computing a local anomaly index which measures the mismatch between the current node operation and a base line of normal operation.

The anomaly indexes which are being collected from the nodes and all the nodes belonging to one cluster are periodically reported to their cluster head, then the cluster heads produces cluster-level anomaly indexes by averaging their collected data and then these cluster heads again periodically transmit the cluster level anomaly indexes to a manager who averages them. Even though it is a distributed solution for intrusion detection, it still have the central entities which are called as the cluster heads and because of the presence of these central entities it makes a central point of failure- the cluster head may not be functional if it gets attacked and all the nodes which are below the cluster head become unuseful. And also these central nodes consume more resources (because of their complex logic) and which in turn decreases the survivability of the network. The authors at [38] has suggested a non-centralized solution, but it do not cater to mobile nodes or MANET.

2.4 Problem Definition

Our challenge is to find a distributed, quantitative and dynamic intrusive detection solution for MANET which involves mobile nodes in a non-cluster based environment.

In addition to this, our other challenge is to develop simulations for MANET which includes as follows:

1. Implementation of the IDS.
2. Implementation of a suitable routing protocol and mobility model.

3. Physical layer which meets the IEEE 802.11 standards.

2.5 Specific needs and challenges

This section of thesis explains about the challenges of the research in more detail which will help us in explaining the solutions for the problems that were faced in developing the intrusion detection system.

Firstly, how do we identify the nodes which are displaying the malicious behavior in the network?

Secondly, existence of central entities in our security solution because of this the survivability of the network depends. So, is our proposed method completely distributed?

Finally, is it possible to change the properties of the IDS or are the properties of the IDS configurable easily?

The answers for the above mentioned questions will be answered in the future chapters of this thesis.

2.6 Conclusion

In this chapter we have discussed about the problem definition of this research, intrusion detection, classification and explanation of the existing detection systems, needs and challenges of the research and finally about the needs of the simulator.
Chapter 3

Simulator

3.1 Introduction

Simulation can be defined in dictionary terms as, the imitation of some real thing, state of affairs, or process. The act of simulating something generally entails representing certain key characteristics or behaviors of a selected physical or abstract system. Simulations are also used to model for scientific modeling of natural systems or human systems in order to gain insight into their functioning [41]. It is also be used to demonstrate the eventual real effects of alternative conditions and courses of action. The other scenarios that simulation can be used are when the real system cannot be engaged, because it may not be accessible, or it is dangerous or unacceptable to engage, or it is being designed but not yet built or it may simply doesn’t exist in real. So, the usage of the simulator is a compulsion and it is highly impossible to do these kinds of tests without a simulator because the research includes the mobile nodes and the wireless network.

One important feature of the simulator is the format of the input that is fed to the system and the data that we get from the simulator (output that we obtain from the simulator). Both the input which has been fed to the simulator and the output that we get from the simulator are equally important.
Some of the key issues in simulation do include acquisition of valid source information about the relevant selection of key characteristics and behaviors, and also the use of simplifying approximations and assumptions within the simulation, and fidelity and validity of the outcomes.

We do require simulation in our research as it deals with the wireless networks and also the use of radio-based transmission. It is practically impossible to implement all these things in real so, it is essential to implement and test all these with the help of a simulator. By changing the variables, predictions may be made about the behavior of the system [41]. Simulators often allow the users to implement protocols for transmission, reception, propagation and other communication aspects to work with their “ether”. With the use of simulators it is easy to analyze the results so that we can work on how to improve the model performance which will eventually allow us to concentrate more on the research instead of the practical/physical implementation of the network model.

This chapter deals with the explanation about the simulator that we have chosen and its specific characteristics, properties and advantages over other competing available simulators.

Basing upon the Table 3.1 by the author at [5], we have chosen Omnet++ as our simulator and the version that we are using for this research is Omnet++ 4.1. The following sections in this chapter give more details about the software.

3.2 Omnet ++

Omnet ++ is an open architecture, open source and object-oriented discrete event
network simulation framework [42]. It can be used (has been used) in several problem domains:

1. Modeling of wired and wireless communications.
2. Protocol modeling

And in general, modeling and simulating of any system where the discrete event approach is suitable.

Omnet ++ itself is not a complete simulator of anything concrete, but it rather provides infrastructure and tools for writing simulations. Its components can be defined as a nested hierarchical structure which can be defined either through simple text files or using GUI which is very easy to learn, in addition being very expressive.

The behavior of the components of a module can be declared using C++ functions. It also has a basic output analyzer which can display in graphical formats. The basic infrastructure of this software is very extensible and is easy to modify. Based up on the table 3-1, comparatively Omnet ++ is far better than its competitors. An Omnet++ model consists of modules which communicate with message passing. The active modules are termed as simple modules which are written in C++, using the Omnet ++ simulation library. The number of hierarchy levels is unlimited. Models can be assembled from reusable components and can also form compound modules, any well-written modules are truly reusable and they can just be combined like LEGO blocks to form different things[42].
Table 3.1 Comparison of Simulators [5]

<table>
<thead>
<tr>
<th>Factor</th>
<th>Opnet</th>
<th>Glomosim/QualNet</th>
<th>NS-2</th>
<th>OMNET ++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Architecture</td>
<td>Hierarchical</td>
<td>Layered</td>
<td>Layered</td>
<td>Hierarchical, nested</td>
</tr>
<tr>
<td>Component/topology definition</td>
<td>Finite State Model</td>
<td>Skeleton Interface files</td>
<td>OTcl object model</td>
<td>Text files</td>
</tr>
<tr>
<td>Topology definition language/Model</td>
<td>Proto-C, OO models</td>
<td>Parsec, C</td>
<td>C ++, OTcl</td>
<td>Flat files, C++</td>
</tr>
<tr>
<td>Topology definition language/Model</td>
<td>GUI based editor, Proto-C</td>
<td>Flat files</td>
<td>OTcl based files</td>
<td>Flat files</td>
</tr>
<tr>
<td>Input/output definition</td>
<td>GUI customizable</td>
<td>Flat files (GUI in QualNet)</td>
<td>OTcl interfaces and script files</td>
<td>Flat files, GUI</td>
</tr>
<tr>
<td>Experiment setup</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
<td>Available</td>
</tr>
<tr>
<td>OSI layers</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mobility models</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Radio propagation models</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 3.1 Continued

<table>
<thead>
<tr>
<th>Factor</th>
<th>Opnet</th>
<th>Glomosim/QualNet</th>
<th>NS-2</th>
<th>OMNET ++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic generation</td>
<td>Available</td>
<td>Unknown (Java-based trace, monitoring and statistics reporting)</td>
<td>Available</td>
<td>Not available</td>
</tr>
<tr>
<td>Traffic profiling</td>
<td>Available, GUI based</td>
<td>Moderate. Less document-ation (QualNet is better)</td>
<td>Tcl/Tk based analyzer for Trace files</td>
<td>GUI based basic inspectors.</td>
</tr>
<tr>
<td>Usability</td>
<td>Good</td>
<td>Good</td>
<td>Sparse GUI, tough to debug, large footprint</td>
<td>Good</td>
</tr>
<tr>
<td>Modifiability</td>
<td>Moderate</td>
<td>Not so easy</td>
<td>Complex</td>
<td>Good</td>
</tr>
<tr>
<td>Extensibility</td>
<td>Yes</td>
<td>Yes</td>
<td>Not easy</td>
<td>Easy</td>
</tr>
<tr>
<td>Scalability</td>
<td>Yes</td>
<td>Yes</td>
<td>Not Good</td>
<td>Good</td>
</tr>
<tr>
<td>Licensing</td>
<td>Commercial (discounted for universities)</td>
<td>Free for universities (QualNet commercial)</td>
<td>Public Domain</td>
<td>Commercial for Business Reasonable</td>
</tr>
<tr>
<td>Scientific acceptance</td>
<td>Reasonable</td>
<td>Good</td>
<td>Very good</td>
<td>Reasonable</td>
</tr>
</tbody>
</table>
3.2.1 Topology Description

The structure of a model is defined in NED (Network Description) language.

3.2.2 Parameters

Parameters for the modules can be assigned either using the NED file or by using the `omnetpp.ini` configuration file. These parameters can be of string, numeric or Boolean values. User can put numeric values including expressions using other parameters and also calling C functions.

3.2.3 Building and running simulations

Any module in Omnet++ will consists of the following parts [42]:

1. NED language topology description(s) which are the `.ned` files, describes about the structure of a module with parameters, gates etc., these can be edited using both graphically and also using text based files.

2. Message definitions (`msg`) files are used to defined various message types and also add data fields to them. These message definitions are further translated into full-fledged C++ classes by Omnet ++.

3. Simple module sources, these are C++ files with `.h` and `.cc` suffix.

The simulation system of Omnet ++ provides the components as follows[42]:

1. Simulation kernel library, this contains the code which manages the simulation class library and the simulations and it is written in C++ and then compiled into a shared library.

2. User interfaces, these are used during the simulation execution, to facilitate demonstration, debugging or batch execution of the simulations.
3.2.4 NED overview

As described earlier in 3.2.1, NED stands for Network Description. Omnet ++ provides its infrastructure to combine the simulations from the .ned files and the .ini files which are used for configurations. The NED files can be edited both graphically using GUI and also using text based editor. The graphical (GUI based) representation and the textual representation are shown for an examples in the Figure 3-1 and Figure 3-2 respectively.

Figure 3-1 GUI representation of a NED file
Figure 3-2 Textual representation of a NED file

Figure 3-3 GUI representation of .ini file
The NED file is the description of a network which has 6 “tic” sub modules and which keeps transferring the message from one node to another. Simulation runs for a network can be configured only using the omnetpp.ini files and they are also simple text files too. In fact, the omnetpp.ini files also can be viewed in two different ways as shown in the figures below. The Figure 3-3 is the GUI based representation of the omnetpp.ini file and the Figure 3-4 is the text based representation of it.

### 3.2.5 User interface

The main purpose of the user interface is to make the internals of the network visible to the user, so that the user can control the simulation execution. Omnet ++ provides runtime user interface and environments for simulations and viz., Tkenv and cmdenv. The user can run the simulation according to their preference either from cmdenv or Tkenv and they can specify this in the omnetpp.ini itself. The graphical representation of the network while it is in the execution environment is as shown in the Figure 3-5 and the textual representation is as in the Figure 3-6.
Figure 3-5 Graphical representation of execution environment

Figure 3-6 Textual representation of execution environment
3.2.6 Description of Components

In this section, description of components has been done in a detailed way.

```
simple Txc
[
parameters:
  @display("i=block/routing");
gates:
  input in[];  //declare in[] and out[] to be
vector gates
  output out[];
]
```

Figure 3-7 Simple module definition

A simple module can be defined as shown in the Figure 3-7 and from that figure we can understand that, a simple module has been defined with the name Txc (=transmission) and it has a symbol with two gates in[ ] and out[ ] and they are declared to be as vector gates.

A network module is formed by combining simpler components like Txc with connections to all of them and also with a delay for each transmission and is as shown in the Figure 3-8.

```
network Tictoc
[
submodules:
  tic[6] : Txc;
connections:
  tic[0].out++ --> [delay = 100ms] --> tic[1].in++;
  tic[0].in++ <-- [delay = 100ms] <-- tic[1].out++;
  tic[1].out++ --> [delay = 100ms] --> tic[2].in++;
  tic[1].in++ <-- [delay = 100ms] <-- tic[2].out++;
  tic[1].out++ --> [delay = 100ms] --> tic[4].in++;
  tic[1].in++ <-- [delay = 100ms] <-- tic[4].out++;
  tic[3].out++ --> [delay = 100ms] --> tic[4].in++;
  tic[3].in++ <-- [delay = 100ms] <-- tic[4].out++;
  tic[4].out++ --> [delay = 100ms] --> tic[5].in++;
  tic[4].in++ <-- [delay = 100ms] <-- tic[5].out++;
]
```

Figure 3-8 Compound module definition
3.2.7 Description of a component behavior

In Omnet++, it allows us to define the components using the NED language which will be further defined using C++. Even for a simple module which doesn’t have any kind of nested components for it also has to be defined using the C++ definition. The code which is in Figure 3-9 is just for an example.
/**
 * what we do here is keep the original packet and send only copies of it. We delete the original when toc's acknowledgement arrives. To make it easier to visually verify the model, we'll include a message sequence number the message * names.**/

namespace myrealnetwork {

Define_Module(Txc);

void Txc::initialize()
{
    char msgname[20];
    sprintf(msgname, "tic-%d", getIndex());
    cMessage *msg = new cMessage(msgname);
    scheduleAt(0.0, msg);
}

void Txc::handleMessage (cMessage *msg)
{
    forwardMessage(msg); // We need to forward the message
}

void Txc :: forwardMessage(cMessage *msg)
{
    //In this, we just pick a random gate to send it on.
    //We draw a random number between 0 and the size and the gate
    //"out[]"
    int n = gateSize("out");
    int k = intuniform(0,n-1);
    EV << "Forwarding message" << msg << " on port out [" << k << "]\n";
    send (msg, "out",k);
}; // namespace

Figure 3-9 Definition of code for a simple module in C++

3.2.8 Running the simulation

During the simulation there are different options in running the simulation available for the user in Omnet ++ depending upon the network that the user is working. There are different stages before the user can actually run a simulation for any network. Those stages can be broadly classified namely, coding, building and then simulate. A brief description about each classification is given below:

3.2.8.1 Coding

The user has to decide first on how the network should look like and also decide on considering the necessary components for the desired network which has to be implemented with the use of the simulator. Once the user has decided on the above mentioned things then the user has to create appropriate files and among them there are some compulsory files which have to be created are .ned, .msg files.

After creating these two kinds of files then NED tool will automatically create name_n.cc files after it reads the .ned files. After this then the user has to create .cc files and then it automatically creates .h header files, last but not the least file to be created is the omnetpp.ini file which is used to set the configuration of the network. With the creation of all these files coding has been completed.

3.2.8.2 Building

This plays a leading role in the process of implementing a network in the simulator. Once the user has done with the coding phase then the next action to be taken is to build the project. Using the command prompt with the tool “mingwenv”, after going to the directory where the current project is present and then run the command “opp_makemake”, which is building the project in the command prompt; this can also be done in GUI by “build project” option. After building the project in
any of the above mentioned ways then the header (.h) files, Makefile, libraries and executables.

At this stage, the simulator will display if there are any errors after building the project and it is possible to run the simulation only after fixing the errors (if any). And also whenever you make any changes to the code, it's a compulsion to build the project, otherwise the new changes which have been made will not be considered and finally will not be displayed in the output of the simulation.

### 3.2.8.3 Simulation

Finally, after building the project then it is ready for the simulation. Omnet ++ provides the option of simulating a project on basing different speeds of the simulation. Those different options are briefly described below:

#### 3.2.8.3.1 One step

In this, the execution of the simulation is done for only one step in the whole process. After executing for one step in the simulation the simulation stops until the user specifies the next task.

#### 3.2.8.3.2 Slow execution

When this option is chosen for running the simulation, then the simulation will run in a very slow mode and the user can monitor the execution more carefully.

#### 3.2.8.3.3 Run

In this mode, the simulation runs a little bit faster than the above mode. Usually this is the mode that gets activated whenever the user runs the simulation for any network.
3.2.8.3.4 Fast run

This mode is a little bit advanced than the “Run” mode; the display updates are very rare in this mode of simulation.

3.2.8.3.5 Express run

In this mode of simulation, the tracing is completely not visible or in other words, there is no animation provided in this mode.

3.2.8.3.6 Run until

This is the last mode among the available modes for running a simulation and in this the user can set the values to run the simulation until a particular interested event.

3.2.8.3.7 Cmdenv

This is not a GUI and it is a command line environment mode. In this mode, the results will be visible only as command lines and display the output. This mode is really useful to run the simulations in parallel with different runs and different configuration parameters.

The description about running the simulations ended here. For cross reference the user may also check some demonstration videos which are available in the Omnet++ website and these are really helpful for the implementation of the project using a simulator [43].

3.2.9 Result Recording in the simulator

Omnet ++ provide limited options to view what has happened during the simulation of the network which was designed by the user. All the data will not be saved after the simulation has been done, only incidents of those events will be stored in the software and whenever the user requests to plot the graph (limited) from the
obtained results only then the whole data will be downloaded to the module, so that’s the reason it takes some time to show the graphs from the data if they are available.

### 3.2.9.1 Warm-up period

There is a feature called warm-up period available in Omnet ++. With this feature, the initial time (warm up) time of the simulation will not be considered and during this warm-up period output will not be counted. This option is really very useful for steady state simulations. The default option for this is 0sec. But this can be changed (most cases) according the necessity of the user.

### 3.2.9.2 Output vectors

Recording of these output vector usually need a very large amount of data. Examples for output vectors (time series data) are end-to-end delays, number of packets sent or received, round trip times, hop count of messages. User can use this output vectors if they want to record any data necessary during the process of simulation.

### 3.2.9.3 Scave Tool

This scave tool is used for basic processing and filtering of the results obtained from the simulation. This tool is not capable of any graphics but it can be used to export the results from the simulator so that they can be digestible for other tools [42]. The previous versions of Omnet ++ used to have a tool called Plove and it is not available anymore. An example for the graphical output for hop count in a network is as shown in the Figure 3.10.
The above Figure 3.10 shows the number of hops taken by a message to reach from node 1 to node 5. It clearly shows all the hops in detail where in the figure each line represents each node in the network so in the above figure the message has taken 15 hops to transfer the message from its source (node 1) to its destination (node 2). The graph which is shown in the Figure 3-10 is just for demonstrating how the data can be taken out from the simulation results and to show how the data can be plot on a graph for better understanding and analyzing the results.

3.3 Conclusion

In this chapter we have discussed all the requirements and characteristics of a good simulator for mobile networks. Comparison of different simulators which are competitive in this field and a table is also updated which shows the major differences between the available simulators. And also in the subsections of 3.2 we have deeply discussed about our chosen simulator Omnet++ and discussed things like topology, parameters, running simulations with all the available options in the simulator, user interfaces i.e., both the command line and the GUI, description of components and their behavior and finally about result recording.
Chapter 4

Research Approach

4.1 Research Solution Proposed

The proposed solution to our research challenge is discussed in detail in this chapter. This solution is based on the quantitative intrusion detection techniques which have been proved in [5, 38], but the solution is applied to a MANET which contains mobile nodes.

The main challenges to develop simulations for MANET are broadly classified into four parts. They are as follows:

1. Intrusion Detection
2. Availability of mobility models
3. Availability of routing protocol implementation
4. Physical layer with Ieee standards for 802.11

4.2 Intrusion Detection

Intrusion Detection has been clearly explained in the second chapter of this thesis. Especially to deal with the insider attacks of a network, IDS techniques have been developed for detecting compromised nodes and also removing malicious nodes from the network in order to receive high survivability of the network [16, 44] and
also to make the data secure. Pattern recognition approach is also a kind of approach which is used for intrusion detection [40].

We are following the behavioral based detection for our IDS. In support with this the detection of malicious nodes can be done in two steps. First we need to identify which nodes are displaying the malicious/abnormal/unexpected behavior in the network and once we get the suspicion about the nodes which are in the network and then our process justifies its suspicion i.e., finalizing whether the suspicious node is the malicious node or not.

4.2.1 Identifying the malicious nodes

The term “malicious” signifies that something is wrong which has been termed as malicious whatever it could be. In this situation, it applies to a node(s) which are displaying this behavior and the process of identifying those node(s) is called as “Identifying the malicious node(s)”.

This whole process can be broadly classified into two major steps and they are as follows:

1. Recognizing a suspicious node
2. Confirming that the node is malicious

In the following sections of this chapter, we will discuss about these above mentioned steps in detail.

4.2.1.1 Recognizing a suspicious node

The scope of this current research allows us to define the nodes to be termed as “malicious” when any node in the network is observed to have a different behavior than the regular behavior. Li, T., Dr.Alam et al, in [38] have proposed a method that the nodes are expected to acknowledge the messages that they had received and also every node measures the acknowledgements that it has received and they have
calculated that and the value is a measure of the near-term behavior. After a certain period of time this calculated value is called as the “Stability” of the nodal behavior and is referred to as “STB” from now on. With this the transmission quality of data is also calculated and it called as “Data Transmission Quality” (DTQ).

In current research, the transmission of packets is considered as either a packet has been transmitted completely or the packet has not be transmitted at all which is like on and off of a switch i.e., either 1 or 0. But there will be nothing like partially transmitted packet [5]. Each node calculates this “DTQ” value and also maintains it for its neighboring nodes (= nodes which are only in the transmission range for a node). If there is a fall of the DTQ value which is less than the threshold then that particular node can be a malicious node in the network. And even the threshold value will be updated periodically in the network.

4.2.1.2 Confirming that a node is malicious

This is the second step in the process of identifying a malicious node in the network. After the successful completion of the process in the above step, the confirmation that a node is malicious has to be done in this step. The process of confirming that a node is malicious is done by voting process. The node which observed suspicious activity on any other node will start the voting process in that network about that node. Depending upon the votes that it receives from the neighboring nodes in the network, the suspicious node will be either will continue its stay in the network or it will be out of the network (which means blacklisted) and finally be removed from the network as it will be confirmed that the suspicious node is the malicious node in the network.
4.3 Choice of the simulator

As mentioned in the section 3.2 of the earlier chapter, we have chosen that Omnet ++ is our simulator. The following sections here give us more details about the other choices that we have made for this research.

4.3.1 Mobility models

As the nodes are mobile in this research and it is necessary to discuss about the mobility here. Omnet ++ has a framework extension to its core modules, and is called as “INETMANET” which provides the support for mobility. There are different mobility models that are available and are discussed below:

4.3.1.1 Ns2 mobility:

This mobility model uses the native file of the Bonn motion mobility model [45, 46] and is imported and developed that to Ns2 motion mobility model by the authors at [47]. Every line in the text file describes the motion of one host. And a line may consist of one or more triplets of real numbers like:

(t1 x1 y1 t2 x2 y2 t3 x3 y3 .......)

It means that a node gets to (xk,yk) at time tk and in this there is no separate notation for wait, because of that the x and y coordinates gets repeated here[48].

4.3.1.2 Restricted constant speed mobility

This mobility models all the controls related to movements of a host. According to the authors at [49], the user can define a velocity for each host and also update an interval. If the velocity is not zero then it is obvious that the host is moving. This mobility module calculates a random target position for the host. Depending upon the update interval and velocity it calculates the number of steps to reach the
destination. As soon as the target is reached then the module calculates for a new
target position.

This was originally in Mobility Framework and later imported to INET as the
Mobility Framework has not being developed anymore.

4.3.1.3 Restricted linear mobility

This is another kind of linear mobility model with the parameters like speed,
angle and acceleration. It defines a sub area and it restricts the mobility of the node(s)
to this predefined sub area. Whenever a node hits the wall then it reflects off the wall
at the same angle when it hits the wall [50].

4.3.1.4 Chiang mobility

This mobility module is developed by the author at [51]. In this model each
node moves randomly with a preset average speed [52]. This framework was initially
developed by TKN and later imported into INET.

4.3.1.5 Gauss Markov mobility

In this model the nodes start with an initial speed and direction. And at fixed
intervals of time t, movement occurs by updating the speed and direction of each node
and it is based on random values from a Gaussian distribution. It can also be defined
that a model which uses one tuning parameter to vary the degree of randomness in the
mobility pattern of the nodes [52, 53].

4.3.1.6 Random Waypoint mobility model

In this the node moves in line segments. A random destination (which is
distributed uniformly in the playground) and a random speed are chosen. The user
can define speed as a variant from which a new value will be drawn for each line
segment.
When the node reaches the target position, it waits for the wait Time which can also be defined as a variant. In this the movement pattern with a speed ranging between (minSpeed, maxSpeed) then a pause for a given pause time, periodically. This mobility model is initially developed by the author at [54] and is modified by Varga,A. [48] after importing into INET.

This ends the discussion about the mobility models for the nodes in the mobile ad hoc network and the following sections of this chapter will give more details about the routing protocols for MANET.

4.3.2 Routing protocols

In order to introduce mobility to the nodes in the network there is a need to use a routing protocol for that purpose. As discussed in the earlier chapters of this thesis about the routing protocols in the section 1.1.2.1.1, further sections of chapter 1 and in the Figure 1-1, there are different routing protocols. The routing protocols which are used for wired networks cannot be used for mobile ad hoc networks because of the mobility of the networks [55]. Clustering protocols have more disadvantages rather than advantages for the mobile ad hoc networks [54, 56, 57].
Table 4.1: Comparison among table driven routing protocols[58]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>DSDV</th>
<th>CGSR</th>
<th>WRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time complexity (link addition/failure)</td>
<td>O(d)</td>
<td>O(d)</td>
<td>O(h)</td>
</tr>
<tr>
<td>Communication complexity (link addition/failure)</td>
<td>O (x = N)</td>
<td>O (x = N)</td>
<td>O (x = N)</td>
</tr>
<tr>
<td>Routing philosophy</td>
<td>Flat</td>
<td>Hierarchical</td>
<td>Flat</td>
</tr>
<tr>
<td>Loop free</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes, but not instantaneous</td>
</tr>
<tr>
<td>Multicast capability</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Number of required tables</td>
<td>Two</td>
<td>Two</td>
<td>Four</td>
</tr>
<tr>
<td>Frequency of update transmissions</td>
<td>Periodically and as needed</td>
<td>Periodically as needed</td>
<td>Periodically and as needed</td>
</tr>
<tr>
<td>Updates transmitted to</td>
<td>Neighbors</td>
<td>Neighbors &amp; cluster head</td>
<td>Neighbors</td>
</tr>
<tr>
<td>Utilizes sequence numbers</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Utilizes “Hello” messages</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Critical nodes</td>
<td>No</td>
<td>Yes (cluster head)</td>
<td>No</td>
</tr>
<tr>
<td>Routing metric</td>
<td>Shortest path</td>
<td>Shortest path</td>
<td>Shortest path</td>
</tr>
</tbody>
</table>

N = Number of nodes in the network  
d = Network diameter  
h = Height of routing tree  
x = Number of nodes affected by a topological change

Table driven routing protocols are broadly classified and are given as follows:

1. Destination-Sequenced Distance Vector Routing (DSDV)  
2. Cluster Gateway Switch Routing (CGSR)  
3. Wireless Routing Protocol (WRP)

The differences between different table driven routing protocols are given in the above Table 4.1 and the differences between on-demand routing protocols are given in the Table 4.2.
Table 4.2 Comparison among on-demand routing protocols[58]

<table>
<thead>
<tr>
<th>Performance Parameters</th>
<th>AODV</th>
<th>DSR</th>
<th>TORA</th>
<th>ABR</th>
<th>SSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time complexity (initialization)</td>
<td>O(2d)</td>
<td>O(2d)</td>
<td>O(2d)</td>
<td>O(d+z)</td>
<td>O(d+z)</td>
</tr>
<tr>
<td>Time complexity (post failure)</td>
<td>O(2d)</td>
<td>O(2d) or</td>
<td>O(2d)</td>
<td>O(l+z)</td>
<td>O(l+z)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 (cache</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hit)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication complexity (initialization)</td>
<td>O(2N)</td>
<td>O(2N)</td>
<td>O(2N)</td>
<td>O(N+y)</td>
<td>O(N+y)</td>
</tr>
<tr>
<td>Communication complexity (post failure)</td>
<td>O(2N)</td>
<td>O(2N)</td>
<td>O(2x)</td>
<td>O(x+y)</td>
<td>O(x+y)</td>
</tr>
<tr>
<td>Routing philosophy</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
</tr>
<tr>
<td>Loop free</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multicast capability</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Beaconing requirements</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple route possibilities</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Routes maintained in</td>
<td>Route</td>
<td>Route</td>
<td>Route</td>
<td>Route</td>
<td>Route</td>
</tr>
<tr>
<td>Utilizes route</td>
<td>table</td>
<td>cache</td>
<td>table</td>
<td>table</td>
<td>table</td>
</tr>
<tr>
<td>expiration timers</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Route reconfiguration methodology</td>
<td>Erase</td>
<td>Erase</td>
<td>Link</td>
<td>Localized</td>
<td>Erase</td>
</tr>
<tr>
<td></td>
<td>Route;</td>
<td>Route;</td>
<td>Reversal;</td>
<td>broadcast</td>
<td>route;</td>
</tr>
<tr>
<td></td>
<td>Notify</td>
<td>Notify</td>
<td>Route</td>
<td>query</td>
<td>Notify</td>
</tr>
<tr>
<td></td>
<td>source</td>
<td>source</td>
<td>repair</td>
<td></td>
<td>source</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routing metric</td>
<td>Freshest</td>
<td>Shortest</td>
<td>Shortest</td>
<td>Associativity</td>
<td>Associativity</td>
</tr>
<tr>
<td></td>
<td>&amp; Shortest</td>
<td>Path</td>
<td>path</td>
<td>&amp; Shortest</td>
<td>&amp; Stability</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>path &amp;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>others</td>
<td></td>
</tr>
</tbody>
</table>

\( l = \text{Diameter of the affected network segment} \)

\( y = \text{Total number of nodes forming the directed path where the REPLY packet transits} \)

\( z = \text{Diameter of the directed path where the REPLY packet transits} \)
On-demand routing protocols are broadly classified as follows:

1. Ad hoc on-demand distance vector (AODV)
2. Dynamic Source Routing (DSR)
3. Temporally Ordered Routing Algorithm (TORA)
4. Associativity Based Routing (ABR)
5. Signal Stability Routing (SSR)

Finally another comparison among table driven and on-demand routing protocols is given as below:

Table 4.3 Comparison among On-demand and Table driven[58]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>On-demand</th>
<th>Table driven</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability of routing information</td>
<td>Available when needed</td>
<td>Always available regardless of need.</td>
</tr>
<tr>
<td>Periodic route updates</td>
<td>Not required</td>
<td>Required</td>
</tr>
<tr>
<td>Coping with mobility</td>
<td>Use localized route (ABR and SSR)</td>
<td>Inform other nodes to achieve a consistent routing table</td>
</tr>
<tr>
<td>Signaling traffic</td>
<td>Grows with increasing mobility of active routes</td>
<td>Greater than that of on-demand routing</td>
</tr>
</tbody>
</table>

Mainly we discuss about two ad hoc routing protocols Destination-Sequenced Distance-Vector Routing (DSDV) and Ad-hoc On Demand Distance Vector Routing (AODV) protocol. DSDV is a proactive routing protocol that depends on routing tables which are maintained at each node, whereas AODV is a reactive protocol, which finds a route destination only on demand when the communication is needed [8, 53-55]. And the comparisons for these two routing protocols are given in the Table 4.4.
Table 4.4 Comparison among DSDV and AODV [59]

<table>
<thead>
<tr>
<th>Name of the property</th>
<th>DSDV</th>
<th>AODV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast routes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Distributed</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Unidirectional link support</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Multicast</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Periodic broadcast</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>QoS Support</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Routes maintained in Reactive</td>
<td>Route table</td>
<td>Route table</td>
</tr>
</tbody>
</table>

In detail we will be discussing about two routing protocols and those are DSDV and AODV routing protocols.

4.3.2.1 Destination Sequenced Distance Vector (DSDV) protocol

This is a proactive routing protocol which is based on Bellman-Ford routing algorithm [60, 61]. This routing protocol was first developed by the authors at [13]. A routing table is maintained at each node in the network and with the help of this routing table, transmission of packets is done from one node to another node in the network [62, 63]. The improvement that has been made to Bellman-Ford routing algorithm is that in this, sequence numbers are used in this instead of loops in routing tables [59] and this sequence number is originated by the destination node.

The consistency is maintained by and when each node transmits and updates its routing table periodically. If the packets are broadcasted between nodes then it indicates that those nodes are accessible and how many hops are required to reach that particular node. These packets may be transmitted and containing the layer 2 or layer 3 address [64]. All the nodes advertise its own route tables to all of its neighbors in the network and this is the requirement of the DSDV routing protocol. As the entries of the routing tables change frequently, the advertisement should also be frequently
updated so that all the nodes in the network have the information about all of their neighbors in the network. The purpose of doing this is to make sure that the paths to reach a destination will have the number of hops for the routes, so that through this way even there is no direct connection from one node to another they can still communicate and exchange data.

As we have said earlier that each node transmits data and that data consists of new sequence number and the following information for each new route:

1. Destination address
2. Number of hops required to reach the destination and
3. The new sequence number originally stamped by the destination

The routing tables that are transmitted contain the information about the hardware address, network address of the host. The latest sequence number is preferred as the basis for making the forwarding decision of the data in the network between the hosts. In the cases like if the sequence numbers are the same then the one with better metrics is preferred. Even the sequence numbers are updated to all hosts in the network so that the nodes will decide on maintaining the routing entry for that originating mobile node. As soon as the route information is received, the receiving node increments the metric and transmits the information by broadcasting and this incrementing process is done only after the transmission because, even the incoming packet has to travel one more hop to reach its destination.

The mobile node(s) can cause broken links as they are mobile and these will be detected at the layer 2 protocol, which can be described as infinity. Whenever there is a broken route in the network, then that metric is assigned an infinity metric there by confirming that there is no hop and the sequence number are also updated. The
sequence numbers are defined to be even numbers and the infinity metrics are defined as odd numbers.

In DSDV protocol, the broadcasting of information can be done in two types namely: full dump and incremental dump [65, 66]. Full dump broadcasting carries all the routing information while incremental dump carry the information which has changed since the last full dump. Broadcasting is done in Network protocol data units (NPDU). A full dump requires multiple NPDU’s while the incremental dump requires only one NPDU to fit in all the required information [62]. Whenever a node enters into the network, then it will announce itself and the other nodes in the network update their routing information about that node as a new entry in their routing tables. Each mobile node advertises about reachability, information about layer 3 protocols at that destination [13].

**4.3.2.1.1 Example for DSDV operation**

In order to explain DSDV in detail, an example [13, 67] has considered below to explain that and is as below in the following sections.

![Diagram of Mobile host movement in ad hoc networks](image)

Figure 4-1 Mobile host movement in ad hoc networks[13]
The above Figure 4.1 has 8 hosts in the network. Through this example, we will be looking at the changes to the MH4 forwarding table in the Table 4.5 with the movements of MH1 reference. First all the nodes in the network advertise about their routing information to all the nodes in the network and so the routing table at MH4 looks as follows:

Table 4.5 Forwarding table of MH4 [13]

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence number</th>
<th>Install</th>
<th>Stable data</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH1</td>
<td>MH2</td>
<td>2</td>
<td>S406_MH1</td>
<td>T001_MH4</td>
<td>Ptr1_MH1</td>
</tr>
<tr>
<td>MH2</td>
<td>MH2</td>
<td>1</td>
<td>S128_MH2</td>
<td>T001_MH4</td>
<td>Ptr1_MH2</td>
</tr>
<tr>
<td>MH3</td>
<td>MH2</td>
<td>2</td>
<td>S564_MH3</td>
<td>T001_MH4</td>
<td>Ptr1_MH3</td>
</tr>
<tr>
<td>MH4</td>
<td>MH4</td>
<td>0</td>
<td>S710_MH4</td>
<td>T001_MH4</td>
<td>Ptr1_MH4</td>
</tr>
<tr>
<td>MH5</td>
<td>MH6</td>
<td>2</td>
<td>S392_MH5</td>
<td>T002_MH4</td>
<td>Ptr1_MH5</td>
</tr>
<tr>
<td>MH6</td>
<td>MH6</td>
<td>1</td>
<td>S076_MH6</td>
<td>T001_MH4</td>
<td>Ptr1_MH6</td>
</tr>
<tr>
<td>MH7</td>
<td>MH6</td>
<td>2</td>
<td>S128_MH7</td>
<td>T002_MH4</td>
<td>Ptr1_MH7</td>
</tr>
<tr>
<td>MH8</td>
<td>MH6</td>
<td>3</td>
<td>S050_MH8</td>
<td>T002_MH4</td>
<td>Ptr1_MH8</td>
</tr>
</tbody>
</table>

Table 4.6 Advertised route table of MH4 [13]

<table>
<thead>
<tr>
<th>Destination</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH1</td>
<td>2</td>
<td>S406_MH1</td>
</tr>
<tr>
<td>MH1</td>
<td>1</td>
<td>S128_MH2</td>
</tr>
<tr>
<td>MH1</td>
<td>2</td>
<td>S564_MH3</td>
</tr>
<tr>
<td>MH1</td>
<td>0</td>
<td>S710_MH4</td>
</tr>
<tr>
<td>MH1</td>
<td>2</td>
<td>S392_MH5</td>
</tr>
<tr>
<td>MH1</td>
<td>1</td>
<td>S076_MH6</td>
</tr>
<tr>
<td>MH1</td>
<td>2</td>
<td>S128_MH7</td>
</tr>
<tr>
<td>MH1</td>
<td>3</td>
<td>S050_MH8</td>
</tr>
</tbody>
</table>

The above Table 4.6 is about the advertised route table at MH4. As shown in the Figure 4-1 MH1 was moving its location nearer to MH7 and MH8, then the link between MH2 and MH1 is broken which results in the assignment of infinity as a metric at MH2 for MH1 and along with this the sequence numbers related to that will be changed to odd numbers in the routing table at MH2. Eventually MH2 will update this information to the other hosts in the network. And as there are no neighbors for
MH7 and MH8 they update their own routing tables and broadcast the message. At the end, MH4 will receive the updated information from MH6 and MH6 will receive two information packets from different neighbors to reach MH1 with same sequence number, but different metric. The route selection will depend upon the less hop count when the sequence number matches. And the updated forwarding table is as shown in Table 4-7 which is below:

Table 4.7 Forwarding table of MH4 (updated) [13]

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Metric</th>
<th>Sequence number</th>
<th>Install</th>
<th>Stable_data</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH1</td>
<td>MH6</td>
<td>3</td>
<td>S516-MH1</td>
<td>T810-MH4</td>
<td>Ptr1-MH1</td>
</tr>
<tr>
<td>MH2</td>
<td>MH2</td>
<td>1</td>
<td>S238-MH2</td>
<td>T001-MH4</td>
<td>Ptr1-MH2</td>
</tr>
<tr>
<td>MH3</td>
<td>MH2</td>
<td>2</td>
<td>S674-MH3</td>
<td>T001-MH4</td>
<td>Ptr1-MH3</td>
</tr>
<tr>
<td>MH4</td>
<td>MH4</td>
<td>0</td>
<td>S820-MH4</td>
<td>T001-MH4</td>
<td>Ptr1-MH4</td>
</tr>
<tr>
<td>MH5</td>
<td>MH6</td>
<td>2</td>
<td>S502-MH5</td>
<td>T002-MH4</td>
<td>Ptr1-MH5</td>
</tr>
<tr>
<td>MH6</td>
<td>MH6</td>
<td>1</td>
<td>S186-MH6</td>
<td>T001-MH4</td>
<td>Ptr1-MH6</td>
</tr>
<tr>
<td>MH7</td>
<td>MH6</td>
<td>2</td>
<td>S238-MH7</td>
<td>T002-MH4</td>
<td>Ptr1-MH7</td>
</tr>
<tr>
<td>MH8</td>
<td>MH6</td>
<td>3</td>
<td>S160-MH8</td>
<td>T002-MH4</td>
<td>Ptr1-MH8</td>
</tr>
</tbody>
</table>

After the above update, there will be another update in the routing table at MH4 too and is as shown below in the Table 4-8:

Table 4.8 Route table at MH4 (updated) [13]

<table>
<thead>
<tr>
<th>Destination</th>
<th>Metric</th>
<th>Sequence Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH1</td>
<td>3</td>
<td>S516_MH1</td>
</tr>
<tr>
<td>MH2</td>
<td>1</td>
<td>S238_MH2</td>
</tr>
<tr>
<td>MH3</td>
<td>2</td>
<td>S674_MH3</td>
</tr>
<tr>
<td>MH4</td>
<td>0</td>
<td>S820_MH4</td>
</tr>
<tr>
<td>MH5</td>
<td>2</td>
<td>S502_MH5</td>
</tr>
<tr>
<td>MH6</td>
<td>1</td>
<td>S186_MH6</td>
</tr>
<tr>
<td>MH7</td>
<td>2</td>
<td>S238_MH7</td>
</tr>
<tr>
<td>MH8</td>
<td>3</td>
<td>S160_MH8</td>
</tr>
</tbody>
</table>

4.3.2.2 Ad-hoc On Demand Distance Vector Protocol (AODV)

The AODV routing protocol is designed for adhoc mobile networks and it can handle unicast routing and as well as multicast routing [59, 68, 69]. This protocol has
the advantageous features of both DSR and DSDV algorithms and this protocol is an example of On-demand routing protocol which means the routes will be created only when there is a demand and also it maintains the routes only as long as they are needed. Creating and maintaining the routes in the network only when they are needed/demand makes this AODV protocol very useful and also a good algorithm for mobile ad hoc networks (MANET) [70].

4.3.2.2.1 Working of AODV

All the nodes in the network have routing tables of their own and they also maintain sequence numbers in order to avoid looping problems[70]. If a source node wants to send some data to a destination node and if it doesn’t have a route to the destination at that time then the source node broadcasts a route request (RREQ) packet throughout the network [59, 71]. The structure of the RREQ contains as below [62, 72, 73]:

1. Source address
2. Source Sequence number
3. Destination address
4. Destination sequence number
5. Hop count

The nodes will reply with a RREP if either the destination node or the intermediate node which is on the way to find the destination node and the structure of RREP format is as follows [74]:

1. Destination address
2. Destination sequence number
3. Source address
4. Lifetime
5. Signal Stability Routing (SSR)

In detail a node which receives the PREQ will send a reply (RREP) only if it is either the destination or if it is a path/route to the destination with a corresponding sequence number and only when that number is greater than or equal to the number which contains the RREQ [59]. In cases like this the nodes will unicasts a RREP to the source, otherwise; the nodes will rebroadcast the RREQ. The nodes will discard the RREQ and do not forward them if they have been processed those already. And the RREP will set up forward pointers to the destination by propagating back to the source nodes [59, 75, 76] When the source node receives the RREP, it records the latest sequence number to the requested destination and this process is called as Forward Path setup [62].

The intermediate nodes that receives another RREP after they had propagated the previous RREP towards the source, it then checks and compares the new destination sequence number of the new RREP with the previous RREP. These intermediate nodes updates their routing information and propagates a new RREP only when,

1. The destination sequence number is greater or
2. The new sequence number is same but the hop count is small or

It will just skip the new RREP. This process ensures that this algorithm is not making any loops and only the most effective is chosen [70]. If the data packets keep travelling from one node to another node along a certain path only then the route remains active otherwise the links will timeout and then be deleted from the routing tables of the intermediate nodes. In situations like where the links break while the route is being active then the node upstream of the link break generates a route error (RERR) to the source node to inform that it is not reachable to the destination(s).
After the source node receives this (RERR) message, then even if the source node still needs the route then it will reinitiate the route discovery to that destination [69, 77].

4.3.2.2.2 Management of the routing table

For each and every destination of interest, a route table entry is maintained for all the mobile nodes in the network. And the route table entry contains the following information [72]:

1. Destination
2. Next hop
3. Number of hops
4. Sequence number for the destination
5. Active neighbors for this route
6. Expiration time for the route table entry.

In addition to the source and destination sequence numbers the other useful information which is in the route entries is called as soft-state information associated to the route entry. For any active route, the information about the active neighbors is maintained so that whenever a link along a path to the destination breaks, then all the active source nodes will be notified [62, 72]

4.3.2.2.3 Example of AODV operation

An explanation with step by step details is given for the following Figure 4-2 which explains the operation/working of AODV routing. In this example [62], the source is denoted as “S” and the destination is denoted as “D”.

1. Source “S” has to send the data to its destination “D”.
2. Initially “S” sends the message “RREQ” to its neighbors A, B and C.

53
3. Among all the neighbors that “S” has sent the “RREQ” only “B” finds the path in its routing table with the entries, destination sequence number as S1 and hop count as C1, so “B” sends a reply message “RREP” to “S”.

4. “C” sets up reverse path.

5. Now, “C” forward RREQ to its neighbors to “D” and “E”.

6. “E” sets up the reverse path.

7. Now, “E” forwards RREQ to its neighbors F and G.

8. Slowly, “E” deletes the reverse path after a time out period as it does not receive any RREPs from F and G.

9. “D” also finds the path (with the entries destination sequence number s2 and hop count C1) and then it sends the message RREP to “C”.

10. “C” receives RREP from “D” and sets up a forward path and also forward RREP to S.

11. “A” also sets the reverse path; forwards RREQ to its neighbors; receives RREP (with path of hop count c2 >c1); sets forward path; and forwards this RREP to “S”.

12. The source “S” also receives a path information from C (with the entries destination sequence number S1 and hop count C1), and another path information from B (with destination sequence number S1 and hop count C1), and another path information from A (with destination sequence number x which is less than S1 and S2 and hop count C2 which is less than C1).

13. But finally, “S” choses path information from “C” (which was originated from D), giving first priority to the path with greatest destination sequence number and then the second priority to the path with smallest hop count.
14. Even though path given by A is of smallest hop count, it is ignored because the destination sequence number is greater than the path from C.

```
Source (S)
Initiate route discover by sending RREQ to the nodes A, B and C.
Receive RREP from the nodes A, B and C
Choose path via C as it is with greatest destination sequence number (hop count is the second priority).
Set information in the route table.
Start communicating with the destination through the chosen path.
```

```
A
Check for destination route.
No route found.
Set reverse path.
Propagate RREQ.
Send RREP with hop count = C2 (<c1)
```

```
B
Check for destination route.
Route found with hop count = C1 and dest.
Seq. number = S1
Send RREP.
```

```
C
Check for destination route.
No route found.
Set reverse path.
Propagate RREQ to D and E.
Receive RREP from D.
Send RREP.
```

```
D
Check for destination route.
Route found with dest.
Seq. number = S2 (>S1)
Send RREP.
```

```
E
Check for destination route.
No route found.
Set reverse path.
Propagate RREQ to F and G.
Delete reverse path after time out.
```

```
F
Check for destination route.
No route found.
```

```
G
Check for destination route.
No route found.
```

Figure 4-2 Process of route finding in AODV Routing Protocol

55
4.4 **Research approach features**

This section answers the questions which are mentioned in the previous chapter in the section 2.4.

1. How to identify the malicious nodes in the network? The process of identifying the malicious nodes and confirming that a node is malicious node are explained in 4.2.1

2. Is our method completely distributed? Yes, all the nodes in the MANET are configured to be responsible for detecting the abnormal behavior.

3. Is it possible to change the properties of the IDS or are the properties of the IDS configurable? We are using simple text files in order to store the properties so our solution is configurable.

4. Is our solution time-continuous? Yes, our solution is based on the transmission, response behavior and every node always checks it neighbors status and update the information about the neighboring nodes and the information about the malicious nodes.

4.5 **Conclusion**

In this chapter, we have discussed about the research solution at a high level and the following chapter presents a detailed solution. And also this chapter was dealt with the mobility models, routing protocols that are available and the next chapter explains in detail about the chosen mobility model and the routing protocol for this research.
Chapter 5

Research Approach Details

5.1 Research solution details

This section discusses about the details of the proposed research approach and also discusses about the implementation of this approach.

5.1.1 Detection of malicious nodes

The detection of malicious nodes in the network consists of a two-step process and each step is discussed in detail in the following sections.

5.1.1.1 Recognizing malicious nodes

The process of recognizing the nodes which are displaying the malicious behavior is a continuous process which is followed by all the nodes in the network and this process has been clearly explained in the flowchart[5] which is as shown in the Figure 5-1.

5.1.1.2 Confirming the malicious node

Once the process of recognizing the nodes which are displaying malicious behavior is done in the above step and then that suspicion has to be finalized and confirming that the node(s) is malicious node in the network. The confirmation process is done by following the flowchart which is in Figure 5-2. For example, if node A has detected that node B’s DTQ value has fallen below the threshold, then that
node wants to confirm its suspicion; which invokes the voting process and that will start sending a broadcast request for confirmation.

All the nodes in the network have DTQ values stored in their tables. When any node receives those kinds of requests, then that node(s) check on the DTQ values in their tables for the node that they received the request. After checking the tables they reply to the node with a positive or a negative vote.

This process is done at each and every node and the nodes will inform their decision to node A. Depending upon the number of votes that Node A has received about node B from the neighbors in the network, node B will exist in the network if it receives sufficient number of positive votes and it will be blacklisted and removed from the network if it receives all negative votes. If the node has received enough number of positive votes then the node stays in the network and also all the nodes will update their DTQ values in their tables for node B and also updates the threshold value.
Figure 5-1 Process for identifying malicious nodes[5]

Simulation Starts Node B has instantiated

Node B starts transmission to Node A

Node B waits for acknowledgement

Update # of packets transmitted (D)

Update historical throughput (H) for node A based on ack for N packets

Calculate recent (R) throughput for a fraction M of N packets

Calculate STB = R/H exp (a), Where a = step power function

\[ DTQ \approx \frac{D \cdot STB(\cdot)}{P \cdot E} \]

DTQ > “Threshold”

Start voting process

Node B back to wait for receive/ready to transmit state
Node A starts voting process for Node B

Node A broadcasts voting requests for node B

Has vote-request timeout been reached?

Condition to be satisfied for the Good node ($G_n$)

Did at least 60% of the nodes voted?

Yes

B stays in the network

All nodes update DTQ value for B

Update “Threshold” value

No

Node(s) will be blacklisted and the information is notified and updated

% of good votes < $G_n$ and $\geq$ Thsld1

% of good votes < Thsld1 and $\geq$ Thsld2

% of good votes $\geq$ Thsld 2

Voting process for two times, after that node(s) are moved to standby list and the tables are updated appropriately.

Voting process for one time and then node(s) are moved to standby list.

Node(s) will be blacklisted and the information is notified and updated.

WAIT

Has vote-request timeout been reached?

Did at least 60% of the nodes voted?

Yes

Condition to be satisfied for the Good node ($G_n$)

Yes

No

% of good votes $\geq$ Thsld 1 and $\geq$ Thsld2

Figure 5-2 Voting process
5.1.1.3 Configurable values

In the voting process of our IDS, there are some configurable values that the user can change them according to their requirement. The configurable values that we have used for this research are as follows:

1. Percentage value for Good node is 90%
2. Threshold 1 is 60%
3. Threshold 2 is 30%

The above mentioned values have been used for the experimentation setup in this research and these values can be configurable by the user at any time according to their requirement. The value of Threshold 1 is always bigger than the value of Threshold 2. Finally, the relation between the above configurable values is “Percentage value for Good node (Gn) > Threshold 1 > Threshold 2”.

5.1.2 Details of voting process

The voting process is as shown in the Figure 5.2 which is the flowchart representation for that implementation of the process. Further details of the voting process are discussed in detail as follows:

5.1.2.1 Arrival of vote

The node which initiates the voting process keeps track of the votes that it receives. For any particular vote-request, the vote initiating node doesn’t count more than one vote from the same neighbor (example: In the situations if a node wants to vote for two decisions). As soon as it receives the node it will decide whether it is for or against the node. In order to implement this we are considering all the nodes other than the vote-initiating node i.e., “N-1” nodes where “N” is the number of nodes present in the network at that time.
5.1.2.2 Voting nodes

All the nodes (= nodes that lie in the transmission range) that receive the voting request will attempt to vote.

5.1.2.3 Timeout for vote request

As soon as the vote-initiator sends out the vote request, it will also start a timer to receive the vote responses from the participating nodes. Only in ideal cases, all the existing nodes respond and vote their decision. But in reality, this might not happen because there might be some loss of packets and also some nodes may not respond for voting. For these kinds of situations, the vote-initiator cannot wait for longer periods of time. All the votes that are received after the timeout are useless. And we also have a condition for the number of votes that we receive; we should at least receive 60% of the votes from the nodes that are present in the network. If not it will reinitiate the voting process for 3 times more.

5.1.2.4 Percentage of votes received

After receiving the votes from the above process then there is a condition to check whether we have received at least 90% of majority of votes either in favor of or against a node. The condition here is calculated by a formula and is referred to as “Good node (Gn)”. The purpose of this condition is to avoid situations like 51 percent positive votes and 49 percent negative votes. In those kinds of situations, taking a decision means compromising with the security of the network to a little extent. But by passing through this condition those situations can be avoided.

5.1.2.5 Process after receiving the votes

Once the number of votes satisfies the condition, which means positive votes are 90% of the total number of votes received. Then the node is considered to be as
the good node and otherwise it enters to the other scenarios and the detailed information can be found in the Figure 5-2. The formula for checking whether the node receives 90% of the positive votes is given by the formula below:

The positive votes are represented by $X_1$, $X_2$, ..., $X_m$, the sum of positive votes are given by $\sum_{i=1}^{m} X_i$ and the negative votes are represented by $Y_1$, $Y_2$, ..., $Y_n$ and the sum of the negative votes are given by $\sum_{j=1}^{n} Y_j$. In order to confirm that a node is a good node, it should at least receive a minimum number of 90% of positive votes and the equation to satisfy that condition is as follows:

Condition to be satisfied for a good node:

$G_n = \left( \frac{\sum_{i=1}^{m} X_i}{\sum_{i=1}^{m} X_i + \sum_{j=1}^{n} Y_j} \right) \times P() \times 100$

…………………. (5-1)

Where $X_1$, $X_2$, ..., $X_m$ are positive votes

$Y_1$, $Y_2$, ..., $Y_n$ are negative votes

$P()$ = Probability of successful transmission

The value of the “$G_n$” will always range only from 0.9 to 1.0. This is because the condition is to test whether the node(s) are receiving 90% of the majority from the network. The whole product is multiplied by “100” inorder to get the value as a percentage.

5.1.2.6 Process for confirming the suspicious node as malicious

If a node was not able to satisfy the above condition then it falls into the category of suspicious node(s). As soon as the node doesn’t satisfy the condition then
depending upon the number of positive votes it will fall into any of the 3 sub-categories. The three sub categories and their process are explained in detail as below:

1. Level 1: If the number of positive votes that were received doesn’t satisfy the above condition and they are less than the value of threshold 1, then the voting process will be initiated two more times and then the node(s) will be moved to a standby list.

2. Level 2: When the node(s) doesn’t fit to the first level where the number of positive votes value lies between the threshold 1 and threshold 2, then they will fall into this category and here the voting process will be initiated one more time and then the node(s) will be moved to the standby list.

3. Level 3: This is the last category when the nodes doesn’t satisfy the above two conditions i.e., when the number of positive votes is the lowest which is less than threshold 2 and these nodes will be blacklisted directly.

5.1.2.7 Standby list

The node(s) which doesn’t satisfy the condition of “Good node” and eventually which fails the level 1 and level 2 conditions will fall into the standby list and when they are in the standby list then they will be penalized by the following conditions:

1. Limited availability for transmission.

2. Limiting the communication with one-hop neighbors.

If the node(s) behavior is bad even when they are in the standby list, then those node(s) will be blacklisted and further communication with those node(s) will be completely stopped and the information about this will be broadcasted for the entire network. At the same time, when the node(s) does comply with the restrictions
then the node(s) will be removed from the standby list and also it will be informed to the whole network.

5.1.2.8 Process after vote decision

1. Blacklisted: Basing upon the decision after the votes, if a node is blacklisted, then it means that a message is sent out to all the nodes with this information that a particular node has been blacklisted. As soon as all the nodes receive the information that the node is blacklisted then there will be no communication from such nodes in the network.

2. Acquitted: After the vote decision if a node is acquitted then nothing is done to the node and it just stays in the network and all the nodes update their DTQ values for this node and also the Threshold value is also updated.

5.1.2.9 Collection of data

For every neighboring node that a node tries to transmit data, DTQ is collected by those nodes and the collected data is in buckets corresponds to a certain set of messages that have been sent to those neighboring nodes. There are two bucket sizes namely the long-term-bucket and the near-term bucket [5]. The long term bucket is the first and bigger in size and this is collected when the number of sent messages collected is N and then the near-term bucket is the second and the smaller one when compared to N and is called as “M”. The values N and M are defined such that N is divisible by M. And also we select N and M such that (N mod M =0). Here N is divided into M buckets, where they contain the statistics for the count of (mod(N+M)) messages attempted to be sent to a neighbor node. As we can see that the statistics that are collected for M statistical buckets of (N+M) messages represent the recent behavior of the node whereas the historical data is given by the statistics collect for N sent messages. We do also collect the number of messages acknowledged per (N mod
M) messages attempted to be sent to a neighbor node N, and store M buckets of such data.

According to the authors, Li, T., Alam, M., et al. in [33, 38] after collecting the data then DTQ can be calculated by using the following formula:

\[
DTQ = k \times \frac{D \times STB()}{ExP()} \quad \text{.................. (5-2)}
\]

The values of the function “DTQ” are given as follows:

\[ D = \text{Total packets transmitted successfully,} \]
\[ E = \text{Total energy spent to transmit data burst,} \]
\[ P() = \text{Probability of successful TX (property function of the environment)} \]
\[ STB() = \text{Stability factor} \]
\[ k = \text{constant} \]

The stability factor, STB () is defined as:

\[
\sum_{i=0}^{N} \left( \frac{d_i}{u_i} \right)^{\alpha} \sum_{j=0}^{M} \left( \frac{d_j}{u_j} \right) \quad \text{......................... (5-3)}
\]

where \( d_i \) and \( u_i \) represent the bytes successfully transmitted and the bytes attempted to be transmitted respectively, when sending the past ith data messages. \( \alpha \) is a constant greater than 1.

In our case,

\[
\sum_{i=0}^{N} \left( \frac{d_i}{u_i} \right) = \left[ \begin{array}{cccc}
\delta & \delta & \delta & \cdots \\
\frac{1}{k} & \frac{1}{k} & \frac{1}{k} & \cdots \\
\vdots & \vdots & \vdots & \ddots \\
\frac{1}{k} & \frac{1}{k} & \frac{1}{k} & \cdots \\
\end{array} \right] \begin{array}{c}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{array}
\frac{N \times \text{times}}{M} \begin{array}{c}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{array} \\
\sum_{i=0}^{N} \left( \frac{d_i}{u_i} \right) = \left[ \begin{array}{cccc}
\delta & \delta & \delta & \cdots \\
\frac{1}{k} & \frac{1}{k} & \frac{1}{k} & \cdots \\
\vdots & \vdots & \vdots & \ddots \\
\frac{1}{k} & \frac{1}{k} & \frac{1}{k} & \cdots \\
\end{array} \right] \begin{array}{c}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{array} \begin{array}{c}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{array}
\frac{N \times \text{times}}{M} \begin{array}{c}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{array} \quad \text{........ (5-4)}
\]

66
Where the numerator is a series of sum of \(\frac{0}{k} + \frac{1}{k} \cdot \frac{1}{k} + \cdots\) and so on. This is because, in our case, we either transmit a full packet, or transmit none, and packet size is a constant \(k\). Thus \(\frac{d_i}{u_i} = \frac{1}{k}\) if transmission was a success (i.e., ACK is received). \(\frac{d_i}{u_i} = \frac{0}{k}\) if transmission was a failure.

Thus, we have \(\text{STB}()\) as

\[
\text{STB}() = (\frac{N}{M} \cdot \frac{1}{k}) \quad \ldots \ldots (5-5)
\]

Further simplifying we have this as

\[
\text{STB}() = (\frac{N}{M}) \quad \ldots \ldots (5-6)
\]

Thus, \(\text{DTQ} = k \cdot D \cdot E \cdot P() \cdot (\frac{N}{M} \cdot \frac{1}{k}) \quad \ldots \ldots (5-7)
\]

Finally, \(\text{DTQ}\) will be calculated by using the formulas which are shown in the above equations.

### 5.1.3 Attacks

Mobile ad hoc networks are vulnerable to security attacks because of their properties like lack of a central entity and also the nodes are mobile and dynamic. The attacks will be more in mobile ad hoc networks when compared to wired networks.
because for wired networks the attackers have to gain access to the network physically whereas this is not the case with the mobile ad hoc networks.

5.1.3.1 Types of Attacks

There are different types of attacks in MANET. The attacks that are often encountered in MANET are flooding attack, selective forwarding attack, sinkhole attack, black-hole attack and wormhole attack. Models of some of these attacks are used to test the efficiency of our research idea. The following sub sections will explain the above mentioned attacks in detail.

5.1.3.1.1 Flooding attack

In this attack, the attacker exhausts the network resources such as bandwidth and to consume node’s resources such as computational and battery power or to interrupt the routing operation which causes severe degradation in network performance [78].

5.1.3.1.2 Selective forwarding attack

This attack occurs when a compromised node drops a packet that is meant for a particular destination. So, in this way an attacker can selectively filter traffic from a particular part of the network. Some other possible variations of selective forwarding are involved dropping a percentage or a random number of packets [5].

5.1.3.1.3 Black hole attack

In this attack, the compromised node will not be able to forward any data at all which means it fails to forward any kind of message that arrives at the node [29].
5.1.3.1.4 Sinkhole attack

In this the attacking node tries to offer a very attractive link e.g., to a gateway so this way it attracts more data traffic by advertising themselves as the best path to other destinations in the network [79, 80].

5.1.3.1.5 Sybil attack

A single node will be represented as multiple identities to other nodes in the network is called as Sybil attack [81]. These attacks pose a significant threat to the geographic routing protocols, because location aware routing requires the nodes to exchange coordinate information with their neighbors so that they can effectively route the geographically addressed packets [79]

5.1.3.1.6 Wormhole attack

This attack is particularly severe attack on MANET routing where two attackers, will be connected by a high-speed off-channel link and are strategically placed at different ends of the network [82]. Because of the influence of this wormhole attack the nodes which are distant in the network appear to be very nearer.

5.2 Details of implementation

This section gives us more detailed information about the implementation of the research idea. This gives us the explanation about the modules, packages used for implementing this research.

5.2.1 Routing protocol

Two routing protocols namely DSDV and AODV were discussed in detail in the previous section 4.3.2. We have used the AODV routing protocol which was initially developed by Nicola Concer which was imported by Andras Varga into
Omnet ++ 3.1. But now the one that we are using here for this research was been improved and updated for Omnet ++ 4.1 by Alfonso [83].

5.2.2 Mobility model

Different kinds of available mobility models are described in detail in the previous section 4.3.1. We are using the Restricted Linear mobility model for this research and this is previously described in the section 4.3.1.3.

5.2.3 Attack models

We are implementing selective forwarding, flooding attack and black hole attacks for this research. In order to implement these attacks we have created nodes that just do not acknowledge any kind of packet reception. The flooding attacks can be created by pumping out messages at a very fast rate in the network. Because of these actions the voting process will get invoked and then eventually the malicious nodes will be detected by the IDS.

5.3 Conclusion

In this chapter we have discussed in detail about the research solution and the details of the implementation of the routing protocol, mobility model and attack models used in this. And the following chapter records all data collected from the simulation runs and discusses the observations.
Chapter 6

Discussion of Results

This chapter deals with the discussions about the results that were captured from the simulation runs.

6.1 Detection of malicious nodes by IDS:

The following sections will give the information about the detection of malicious nodes by our IDS in the MANET.

6.1.1 Changes by Mobility of the nodes:

The detection of malicious nodes has been analyzed by varying the speed of the mobility of the nodes in the network.

Settings used for varying speed:

Number of nodes = 10
Simulation run time = 500s
Mobility update interval = 100ms
Malicious nodes = 4

6.1.1.1 Varying speed

By varying the speed at which the node travels in the network. The variation of detection of malicious nodes in the network with our IDS is obtained as follows:
6.1.1.2 Discussion

The plot which is shown in above Figure 6-1 shows the detection of malicious node vs. simulation time (in seconds) with varying the speed of the mobile nodes in the network.

1. All the malicious nodes are successfully detected.
2. If we observe the above graph then we can see that there are also false positives.
3. False positives can be analyzed by the output files omnetpp (.vec and .sca) and especially by observing and analyzing the sent/received message counts that are obtained and they may occur due to one of the following reasons:
   a. Vote replies may be lost.
   b. False positives decrease when the nodes have comparable movement.

When the node moves very fast then there might be connections that
are lost and loss of packets and eventually there will be some false positives.

- During the process of transit, time-out in receiving the replies, losing connectivity when the nodes are mobile there might be loss of packets. The routing protocol also be the reason for a definite loss factor and is documented in the AODV routing protocol implementation as in [84].

### 6.1.2 Changes by malicious node count

The detection of malicious nodes has been analyzed by varying the number of the malicious nodes in the network.

**Settings used for varying number of malicious nodes:**

- Number of nodes = 20
- Simulation run time = 1000s
- Mobility update interval = 100ms
- Area size = 1000x1000 flat area
- Transmission Range = 250m

**6.1.2.1 Varying the number of malicious nodes:**

By varying the number of the malicious nodes in the network, the variation of detection of malicious nodes in the network with our IDS is obtained as follows:
6.1.2.2 Discussion

The graphs which are shown in the above Figure 6-2 have started with plotting the values from 10% of malicious node count to 50% of malicious node. All the malicious nodes were successfully detected in these tested scenarios. No false positives were happened even though for some cases the simulation has ran for considerable amount of time (some times longer durations). This proves that the IDS has good detection rate and more detailed values about the detection rate were given in the Table 6.1.

6.1.3 Importance of voting process

The key role is played by the “Voting process” which is the crucial part of our IDS in order to find out the malicious nodes in the network.

Settings used:

Number of nodes = 20
Simulation run time = 1000s
Mobility update interval = 100ms
Area size = 1000x1000 flat area
Transmission Range = 250m

6.1.3.1 Turning the voting process on and off

In order to observe the importance of the voting process we have run the simulation by turning the voting process on/off with 30% of the malicious nodes present in the mobile ad hoc network. The results were plotted as shown in the following Figure 6-3.

![Effect of Voting process](image)

Figure 6-3 Effect of voting process

6.1.3.2 Discussion

The above plot which is in Figure 6-3 represents the importance of voting process. It is clearly observed that without the voting process the malicious nodes were not detected correctly which means there was an existence of false positive in the above shown figure. In this we have considered two scenarios for the IDS, so this way it reveals the importance of the presence of voting process. In the above scenario,
all the malicious nodes were detected completely at 392 seconds even though the simulation ran for 1000 seconds in a network which has 30% of the malicious nodes and with the voting process the malicious nodes were detected correctly and without the voting process there was a false positive that was observed. Finally, we can say that there are more number of false positives and false negatives if we do not use the voting process and which makes the IDS have a better performance.

6.1.4 Performance of IDS with different attacks

The performance of the IDS is measured with 3 different attacks that we had implemented in this research and the attacks are Selective forwarding attack, Flooding attack and Black hole attack. The detection success rate and the respective false alarm rates are also being presented in the following Table 6.1.

Settings used:

Number of nodes =20
Mobility update interval = 100ms
Area size = 1000x1000 flat area
Transmission Range = 250m
Speed = 0-20 mps
## Table 6.1 Detection performance of IDS with different attacks

<table>
<thead>
<tr>
<th>Running time (sec)</th>
<th>Selective forwarding attack</th>
<th>Flooding attack</th>
<th>Black hole attack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Detection rate</td>
<td>False positives</td>
<td>False negatives</td>
</tr>
<tr>
<td>100</td>
<td>98.31 ± 0.79 %</td>
<td>1.12 ± 0.14 %</td>
<td>0.60 ± 0.03 %</td>
</tr>
<tr>
<td>200</td>
<td>97.90 ± 0.72 %</td>
<td>1.08 ± 0.09 %</td>
<td>0.45 ± 0.02 %</td>
</tr>
<tr>
<td>300</td>
<td>96.72 ± 0.51 %</td>
<td>2.06 ± 0.04 %</td>
<td>0.31 ± 0.04 %</td>
</tr>
<tr>
<td>400</td>
<td>97.18 ± 0.32 %</td>
<td>1.82 ± 0.07 %</td>
<td>0.63 ± 0.05 %</td>
</tr>
<tr>
<td>500</td>
<td>96.25 ± 0.58 %</td>
<td>1.94 ± 0.21 %</td>
<td>0.52 ± 0.11 %</td>
</tr>
</tbody>
</table>
6.1.4.1 Discussion

The detection performance of the IDS that is developed in this research has been tested with 3 different kinds of attacks to evaluate its performance. For each result we had run the simulation for ten times with different speed values for the mobile nodes in the MANET and the average number of the values that obtained has been given in our results in Table 6.1. Finally, we can strongly say that the performance of the IDS stands good and a better window size can be choosen in order to minimize the false positives/negatives. On the whole, the detection rate of IDS has never gone below 95% on whole and the values of the false positives (worst case = 3.39 %) and the false negatives (worst case = 1.28 %) are comparibly very low. Finally on the whole the false alarm rate has never gone above 5% in most simulation cases.

6.2 Effect of different parameters while varying the speed

The overall performance of the network with the IDS is considered and the evaluation is done for comparing different values by varying speed and they are explained in detail as below with different scenarios.

6.2.1 Reception of packets

The following graph is a plot between number of packets received and the speed of the nodes in the network.

Settings used:

Number of hosts = 20
Area size = 1000x1000 flat area
Transmission Range = 250m
6.2.1.1 Discussion

The above Figure 6-4 and Figure 6-5 gives the information about the number of packets received while varying the speed of the mobile nodes. From the above graph it is clearly observed that the performance of the mobile device decreases when
its speed is increasing or in other words, the higher the speed of a mobile device the less better is its performance. When some nodes are moving at a relatively fast rate, the immobile nodes remain connected to these nodes for very brief periods of time, causing loss of packets. Thus, when the nodes have comparable movement profiles, nodes move to each other’s range more easily and hence they do not lose connectivity for extended periods. Out of all the available routing protocols, AODV is the better choice and packet delivery is also very good in AODV when compared to other routing protocols [85].

6.2.2 Performance of Network Delay

The following plot is drawn between the mean delay and the speed of the nodes in the network.

**Settings used:**

Number of hosts = 20

Area size = 1000x1000 flat area

Transmission Range = 250m

![Performance of Network Delay](image)

Figure 6-6 Performance of Network Delay
6.2.2.1 Discussion

The above Figure 6-6 and Figure 6-7 are the plots between mean delay of the network and the speed of the mobile nodes in the network; these plots are drawn by varying the simulation time of the network. The network delay performance stabilizes and becomes almost constant in a long-term scenario. Finally network performance will be better in long-term scenarios than very short-term scenarios. In reality the speeds of all the nodes will not be the same or constant, so that is the reason we have observed the performance of network delay by varying the speed of the mobile nodes in the network. In the same way, other parameters like acceleration, velocity can also be changed for the mobile nodes and those performances can also be observed.

6.3 Impact of IDS

The following plot is drawn between the throughput and the delay of the mobile nodes in the network with different scenarios. The different scenarios are without the IDS, with different attacks and the presence of IDS.
**Settings used:**

Number of hosts = 20

Number of malicious nodes = 4

Simulation run time = 1000s

Area size = 1000x1000 flat area

Transmission Range = 250m

---

Figure 6-8 Variation of Throughput with Delay

**6.3.1.1 Discussion**

The plot in the above Figure 6-8 gives us the information about the behavior of the throughput of the network with delay. The configuration settings used for this test are mentioned as above. The above plot has been drawn without IDS and with the presence of different attacks and finally compared the performance of the network throughput vs delay respectively. According to the ideal conditions this graph should be gradually decreasing and then almost becomes a constant. Finally, we can say that the above plot has better outcome with better consistency which means the performance of IDS is good.
6.4 Conclusion

In this chapter we have presented all the results that we have obtained by testing the IDS with different attacks, with and without the presence of IDS, voting process and discussed each of them in detail and also presented all of our observations. And in the following chapter we present about the conclusion of this research and also mention about the future continuation of this work.
Chapter 7

Conclusion

7.1 Conclusion

Our aim was to identify the malicious node(s) in a MANET where the nodes are mobile in the network; suspicion and the detection process of finding the malicious node in the mobile ad hoc network is based on the behavior of the node(s).

We have chosen to use Omnet ++ 4.1 as the simulator for creating the environment of the Mobile ad hoc network. As the nodes are mobile, so obviously they do need a routing protocol for the implementation of mobility. So, in order to perform this function we have used the AODV i.e., Adhoc On-demand distance vector routing protocol. The MAC and physical layers for this follows the standards of IEEE 802.11.

The attacks that we had used in this research in order to test the IDS are selective forwarding, flooding attack and black hole attacks. All the malicious nodes were successfully detected. Each point in the plots is an average value of 10 runs. All the data that has been collected has been put in the previous chapter and also we have discussed about the results in detail. Our IDS can detect malicious nodes with almost 95 % proficiency in the worst case scenario. And also the percentages of the false positives and false negatives are also reasonable and have never exceeded 5% for most simulation cases.
7.2 Scope for future work

This section of this chapter discusses about the few areas in which the current work can be extended further.

1. Testing the model with different types of mobility models.

2. Building more attack models like wormhole, Sybil, sinkhole etc. and also test the IDS using their implementation.

3. Invoking of the voting process can be changed with different types of attacks.

4. Introduce and develop fuzzy voting process.

5. Contribution to the open source community of Omnet ++.
References


5. Kumar, K., Intrusion Detection in Mobile Ad Hoc Networks, in Electrical Engineering and Computer Science 2009, University of Toledo: Toledo.


56. Valentin, O. *Clustering in mobile ad-hoc networks*. March 2011.


73. Lin, C., *AODV routing implementation for scalable wireless ad hoc network simulation (SWANS)*, Cornell University.


80. Saini, A. and H. Kumar, *Comparision between various black hole detection techniques in manet*, in *National Conference on Computational Instrumentation* 2010: Chandigarh, India.


84. Lu, Y., Y. Zhong, and B. Bhargava, *CISCO URP grant*, CERIAS, NSF grants: West Lafayette, IN, USA.