DETECTION OF MALICIOUS NODES IN ROUTING OF MOBILE ADHOC NETWORK

BY

MANIAR SWEETY M.

07MCE008

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

MAY 2009

Detection of malicious nodes in Routing of Mobile Adhoc Network

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Maniar Sweety M.

07MCE008

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May 2009

Declaration

This is to declare that the Major Project entitled "Detection of malicious nodes in Routing of Mobile Adhoc Network" submitted by me Maniar Sweety M. (07MCE008), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by me under the supervision and guidance of my guide. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

- Maniar Sweety M.

07MCE008

Certificate

This is to certify that the Major Project entitled "Detection of malicious nodes in Routing of Mobile Adhoc Network" submitted by Maniar Sweety M. (07MCE008), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. S.N. Pradhan
Guide, Professor,
Department Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad

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

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

Abstract

This project aims to find method of detecting selfish and misbehaving node for providing better security in routing of adhoc network.

First part of the project is to generate the adhoc network. In Adhoc network nodes are mobile so the network topology may change rapidly and unpredictably over time. The network is decentralized, where all network activity including discovering the topology and delivering routing messages is executed by the nodes themselves, so one or more of them may misbehave and disturb the network. The misbehavior or attack can be of many types.

In the network the node can work in two ways by exhibiting selfishness or misbehavior and cause disturb once in the network by using different type of attack. To identify or detecting malicious or selfish node Intrusion Detection System (IDS) system is developed.It has different architecture for to detect malicious or selfish node. One is Stand Alone architecture and other is Distributed and co-operative architecture.

IDS System has stand alone Architecture uses WatchDog mechanism to detect selfish and misbehaving node that agree to forward packet but fails to do so. Pathrater is mechanism used for removing path from cache that contain malicious or selfish node. By using the both these mechanisms with DSR protocol result improvement in performance of network.

Acknowledgements

It gives me immense pleasure in expressing my thanks and profound gratitude to Dr. S. N. Pradhan, Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continuous encouragement throughout my Major project. I am heartily thankful to him for his precious time, suggestions and sorting out the difficulties of my topic that helped me a lot during this study.

I would like to give my special thanks to Prof. D.J. Patel, Head, Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad for his encouragement and motivation throughout the Major Project, I am also thankful to Dr. Ketan Kotecha, Director, and Institute of Technology for his kind support in all respect during my study.

I am thankful to Prof. Zunzun Narmavala who constantly helped me with problems in implementing of ns2 code and all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

My family members and friends, who motivated and supported me all throughout the days of hard work, contribute a lot in making this work successful. I thank them from the bottom of my heart.

- Maniar Sweety M.

07MCE008

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 General

The type of wireless networks that is the infrastructure less mobile and has nodes is known as an Mobile ad hoc network(MANET). Infrastructure less mobile networks have no fixed routers and base stations and the participating nodes are capable of movement. Due to the limited transmission range, multiple hops may be required for nodes to communicate across the ad hoc network. Routing functionality is incorporated into each host, thus ad hoc networks can be characterized as having dynamic, multi-hop, and constantly changing topologies.

Figure 1.1 illustrates an example ad hoc network. The participating nodes act both as end hosts and routers forwarding traffic from the source to the destination host.

Due to the lack of stationary infrastructure, the participating nodes in the ad hoc network have to forward traffic on behalf of other nodes that are not in close proximity to the destination node. If they deny participating in the routing process, the connectivity between nodes may be lost and the network could be segmented.

Figure 1.1: Ad hoc network example

Therefore, the functionality of an ad hoc network heavily depends on the forwarding behavior of the participating nodes.

Another very important property of ad hoc networks is their dynamic topology(shown in figure 1.2). Since the topology arbitrarily changes due to node mobility and changes of the surrounding environment(shown in figure 1.3), the utilized routing protocols have to be able to adapt to the dynamic topology. The routing protocols that are currently utilized in ad hoc environments have specifically been designed to handle node mobility and rapidly changing topologies. The devices that are usually employed in the ad hoc networks have their own limitations. Since, the only hardware component that is required to connect a device in an ad hoc network is a wireless interface, PDAs and mobile telephones can be utilized. Furthermore, differences in the radio transmission ranges and reception equipment sensitivities may lead to unidirectional links which could complicate routing in the ad hoc networks. Apart from the communication differences between the nodes, ad hoc networks suffer from limited hardware resources like limited battery, constrained CPUs and small memory capacity.

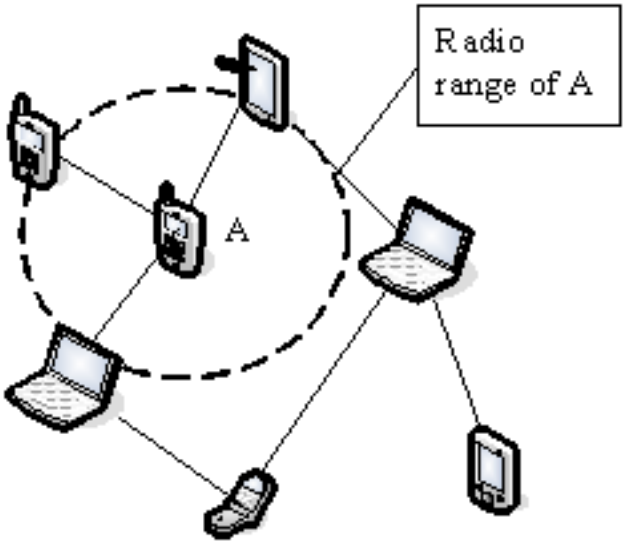

Figure 1.2: Before Movement

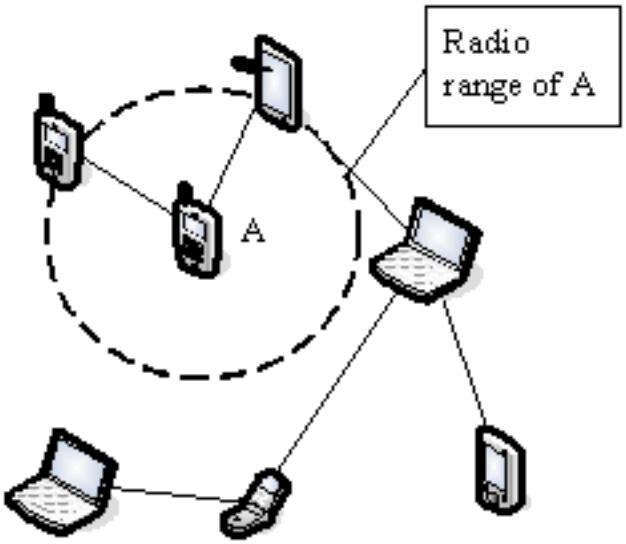

Figure 1.3: After Movement

1.2 Routing of adhoc network

In ad hoc networking environments an application packet from a specific node may have to travel several hops in order to reach its destination. The main function of a routing protocol is to form and maintain a routing table with information relevant to which the next hop for this packet should be in order to reach its ultimate destination. All the nodes have their own routing tables that they consult to forward the routing traffic that it is not destined for them.

Another issue that contributes to the fact that the available routing protocols cannot operate in ad hoc mode is that they were designed with the assumption that all the links are bidirectional. In mobile ad hoc networks this is not always the case. The differences of the wireless networking hardware of the nodes or the radio signal fluctuations may result in some links becoming unidirectional.

Routing protocols for Ad Hoc networking can be classified into four categories viz. based on the routing information update mechanism, the use of temporal information for routing, routing topology, and utilization of specific resources. The following subsections summarize the descriptions of each class.

1.2.1 Table driven routing protocol

Based on the periodically exchanging of routing information between the different nodes, each node builds its own routing table which it can use to find a path to a destination.

Table driven routing protocols attempt to maintain consistent, up to date routing information from each node to every other node in the network.

These protocols require each node to maintain one or more tables to store routing information and they respond to changes in network topology by propagating route updates through out the network.

The routing protocols differ in the method by which the topology change information is distributed and the number of necessary routing related tables.

Examples of the protocols of this class are, Destination Sequenced Distance Vector routing protocol (DSDV), Wireless Routing Protocol (WRP), Cluster-Head Gateway Switch Routing protocol and Source Tree Adaptive Routing protocol (STAR).

1.2.2 On demand routing protocol

The nodes do not exchange any routing information. A source node obtains a path to a specific destination only when it needs to send some data to it.

On demand routing protocols try to eliminate the conventional routing tables and consequently reduce the need for updating these tables to track changes in the network topology.

In contrast to table-driven routing protocols all up-to-date routes are not maintained at every node, instead the routes are created when required. On demand routing protocols calculate a path before data transmission.

When a source wants to send to a destination, it has to invoke the following procedures:

1) Route discovery

2) Route maintenance

3) Route deletion

Route discovery is not required for the transmission of every single data packet, since the discovered path is likely to be valid for a period of time that allows many transmissions to the same destination. Examples of the protocols of this class are, Dynamic Source Routing protocol (DSR), Ad Hoc On-Demand Distance-Vector Routing protocol (AODV), and Temporally Ordered Routing Protocol (TORA).

1.2.3 Hybrid routing protocol

Nodes are grouped into zones based on their geographical locations or distances form each other. Inside a single zone, routing is done based using table-driven mechanisms while an on-demand routing is applied for routing beyond the zone boundaries.

In a comparison between the table driven routing protocols and on-demand routing protocols is introduced. This comparison defines the main differences between the two classes of protocols. Mainly, the availability of routing information is a key advantage of table driven routing protocols, because faster routing decisions - and consequently less delay in route setup process- can be made than in the case of on-demand routing protocols. On the other hand, this important advantage of table driven routing protocols requires periodic routing updates keep the routing tables up to date, which in turn costs higher signaling traffic than the required for on-demand routing protocols.

However, for other functions like path reconfiguration after link failures, there are variations between the protocols of each class. For example, both DSR and TORA are on-demand routing protocols. At the same time, DSR uses global route maintenance schemes while TORA uses a local one which reduced signaling overhead.

From the above, it is important to understand that we can not come to absolute

conclusions about the preference of some class than the other, and such preference conclusions should be done at the protocols level, and not at the class level.

1.3 Motivation

The purpose of this project is to address the need of security in routing of wireless network.

In adhoc wireless network due to lack of infrastructure and dynamic topology mobile node can provide information from one network to other network or one mobile node to other node gets easily.

1.4 Scope of work

As the title suggests the goal of this project, work carried out in this research is useful in improving performance and reliability of the adhoc mobile wireless network.

The work of project is analysis, the current trends in eliminating misbehaving nodes from takeing part in routing and then do design and simulation of the Adhoc Wireless network that provide the security. Intrusion Detection System (IDS) is proposed that identifies to detect mailicous nodes of different categories.

In this project, development of the watchdog mechanism to detect malicious or selfish node. Pathrater is the mechanism that avoids misbehaving nodes from taking part in packet routing. When this mechanisms used with DSR routing then it is affects on overhead and throughput of network.

1.5 Thesis Organization

The Thesis on "Detection of malicious node in Routing of mobile adhoc network" has been divided in chapters as follows:

Chapter 2, *Literature Survey* , presents the problem presents the literature review. It provides overview of adhoc network security and explains the different types of attack.

Chapter 3, *Security scheme & IDS*, includes the security scheme related to adhoc network. It also provides the architecture of IDS system with different component use in IDS. The chapter includes types of mechanism use for the detection of malicious node.

In **chapter 4**, *Mobility Model & NS*, provides the different mobility model and how to generate that model in NS. It also explain the tool related to mobility model.

Chapter 5, *Simulation and Performance Evaluation*, covers simulation of adhoc wireless network with different parameter using ns2 simulator. The chapter includes the output of the simulation and generates the graph.

Finally, in **chapter 6** concludes this project with a summary, and provides possible directions for relevant future research.

Chapter 2

Literature Survey

2.1 General

During the last few years we have all witnessed a continuously increasing growth in the deployment of wireless and mobile communication networks. Mobile ad hoc networks consist of nodes that are able to communicate through the use of wireless mediums and form dynamic topologies. The basic characteristic of these networks is the complete lack of any kind of infrastructure, and therefore the absence of dedicated nodes that provide network management operations like the traditional routers in fixed networks. In order to maintain connectivity in a mobile ad hoc network all participating nodes have to perform routing of network traffic. The cooperation of nodes cannot be enforced by a centralized administration authority since one does not exist. Therefore, a network layer protocol designed for such self-organized networks must enforce connectivity and security requirements in order to guarantee the undisrupted operation of the higher layer protocols.

Security is an essential service for wired and wireless network communications. The success of mobile ad hoc networks (MANET) strongly depends on peoples confidence in its security. However, the characteristics of MANET pose both challenges

and opportunities in achieving security goals, such as confidentiality, authentication, integrity, availability, and non-repudiation. We provide a survey on attacks and countermeasures in MANET in this paper. The countermeasures are features or functions that reduce or eliminate security vulnerabilities and attacks. First, we give an overview of attacks according to the protocols stacks, and to security attributes and mechanisms.

2.2 Literature Review

S. Martiet [1]. provides the information of DSR Routing algorithm. The paper also describe two techniques that improve throughput in an adhoc network in the presence of nodes that agree to forward packet but fails to do so. To mitigating this problem we propose categorizing the nodes based upon their dynamically measured behavior. We use watchdog that identifies misbehavior node and pathrater that helps routing protocols avoid these misbehavior of nodes. Through simulation we evaluate watchdog and pathrater using packet throughput, percentage of overhead(routing) transmission and the accuracy of misbehaving node detection.

2.2.1 DSR(Dynamic Source Routing)

DSR is a on demand source routing protocol. Every packet has a route path consisting of the addresses of nodes that have agreed to participate in routing the packet. The protocol is referred to as "On Demand" because route paths are discovered at the time a source sends a packet to a destination for which the source has no path.

DSR has two main functions: Route discovery and route maintenance. Figure 2.1 illustrate route discovery. Node S(the source) wishes to communicate with node D(the destination) but does not know any paths to D. S initiates a route discovery by broadcasting a ROUTE Request packet to its neighbors that contains the address D. The neighbors in turn append their own addresses to the ROUTE Request packet. D

must now send back a route reply packet to inform S of the discovered route. Since the Route Request packet that reaches D contain a path from S to D, D may choose to use the reverse path to send back reply or to initiate a new request discovery back to S. Since there can be many routes from a source to a destination, a source may receive multiple route replies from destination. DSR caches these routes in a route cache for future use.

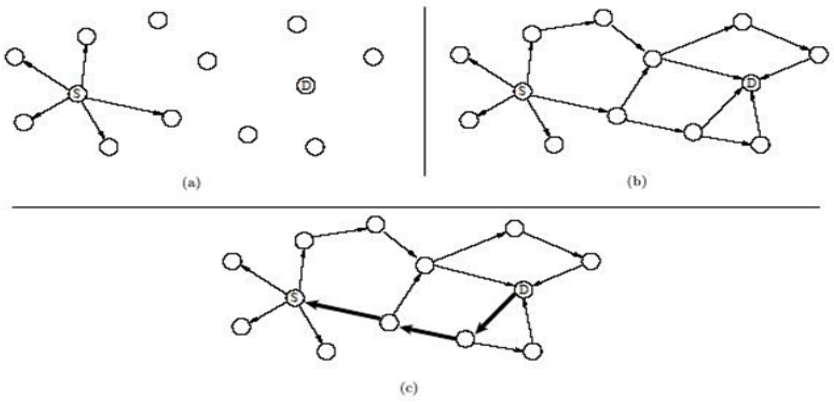

Figure 2.1: (a)Sender broadcasts route request (b)Intermediate nodes stamp and forward request (c) Destination sends a (source routed) reply containing path

The second main function in DSR is route maintenance which handles link breaks. A link break occurs when two nodes on a path are no longer in transmission range. If an intermediate node detects a link break when forwarding a packet to the next node in the route path, it sends back a message to the source notifying it of that link break. The source must try another path or do a route discovery if it does not have another path.

2.3 Network Security

A security protocol for ad hoc wireless networks should satisfy the following requirements. The requirements listed below should in fact be met by security protocols for other types of networks also.

Confidentiality: The data sent by the sender (source node) must be comprehensible only to the intended receiver (destination node). Though an intruder might get hold of the data being sent, he/she must not be able to derive any useful information out of the data. One of the popular techniques used for ensuring confidentiality is data encryption.

Integrity: The data sent by the source node should reach the destination node as it was sent: unaltered. In other words, it should not be possible for any malicious node in the network to tamper with the data during transmission.

Availability: The network should remain operational all the time. It must be robust enough to tolerate link failures and also be capable of surviving various attacks mounted on it. It should be able to provide the guaranteed services whenever an authorized user requires them.

Non-repudiation: Non-repudiation is a mechanism to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message. Digital signatures, which function as unique identifiers for each user, much like a written signature, are used commonly for this purpose.

Authentication: Enables a node to ensure the identity of the peer node it is communicating with. Without authentication, an adversary could masquerade a node, thus gaining unauthorized access to resource and sensitive information so it is

interfering with the operation of other nodes. [2]

2.4 Issues And Challenges for Security

Hao yang [3] covers basic challenge and issues related to secure routing. Designing a foolproof security protocol for ad hoc wireless is a very challenging task. This is mainly because of certain unique characteristics of ad hoc wireless networks, namely, shared broadcast radio channel, insecure operating environment, lack of central authority, lack of association among nodes, limited availability of resources, and physical vulnerability. A detailed discussion on how each of the above mentioned characteristics causes difficulty in providing security in ad hoc wireless networks is given below.

2.4.1 Issues And Challenges for MANET Security

Shared broadcast radio channel: Unlike in wired networks where a separate dedicated transmission line can be provided between a pair of end users, the radio channel used for communication in ad hoc wireless networks is broadcast in nature and is shared by all nodes in the network. Data transmitted by a node is received by all nodes within its direct transmission range. So a malicious node could easily obtain data being transmitted in the network. This problem can be minimized to a certain extent by using directional antennas.

Insecure operational environment: The operating environments where ad hoc wireless networks are used may not always be secure. One important application of such networks is in battlefields. In such applications, nodes may move in and out of hostile and insecure enemy territory, where they would be highly vulnerable to security attacks.

Lack of central authority: In wired networks and infrastructure-based wireless networks, it would be possible to monitor the traffic on the network through certain

important central points (such as routers, base stations, and access points) and implement security mechanisms at such points. Since ad hoc wireless networks do not have any such central points, these mechanisms cannot be applied in ad hoc wireless networks.

Lack of association: Since these networks are dynamic in nature, a node can join or leave the network at any point of the time. If no proper authentication mechanism is used for associating nodes with a network, an intruder would be able to join into the network quite easily and carry out his/her attacks.

Limited resource availability: Resources such as bandwidth, battery power, and computational power (to a certain extent) are scarce in ad hoc wireless networks. Hence, it is difficult to implement complex cryptography-based security mechanisms in such networks.

Physical vulnerability: Nodes in these networks are usually compact and handheld in nature. They could get damaged easily and are also vulnerable to theft. [2]

2.4.2 Issues and Challenge for Routing in MANET Security

Detection of malicious node Node is participant in route and do the misuse of information.

Guarantee of correct route discovery We have to check the correctness of route.

Confidentiality of network topology Topology discover by malicious node so it create traffic or DOS attack.

Stability against attacks One node first participant in network. After some time it is work as a malicious node and disturb the routing process. Node must be stable not change the state.

2.5 Network Security Attacks

Kai Inkinen [4]. describes adhoc network, different types of Routing algorithm. It also provided security attack are done by the attacker.

Attacks on ad hoc wireless networks can be classified into two broad categories, namely, **Passive and Active** attacks.

Passive attack does not disrupt the operation of the network; the adversary snoops the data exchanged in the network without altering it. Here, the requirement of confidentiality can be violated if an adversary is also able to interpret the data gathered through snooping. Detection of passive attacks is very difficult since the operation of the network itself does not get affected. One way of overcoming such problems is to use powerful encryption mechanisms to encrypt the data being transmitted, thereby making it impossible for eavesdroppers to obtain any useful information from the data overheard.

Active attack attempts to alter or destroy the data being exchanged in the network, thereby disrupting the normal functioning of the network. Active attacks can be classified further into two categories, namely, **External and Internal** attacks.

External attacks are carried out by nodes that do not belong to the network. These attacks can be prevented by using standard security mechanisms such as encryption techniques and firewalls.

Internal attacks are from compromised nodes that are actually part of the network. Since the adversaries are already part of the network as authorized nodes, internal attacks are more severe and difficult to detect when compared to external attacks.

2.5.1 Internal Attack

Wormhole : The wormhole attack is one of the most powerful presented here since it involves the cooperation between two malicious nodes that participate in the network. One attacker, say node A, captures routing traffic at one point of the network and tunnels them to another point in the network, say to node B, that shares a private communication link with A. Node B then selectively injects tunneled traffic back into the network (see Figure 2.2). The connectivity of the nodes that have established routes over the wormhole link is completely under the control of the two colluding attackers.[5]

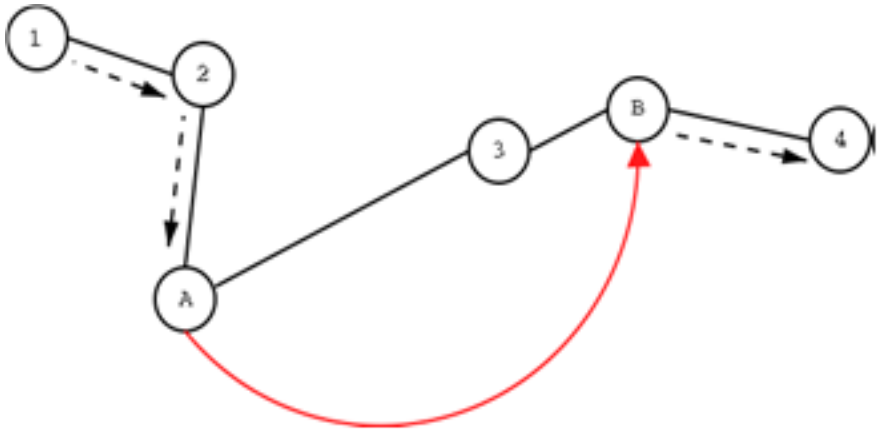

Figure 2.2: A wormhole attack performed by colluding malicious nodes A and B.

Black hole : In a black hole attack a malicious node injects false route replies to the route requests it receives advertising itself as having the shortest path to a destination. These fake replies can be fabricated to divert network traffic through the malicious node for eavesdropping, or simply to attract all traffic to it in order to perform a denial of service attack by dropping the received packets.[6]

Byzantine attack: A compromised intermediate node works alone, or a set of compromised intermediate nodes works in collusion and carry out attacks such as creating routing loops, **forwarding packets through non-optimal paths, or**

selectively dropping packets, which results in disruption or degradation of the routing services.

Resource consumption attack: This is also known as the sleep deprivation attack. An attacker or a compromised node can attempt to consume battery life by requesting excessive route discovery, or by forwarding unnecessary packets to the victim node.

2.5.2 Routing Attack

Kai Inkinen [4] provided routing attack is done by the attacker.

Routing table overflow attack: A malicious node advertises routes that go to non-existent nodes to the authorized nodes present in the network. It usually happens in proactive routing algorithms, which update routing information periodically. The attacker tries to create enough routes to prevent new routes from being created. The proactive routing algorithms are more vulnerable to table overflow attacks because proactive routing algorithms attempt to discover routing information before it is actually needed. An attacker can simply send excessive route advertisements to overflow the victim's routing table.

Routing cache poisoning attack: In route cache poisoning attacks, attackers take advantage of the promiscuous mode of routing table updating, where a node overhearing any packet may add the routing information contained in that packet header to its own route cache, even if that node is not on the path. Suppose a malicious node M wants to poison routes to node X. M could broadcast spoofed packets with source route to X via M itself; thus, neighboring nodes that overhear the packet may add the route to their route caches.

Rushing attack: Two colluded attackers use the tunnel procedure to form a wormhole. If a fast transmission path (e.g. a dedicated channel shared by attackers) exists between the two ends of the wormhole, the tunneled packets can propagate faster than those through a normal multi-hop route. This forms the rushing attack. The rushing attack can act as an effective denial-of-service attack against all currently proposed on-demand MANET routing protocols, including protocols that were designed to be secure, such as ARAN and Ariadne.

Replay : An attacker that performs a replay attack injects into the network routing traffic that has been captured previously. This attack usually targets the freshness of routes, but can also be used to undermine poorly designed security solutions.

Denial of service: Denial of service attacks aim at the complete disruption of the routing function and therefore the whole operation of the ad hoc network. Specific instances of denial of service attacks include the routing table overflow and the sleep deprivation torture. In a routing table overflow attack the malicious node floods the network with bogus route creation packets in order to consume the resources of the participating nodes and disrupt the establishment of legitimate routes. The sleep deprivation torture aims at the consumption of batteries of a specific node by constantly keeping it engaged in routing decisions.

Man in middle attack: An attacker sits between the sender and the receiver sniffs any information being sent between two ends. In some cases the attacker may impersonate the sender to communicate with the receiver, or impersonate the receiver to reply to the sender.[7]

Chapter 3

Security Scheme & IDS

3.1 Security Scheme

There are **two main approaches** in securing ad hoc environments currently utilized.

The first approach is the intrusion detection approach that aims in enabling the participating nodes to detect and avoid malicious behavior in the network without changing the underlined routing protocol or the underling infrastructure. Although the intrusion detection field and its applications are widely researched in infrastructure networks it is rather new and faces greater difficulties in the context of ad hoc networks.

The second approach is secure routing that aims in designing and implementing routing protocols that have been designed from scratch to include security features. Mainly the secure protocols that have been proposed are based on existing ad hoc routing protocols like AODV and DSR but redesigned to include security features. In the following sub sections we briefly present the two approaches in realizing security schemes that can be employed in ad hoc networking environments.

3.1.1 Intrusion Detection System (IDS)

Intrusion is defined as "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource". Intrusion protection techniques works as the first line of defense. However, intrusion protection alone is not sufficient since there is no perfect security in any system, especially in the field of ad hoc networking due to its fundamental vulnerabilities.

Therefore, intrusion detection can work as the second line of protection to capture audit data and perform traffic analysis to detect whether the network or a specific node is under attack. The two type of nodes are in under attack on a network. [8]

Selfish nodes: It doesn't cooperate for selfish reasons, such as saving power. Even though the selfish nodes do not intend to damage other nodes, the main threat from selfish nodes is the dropping of packets, which may affect the performance of the network severely.

Malicious nodes: It has the intention to damage other nodes, and battery saving is not a priority. Without any incentive for cooperating, network performance can be severely degraded.

Once an intrusion has been detected then measures can be taken to minimize the damages or even gather evidence to inform other legitimate nodes for the intruder and maybe launch a countermeasure to minimize the effect of the active attacks.

3.1.2 Secure Routing

This approach attempts to design secure routing protocols for ad hoc networks. These protocols are either completely new stand-alone protocols, or in some cases incorporations of security mechanisms into existing protocols like AODV and DSR.

Generally the existing secure routing protocols that have been proposed can be broadly classified into two categories, those that use hash chains, and those that in order to operate require predefined trust relationships.

The Secure Efficient Ad hoc Distance vector routing protocol (SEAD) employs the use of hash chains to authenticate hop counts and sequence numbers. SEAD is based on the design of the proactive ad hoc routing protocol DSDV. It provides loop freedom and protects the nodes from impersonation and several other attacks.

Another secure routing protocol is Ariadne. Ariadne assumes the existence of a shared secret key between two nodes and uses a message authentication code (MAC) in order to authenticate point-to-point messages between nodes.

SAODV proposes a set of extensions that secure the AODV routing packets. For authenticating the non-mutable fields it uses cryptographic signatures, while one-way hash chains are used for securing every different route discovery process. In order to carry out the asymmetric cryptography it requires the existence of a key management mechanism. The protocol assumes that each node knows a priori the public key of the certification authority that will be used to authenticate the other participating nodes.

Another protocol is the Security-aware Ad hoc Routing (SAR) that extends on-demand ad hoc routing protocols like AODV and DSR. The main aspect of SAR is that it introduces a new security metric in the route discovery and maintenance process, treating secure routing as a quality of service (QoS) issue. SAR uses security attributes such as trust values and trust relationships in order to define this metric.

3.2 IDS Architecture

Liz Nickels [9] has provided information related to Intrusion detection system (IDS). It has provided IDS architecture and related their component. In Distributed Architecture use different mechanism for detecting to node.

Bo Sun [10] paper has been performed about intrusion detection in the areas of mobile ad hoc networks and wireless sensor networks. Then, paper focus on their intrusion detection capabilities. Specifically, we present the challenge of constructing intrusion detection systems for mobile ad hoc networks and wireless sensor networks; survey the existing intrusion detection techniques

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

A network-based IDS runs on a gateway of a network and captures and examines the network traffic that flows through it. Obviously this approach is not suitable for ad hoc networks since there is no central point that allows monitoring of the whole network.

A host-based IDS relies on capturing local network traffic to the specific host. This data is analyzed and processed locally to the host and is used either to secure the activities of this host, or to notify another participating node for the malicious action of the node that performs the attack.

3.2.1 Stand Alone IDS

In this architecture, each host has a IDS and detect attacks independently. There is no cooperation between nodes and all decision is based on local nodes(Figure 3.1). This architecture is not effective enough but can be utilized in an environment where

not all nodes are capable of running IDS

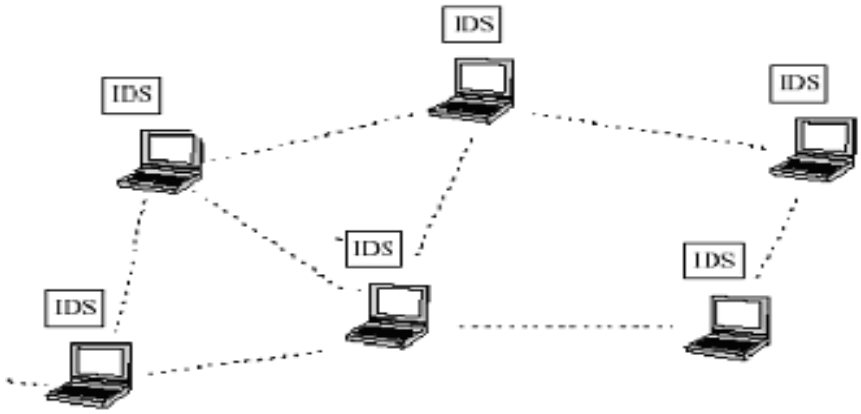

Figure 3.1: Stand Alone Architecture

3.2.2 Distributed And Cooperative IDS

Intrusion detection and response systems should be both distributed and cooperative to suite the needs of mobile ad-hoc networks. In our proposed architecture (Figure 3.2), every node in the mobile ad-hoc network participates in intrusion detection and response. Each node is responsible for detecting signs of intrusion locally and independently, but neighboring nodes can collaboratively investigate in a broader range.

In the systems aspect, individual IDS agents are placed on each and every node. Each IDS agent runs independently and monitors local activities (including user and systems activities, and communication activities within the radio range). It detects intrusion from local traces and initiates response. If anomaly is detected in the local data, or if the evidence is inconclusive and a broader search is warranted, neighboring IDS agents will cooperatively participate in global intrusion detection actions. These individual IDS agent collectively form the IDS system to defend the mobile ad-hoc network.

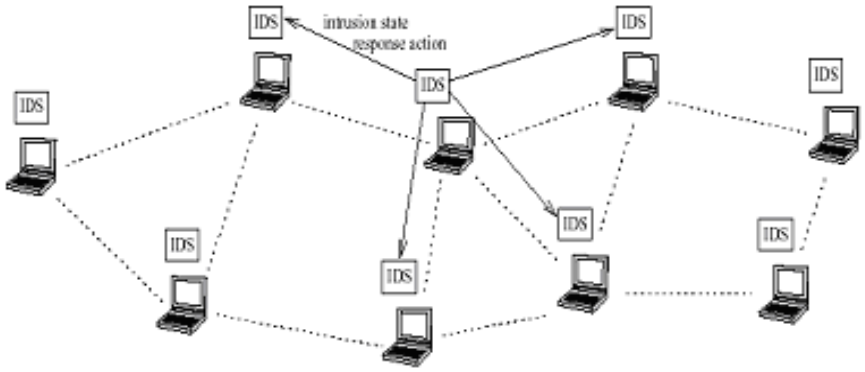

Figure 3.2: Distributed and cooperative Architecture

The internal of an IDS agent can be fairly complex, but conceptually it can be structured into six pieces (Figure 3.3). The data collection module is responsible for gathering local audit traces and activity logs. Next, the local detection engine will use these data to detect local anomaly. Detection methods that need broader data sets or that require collaborations among IDS agents will use the cooperative detection engine.

Intrusion response actions are provided by both the local response and global response modules. The local response module triggers actions local to this mobile node, for example an IDS agent alerting the local user, while the global one coordinates actions among neighboring nodes, such as the IDS agents in the network electing a remedy action. Finally, a secure communication module provides a high configuration dence communication channel among IDS agents.

3.3 Mechanism

Xia Wang [11] describes the various IDS system with different mechanism that used for detection of node.

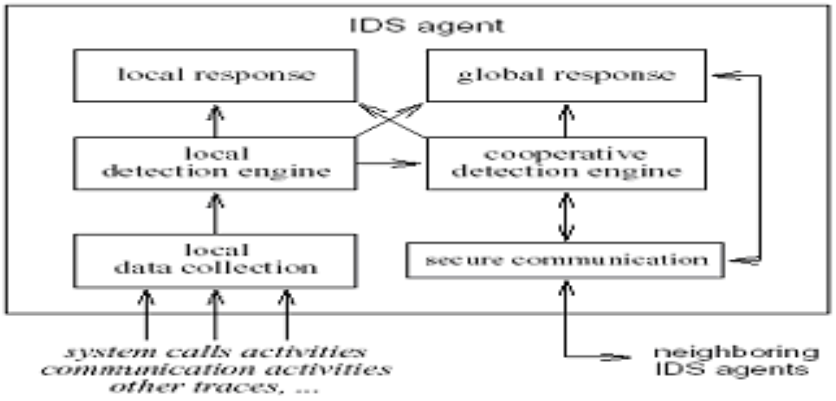

Figure 3.3: Distributed and cooperative architecture component

In the IDS system the mechanism is used to identify or detect the node in network. The different mechanisms are used with different architecture and according to different routing protocol mechanism is change. We have to first check that which architecture used in network for IDS and also which routing protocol is used in network In that stand alone architecture we are using Watchdog and Pathrater.[12]

3.3.1 Watchdog

The watchdog and pathrater scheme consists of two extensions to the DSR routing protocol that attempt to detect and mitigate the effects of nodes that do not forward packets although they have agreed to do so. This misbehavior may be due to malicious or selfish intent, or simply the result of resource overload. Although the specific methods proposed build on top of DSR. The watchdog extension is responsible for monitoring that the next node in the path forwards data packets by listening in promiscuous mode. It identifies as misbehavior nodes the ones that fail to do so.

Every node that participates in the ad hoc network employs the watchdog functionality in order to verify that its neighbors correctly forward packets. When a node transmits a packet to the next node in the path, it tries to promiscuously listen if the

next node will also transmit it. Furthermore, if there is no link encryption utilized in the network, the listening node can also verify that the next node did not modify the packet before transmitting it.

The watchdog of a node maintains copies of recently forwarded packets and compares them with the packet transmissions overheard by the neighboring nodes. Positive comparisons result in the deletion of the buffered packet and the freeing of the related memory. If a node that was supposed to forward a packet fails to do so within a certain timeout period, the watchdog of an overhearing node increments a failure rating for the specific node.

This effectively means that every node in the ad hoc network maintains a rating assessing the reliability of every other node that it can overhear packet transmissions from. A node is identified as misbehaving when the failure rating exceeds a certain threshold bandwidth. The source node of the route that contains the offending node is notified by a message send by the identifying watchdog. [5]

Watchdog Example

In given figure 3.4 is a packet is traveling from S to D. A can overhear B and tell whether B has forwarded the packet. Buffer is maintained for recently sent packets. The overheard packet is compared with the sent packet. If there is a match, discard the packet. If the packet stays till a timeout, increment the failure tally for the node. If tally exceeds a threshold, declare the node as misbehaving.[1]

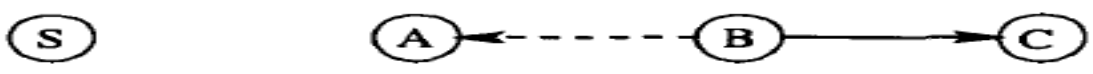

Figure 3.4: A Example of WatchDog

3.3.2 Pathrater

The pathrater assesses the results of the watchdog and selects the most reliable path for packet delivery. One of the base assumptions of this scheme is that malicious nodes do not collude in order to circumvent it and perform sophisticated attacks against the routing protocol.

The pathrater extension to DSR selects routes for packet forwarding based on the reliability rating assigned by the watchdog mechanism. Specifically, a metric for each path is calculated by the pathrater by averaging the reliability ratings of the nodes that participate in the path. This path metric allows the pathrater to compare the reliability of the available paths, or to emulate the shortest path algorithm when no reliability ratings have been collected. The pathrater selects the path with the highest metric when there are multiple paths for the same destination node.

The algorithm followed by the pathrater mechanism initially assigns a rating of 1.0 to itself and 0.5 to each node that it knows through the route discovery function. The nodes that participate on the active paths have their ratings increased by 0.01 at periodic intervals of 200 milliseconds to a maximum rating of 0.8. A rating is decremented by 0.05 when a link breakage is detected during the packet forwarding process to a minimum of 0.0. The rating of -100 is assigned by the watchdog to nodes that have been identified as misbehaving. When the pathrater calculates a path value as negative this means that the specific path has a participating misbehaving node.

The watchdog and pathrater extensions facilitate the identification and avoidance of misbehaving nodes that participate in the routing function. The identification is based on overheard transmissions and the selection of reliable routes is based on the calculated reliability of the paths.

Chapter 4

Mobility Model & NS

4.1 Introduction Mobility Model

Mobile nodes within an ad hoc network move from location to location; so it is necessary to develop and use mobility models that accurately represent movements of the Mobile nodes that will eventually utilize the given protocol. Only in this type of scenario is it possible to determine whether or not the proposed protocol will be useful when implemented. Therefore, it is imperative that accurate mobility models are chosen.

4.1.1 Random Waypoint Model

Each node chooses a random destination and moves towards it with a random velocity chosen from [0, Vmax] After reaching the destination, the node stops for a duration defined by the "pause time" parameter After this duration, it again chooses a random destination and repeats the whole process again until the simulation ends

Parameters: Max Velocity Vmax, Pause time T.

4.1.2 Reference Point Group Model

Each group has a logical center (group leader) that determines the group's motion behavior.

Group Mobility Generator

In simulation, we use two sets of trace files. That is given in Figure 4.1

Single group: all nodes move within one group

Multiple group: each group moves independent of each other and in an overlapping fashion.

Input: -Mobility trace file of group leaders

Output:-Mobility trace file of all nodes

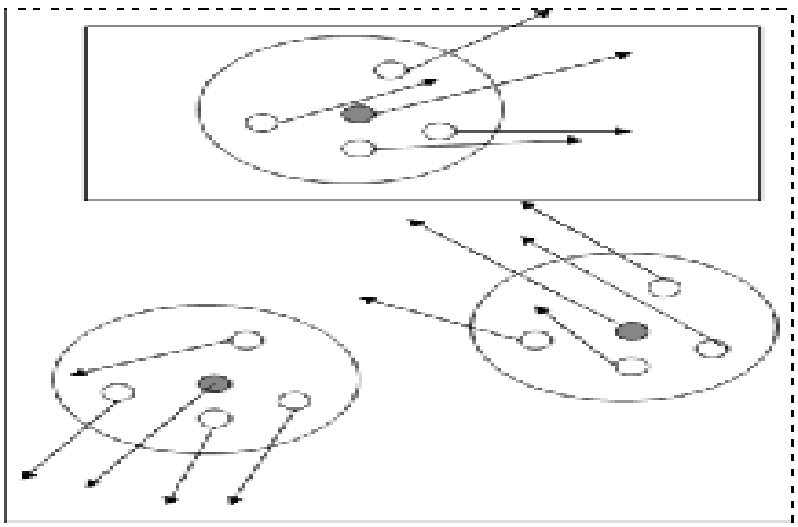

Figure 4.1: Group mobility generator

4.1.3 Freeway Model

Each mobile node is restricted to its lane (shown in Figure 4.2) on the freeway The velocity of mobile node is temporally dependent on its previous velocity If two mobile

nodes on the same freeway lane are within the Safety Distance (SD), the velocity of the following node cannot exceed the velocity of preceding node.

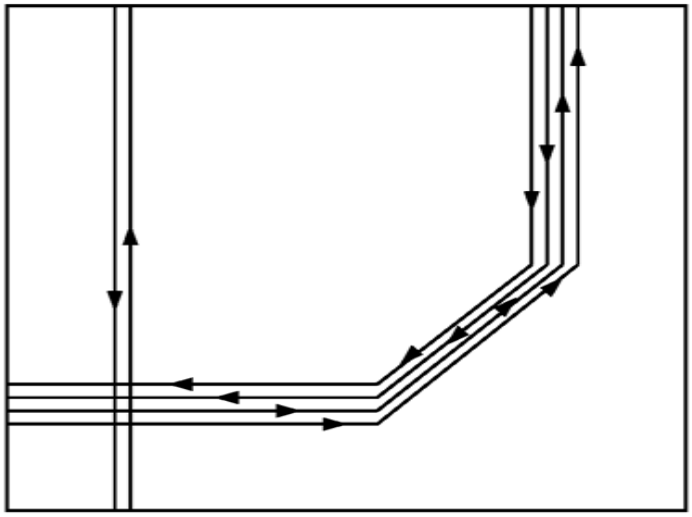

Figure 4.2: Freeway mobility generator Model

4.2 Step To Create Adhoc Wireless Network

Step-1 Create Mobile Node

Step-2 Generate Scenario

Step-3 Create Agents

Step-4 Run simulation

4.2.1 Create Mobile Node

MobileNode is a split object. In addition to the basic node model(shown in Figure 4.3), it consists of a network stack. The network stack for a mobile node consists of a link layer (LL), an ARP module connected to LL, an interface priority queue (IFq), a mac layer(MAC), a network interface (netIF), all connected to a common wireless channel. These network components are created and plumbed together in

OTcl. A packet sent down the stack flows through the link layer (and ARP),the Interface queue, the MAC layer, and the physical layer. At the receiving node, the packet then makes its way up the stack through the Mac, and the LL. [13]

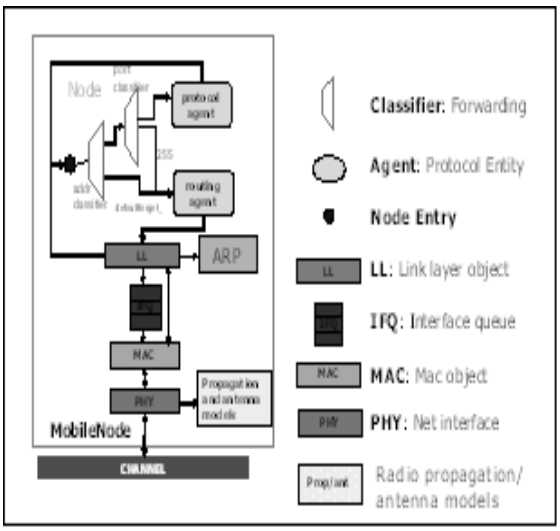

Figure 4.3: Network Stack Model

$nsnode - config - adhocRouting$ $opt(adhocRouting)$-dsdv/dsr/aodv/tora

-llType $opt(ll)$ - specifies link layer object

-macType $opt(mac)$ - specifies mac object

-ifqType $opt(ifq)$ - specifies ifq object

-ifqLen $opt(ifqlen)$ - specifies length of ifq

-antType $opt(ant)$ - specifies antenna object

-propInstance [new $opt(prop)$]- propagation object

-phyType $opt(netif)$ - specifies physical layer object

-channel [new $opt(chan)$] - specifies channel object

-topoInstance $topo - specifies topography

-wiredRouting OFF - for wired cum wireless simulations

-agentTrace ON - specifies agent level trace ON/OFF

-routerTrace OFF - specifies router level trace ON/OFF

-macTrace OFF - specifies mac level trace ON/OFF

A mobile node is created using the following procedure:

for { set j 0 } { $j < $opt(nn) } { incr j }

{

set node ($j) [$ns node]

$node ($i) random-motion 0 ; - disable random motion

}

This procedure creates a mobile node creates an adhoc-routing routing agent as specified, creates the network stack consisting of a link layer, interface queue, mac layer, and a network interface with an antenna, uses the defined propagation model, interconnects these components and connects the stack to the channel.

4.2.2 Generate Scenario

In the scenario generation we have to define the some parameter like number of mobile node, node movement and speed of movement, and the area in which nodes moving etc [14]

Define Parameter

For example ,generating a scenario with 4 nodes, moving with a maximum speed of 20m/s, with a pause time of 10s, within a topology boundary of 670 x 670, for a simulation time of 400s. We specified this scenario in a separate scenario file, scene-4-test. We generated this scenario file by typing the following command in ns-2.33/indep-utils/cmugen/setdest directory:

./setdest -n 4 -p 10.0 -M 20.0 -t 400 -x 670 -y 670 > scene-4-test

Node Movement

The Mobile Node is designed to move in a three dimensional topology. However, the third dimension (Z) is not used. That is the Mobile Node is assumed to move always on a flat terrain with Z always equal to 0. Thus the Mobile Node has X, Y, Z(=0) co-ordinates that is continually updated as the nodes move.

We first need to define the topography creating the mobile nodes. Normally flat topology is created by specifying the length and width of the topography using the following primitive:

$*set* topo [new Topography]

$*topo* load flat grid $*opt*($x$) $*opt*($y$)

where opt(x) and opt(y) are the boundaries used in simulation.

There are two mechanisms to induce movement in mobile nodes. In the first method, starting position of the node and its future destinations may be set explicitly. These directives are normally included in a separate movement scenario file.[15]

The start-position and future destinations for a mobile node may be set by using the following APIs:

$*node* set X <x1>

$*node* set Y <y1>

$*node* set Z <z1>

$*ns* at $*time* $*node* setdest <x2> <y2> <speed>

At $*time* sec, the node would start moving from its initial position of (x1,y1) towards a destination (x2,y2) at the defined speed.

In this method the node-movement-updates are triggered whenever the position of the node at a given time is required to be known. This may be triggered by a query

from a neighboring node seeking to know the distance between them, or the setdest directive described above that changes the direction and speed of the node.

The second method employs random movement of the node. The primitive to be used is:

$*mobilenode* start

which starts the mobile node with a random position and have routine updates to change the direction and speed of the node. The destination and speed values are generated in a random fashion.

4.2.3 Create Agent

Agents are used in the implementation of protocols at various layers. They represent endpoints where network-layer packets are constructed or consumed. The different agents currently supported by NS at the transport layer like TCP, TCP Reno. NS also has routing agents implementing the different routing protocols like DSDV, TORA, AODV and DSR and for application layer we have CBR traffic agent. Once the agent is created then we have to connect different agent so communication between every layer and packet transfer is done.

$*set* udp [new Agent/UDP]

creates a udp agent. Users can create any agent or traffic sources in this way.

$*ns* attach-agent node agent

attaches an agent object created to a node object.

$*ns* connect agent1 agent2

Connects the two agents specified. After two agents that will communicate with each other are created, the next thing is to establish a logical network connection

between them. This line establishes a network connection by setting the destination address to each others' network and port address pair.

4.2.4 Run Simulation

$$ns$$ run starts the simulation.

$$ns$$ at 5.0 "finish"

tells the simulator object to execute the 'finish' procedure after 5.0 seconds of simulation time

4.3 Trace File of DSR

In order to obtain results from simulations, we need to know what exactly happens during a simulation run. NS realizes this by generation of event logs that can be analyzed offline, after a simulation. These event log files [16]however, contain only events of packets being sent, received or dropped, so called Packet Traces. NS does not support the logging of more abstract events, i.e. related to connection establishment. We now give an overview of the new wireless trace format generated by NS simulations.

s 606.210364161 39 RTR — 1306 DSR 44 [13a a 27 800] ——- [39:255 8:255 255 8] 2 [0 0] [0 0 0 0->0] [1 1 8 39->10]

s: means send

606.210364161: time stamp

39: node id

RTR: means router message

1306: uid of this packet

DSR: DSR agent

44: size in the common header hdrcmn()

[13a b 27 800] MAC detail:

13a: means the expected transmission time

b:means the receiving node: 11

27:means the sending node is 39

800: IP header: 0x0800,(ETHERTYPE ARP is 0x0806)

[39:255 8:255 255 8] IP detail:

source address: IP 39 means 0.0.0.39 port 255

dst address: IP 8 means 0.0.0.8 port 255

TTL: 255

Next-hop: 8

2 [0 0] DSR detail:

2: numaddrs()

first 0 route-request option, this is not a route request

the second 0 is labeled for sequence number

[0 0 0 0->0] route-reply option:

first 0: "route-reply?"

second 0:"Rreq seqno"

third 0: "reply length"

fourth 0: "dst of src route"

fifth 0:"src of the src route"

[1 1 8 39->10]

1: shows this is a route error

1: number of route errors

8: tp notify node 8.

39->10: link 39-10 is broken

4.4 AWK And XGRAPH

Awk scannes ascii files or standard input. It can search strings easily and then has a lot of possibilities to process the found lines and output them in the new format. It does not change the input file but sends it's results onto standard output.

An awk script can have three types of blocks. One of them must be there. The BEGIN block is processed before the file is checked. The block runs for every line of input and the END block is processed after the final line of the input file.

Xgraph is a plotting program which can be used to create graphic representations of simulation results. You can create output files in your Tcl scripts, which can be used as data sets for xgraph. Call xgraph to display the results with the command "xgraph <data-file>".

Chapter 5

Implementation

5.1 Implementation Environment

Network simulator provides proper environment for generating adhoc network or any wireless network. To implement adhoc network in ns we have to set different parameter.That is given below.

Scenario :- containing 50 mobile nodes moving within 670mX670m flat topology

Movement Speed 0 to 20 m/s Using DSR ad hoc routing protocol

Random Way point mobility model

0 Pause Time

60 Pause Time

UDP and CBR traffic with 10 connection

Simulation Time 200 second

Using 2 movement Pattern and 2 Pause Time Generate 4 scenario. That is given in following figure nos 5.1, 5.2, 5.3, 5.4

After generating all 4 scenario implement any one misbehavior. For Forwarded packet drop misbehavior is generates for selects number of misbehaving nodes by 0%,

20%, & 40% node of network. The tables nos I, II, III, IV represents the drop of packet by number of node.

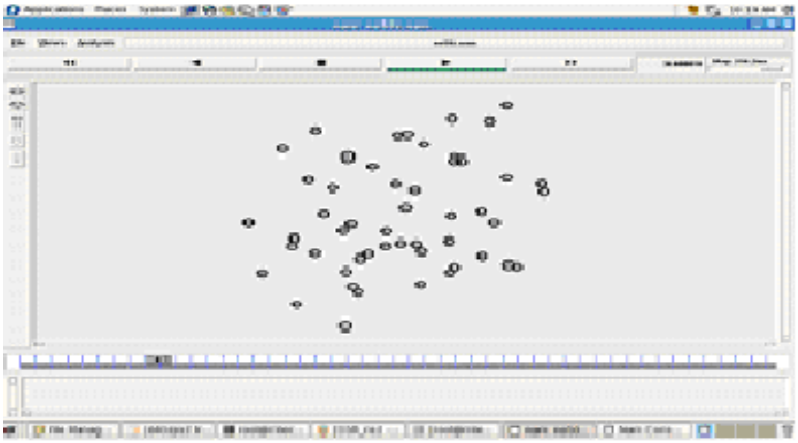

Figure 5.1: Scenario-1 No pause time

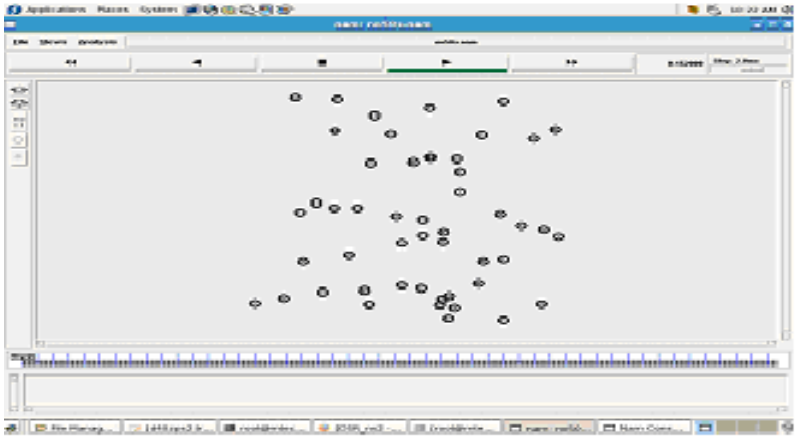

Figure 5.2: Scenario-1 Pause time of 60 seconds

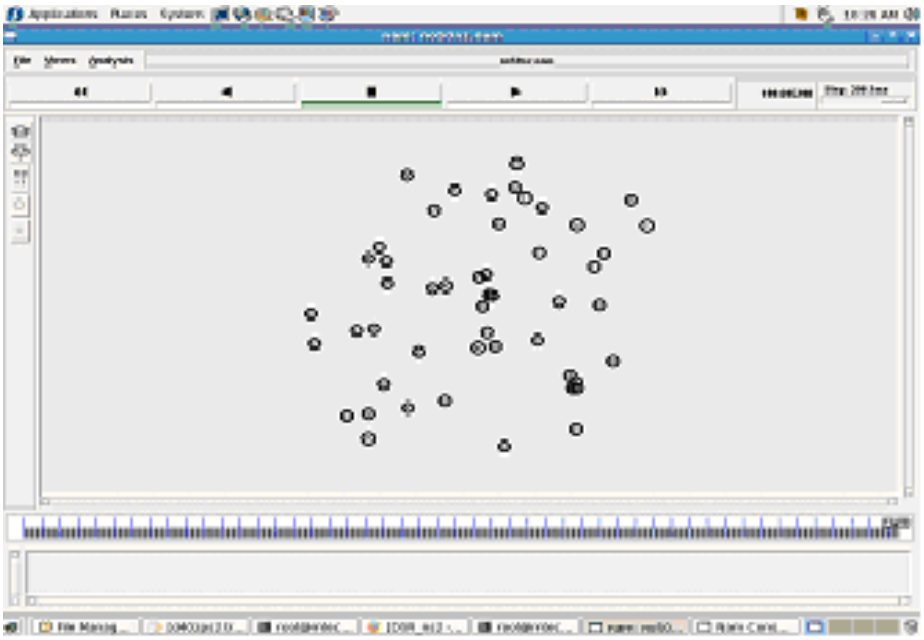

Figure 5.3: Scenario-2 No pause time

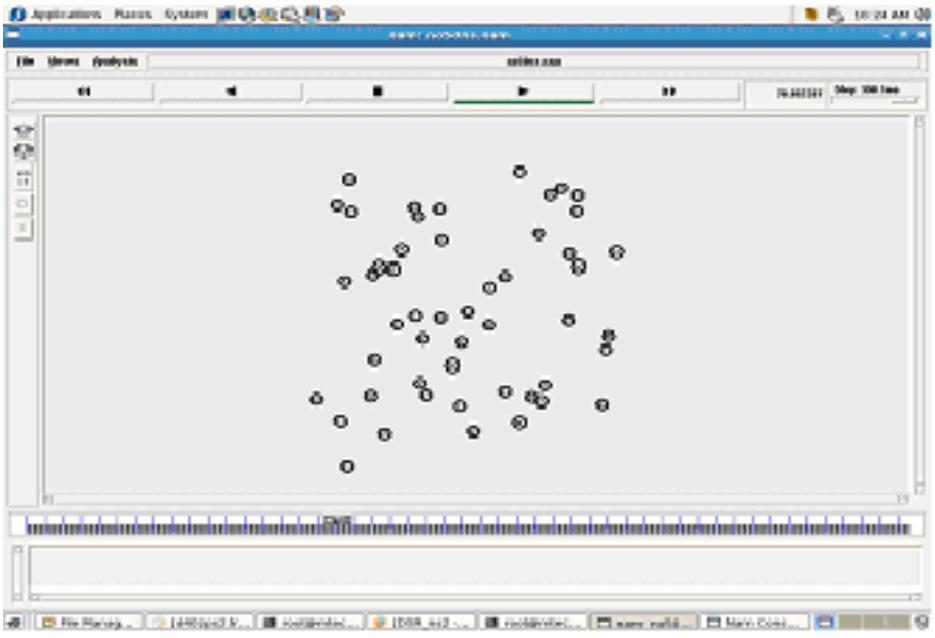

Figure 5.4: Scenario-2 Pause time of 60 seconds

Node number	0 %	20 %	40 %
4th node	0	20	26
5th node	0	0	30
7th node	0	24	30
14th node	0	19	21
16th node	0	1	23
18th node	0	30	40
20th node	0	1	25
22th node	1	20	30
25th node	0	5	32
28th node	0	25	28
30th node	0	2	30
32th node	0	1	27
34th node	0	6	43
36th node	0	15	23
38th node	0	2	30
40th node	0	25	28
42th node	0	2	30
44th node	0	38	40
48th node	0	5	25
49th node	1	1	20

Table I: Drop packet for Scenario-1

Node number	0 %	20 %	40 %
4th node	0	30	20
5th node	0	6	20
7th node	0	50	39
14th node	1	23	22
16th node	0	1	18
18th node	0	45	32
20th node	0	2	15
22th node	2	20	25
25th node	2	1	19
28th node	0	20	30
30th node	2	5	20
32th node	1	10	22
34th node	1	1	24
36th node	2	43	44
38th node	0	0	23
40th node	0	23	30
42th node	1	4	28
44th node	3	28	31
48th node	0	0	44
49th node	0	4	38

Table II: Drop packet for Scenario-1 Pause time 60 seconds

Node number	0 %	20 %	40 %
4th node	0	15	26
5th node	1	1	14
7th node	0	13	15
14th node	2	18	33
16th node	1	3	21
18th node	1	5	18
20th node	0	2	20
22th node	0	34	45
25th node	0	0	33
28th node	1	24	26
30th node	1	1	31
32th node	2	9	27
34th node	2	2	11
36th node	0	13	24
38th node	2	2	30
40th node	0	23	35
42th node	0	0	22
44th node	1	34	43
48th node	2	6	12
49th node	0	3	37

Table III: Drop packet for Scenario-2 no pause time

Node number	0 %	20 %	40 %
4th node	0	20	13
5th node	0	0	22
7th node	0	25	30
14th node	1	31	33
16th node	0	0	22
18th node	1	19	20
20th node	0	0	26
22th node	1	31	33
25th node	1	1	22
28th node	0	12	17
30th node	1	3	21
32th node	0	4	16
34th node	0	2	19
36th node	1	3	14
38th node	0	20	23
40th node	1	21	33
42th node	1	3	19
44th node	0	20	27
48th node	0	1	12
49th node	1	3	25

Table IV: Drop packet for Scenario-2 Pause time 60 seconds

5.2 Parameter Affected

Once the packet drop misbehavior is implemented on all four scenario of ns with different percentage of misbehavior node. During simulation of network some of the parameter are getting affected on network by misbehavior node.

5.2.1 Detection Rate

It is given by number of drop packet by different number of node out of 50 node. If number of drop packet is more then Threshold then it is detect as malicious node. The graphs shown in Figures 5.5, 5.6, 5.7, 5.8 with 0%, 20%, & 40% misbehavior nodes represent that, As threshold changes, detection rate also changes. Detection Rate is calculated for all 4 scenarios.

In DSR routing algorithm, routing path is stored in cache. In that routing path some malicious nodes drop the forwarded packet. In modified DSR, I have removed the routing path that contains malicious nodes. So, removing that routing path improves the performance of the network.

5.2.2 Throughput

Throughput is the percentage of sent packet actually is receive by intended destination. The graph shown in Figures 5.9, 5.10, 5.12, 5.11 represent that throughput are changes according to 0%, 20%, & 40% misbehavior node. Throughput is calculated for all 4 scenario. Throughput is increase in all scenario with modification in DSR routing protocol.

5.2.3 Overhead

Overhead is ratio of routing related transmission to data related transmission.The graph shown in Figures 5.13, 5.14, 5.15, 5.16 represents that overhead are changes

according to 0%, 20%, & 40% misbehavior node. Overhead is calculated for all 4 scenario. Overhead is decrease in all 4 scenario with modification in DSR routing protocol.

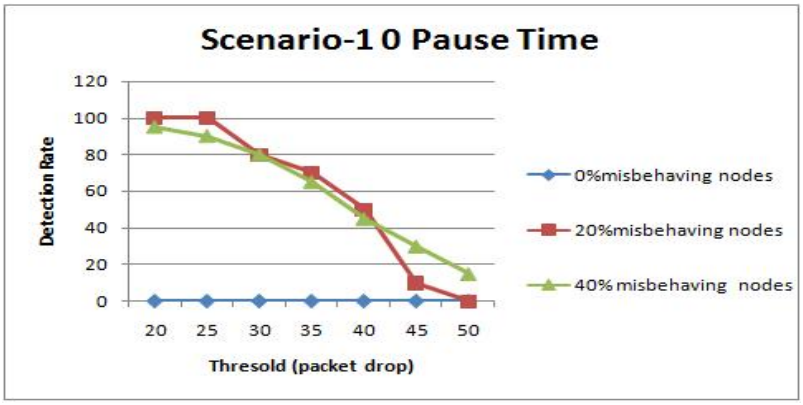

Figure 5.5: Detection Rate for scenario-1 no pause time

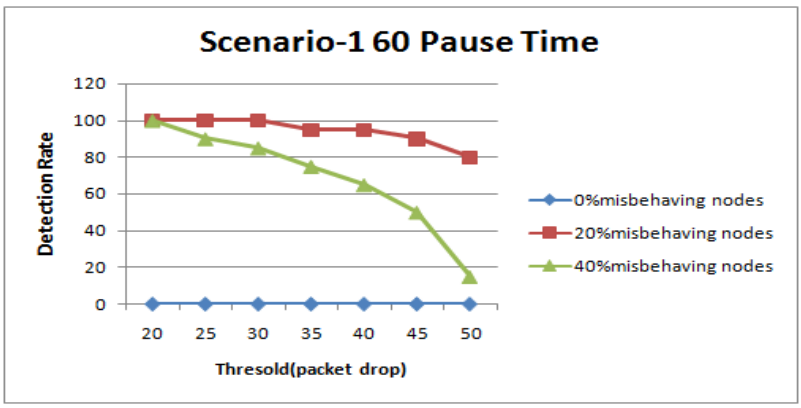

Figure 5.6: Detection Rate for scenario-1 Pause time 60 seconds

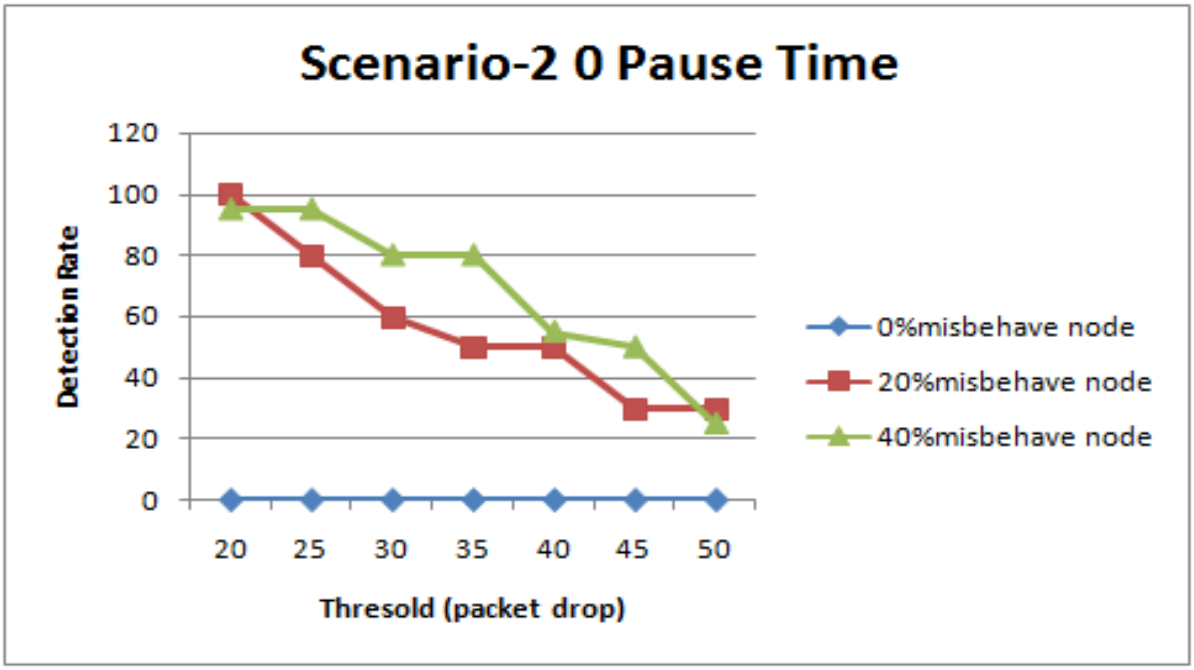

Figure 5.7: Detection Rate for scenario-2

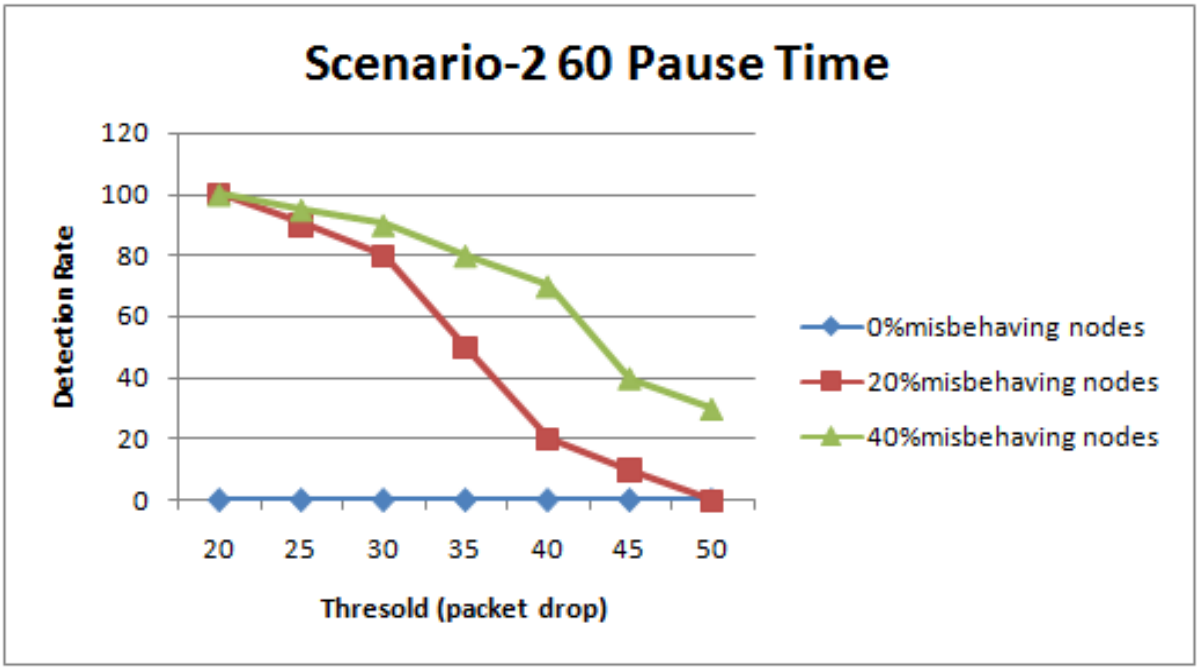

Figure 5.8: Detection Rate for scenario-2 Pause time 60 seconds

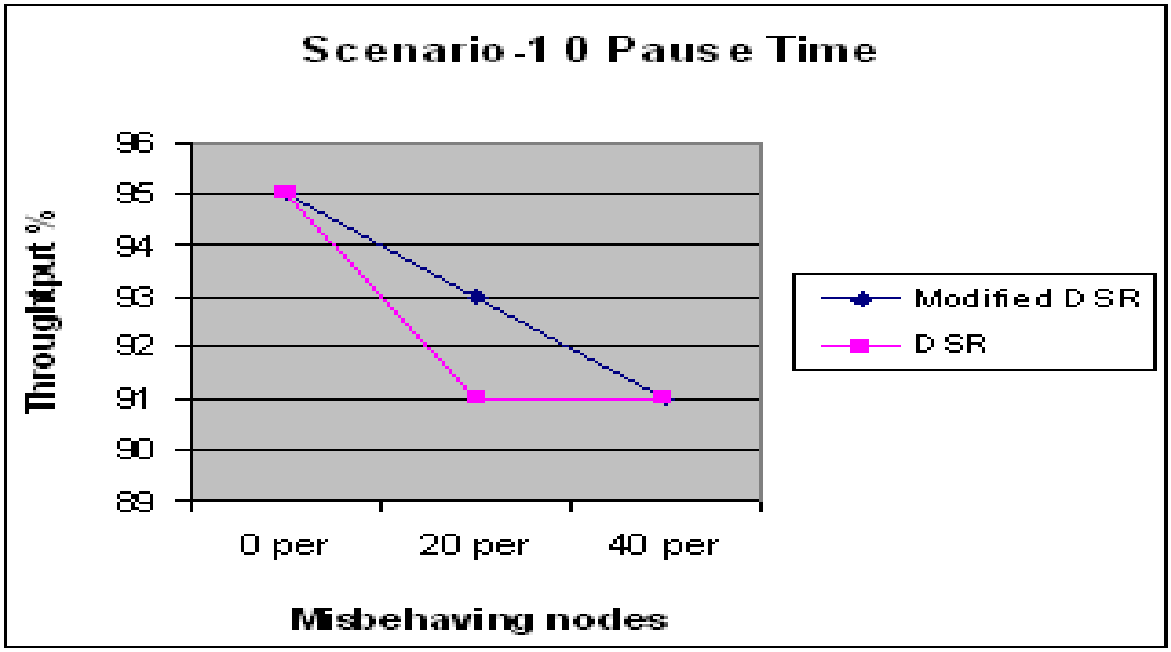

Figure 5.9: Throughput for scenario-1 no pause time

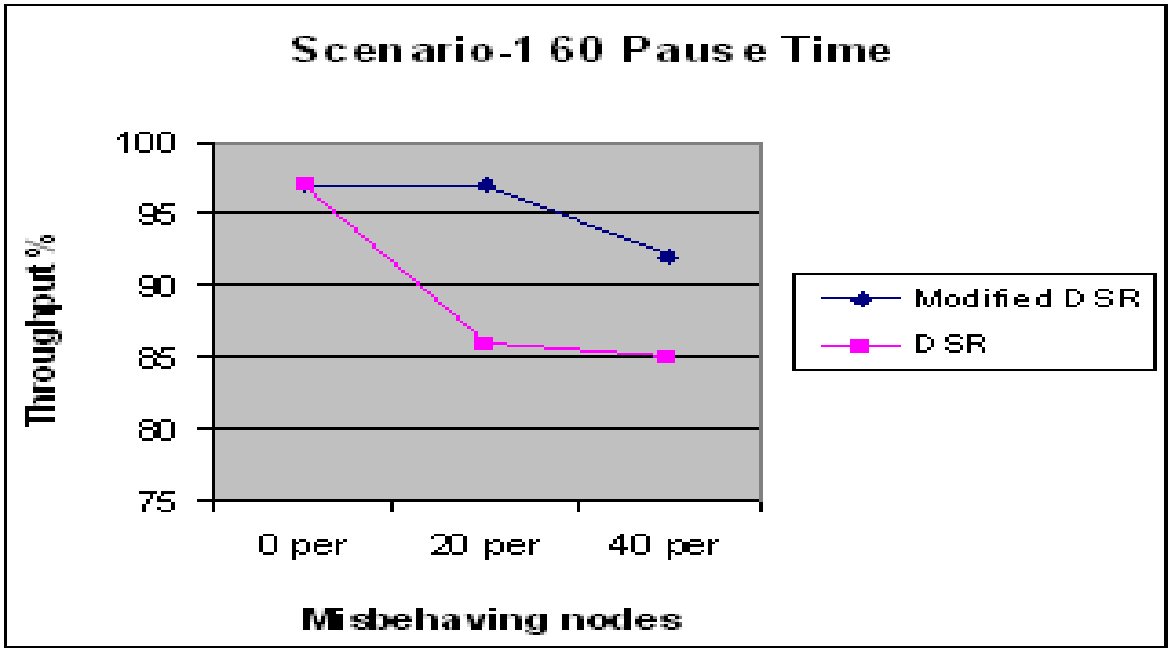

Figure 5.10: Throughput for scenario-1 Pause time 60 seconds

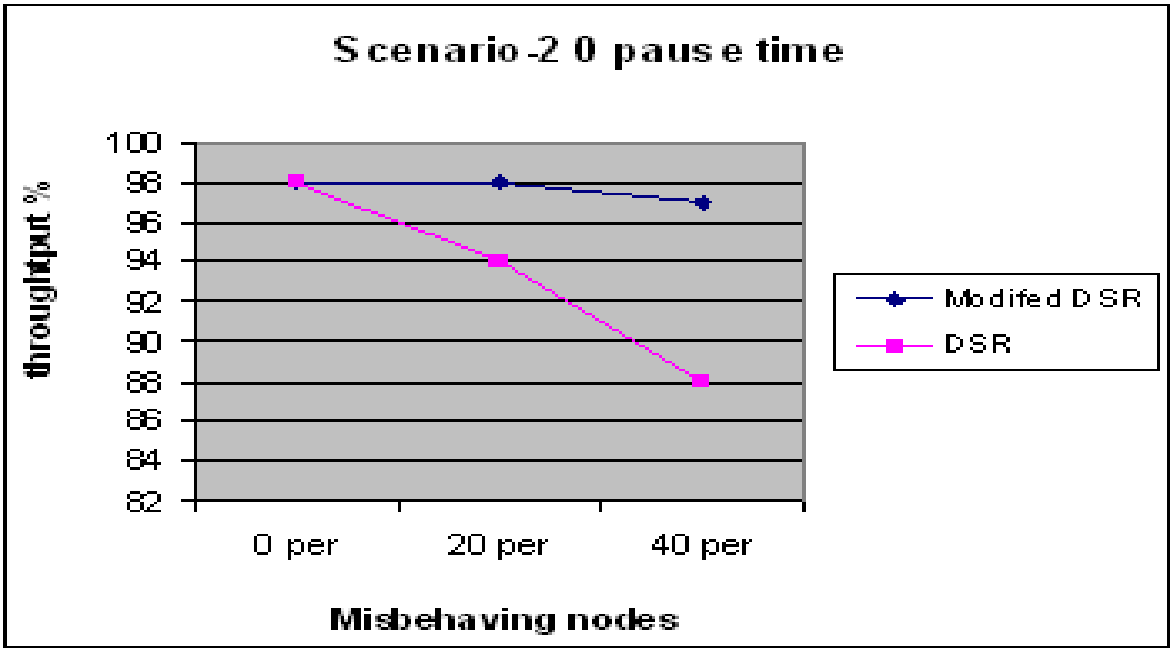

Figure 5.11: Throughput for scenario-2 no pause time

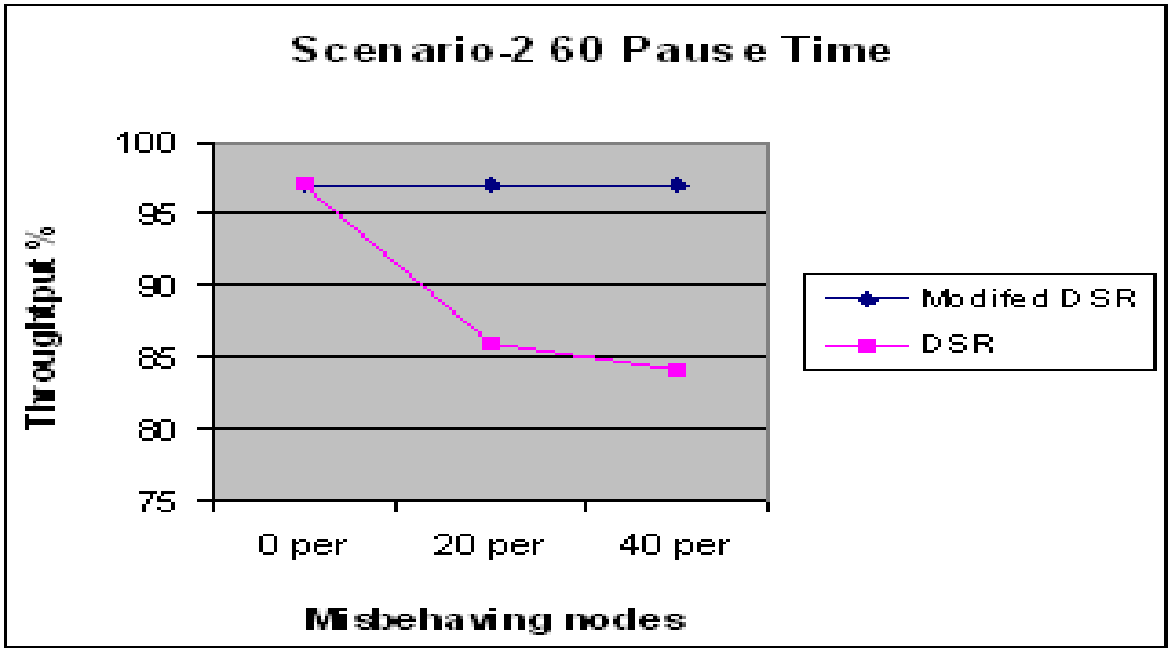

Figure 5.12: Throughput for scenario-2 pause time 60 seconds

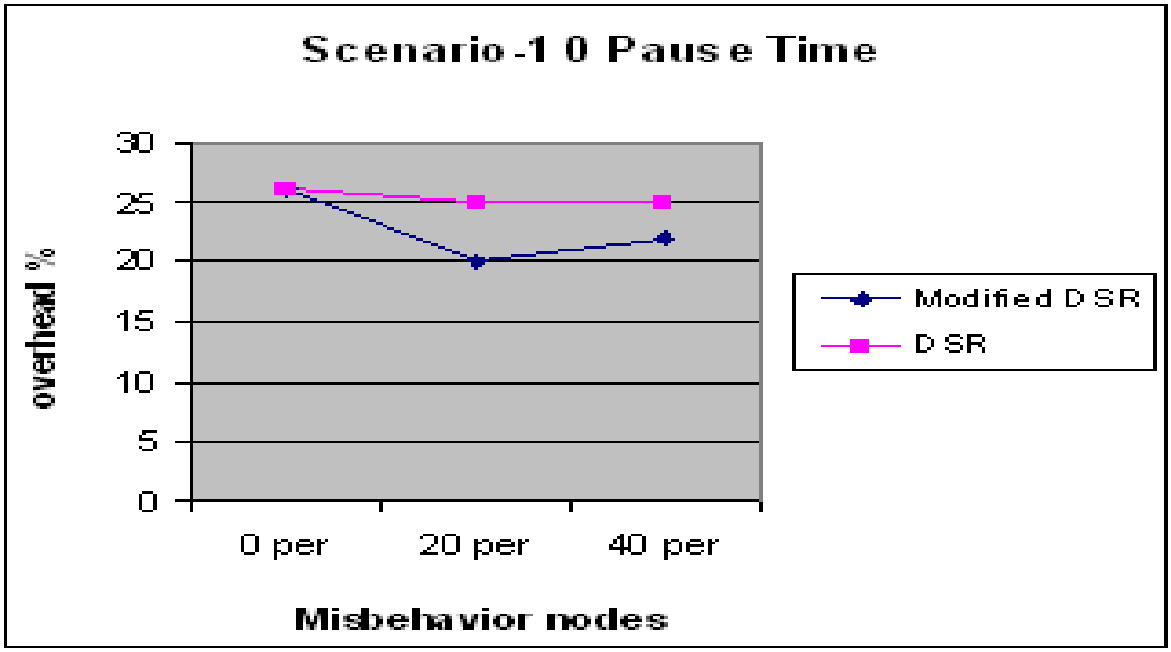

Figure 5.13: Overhead for scenario-1 no pause time

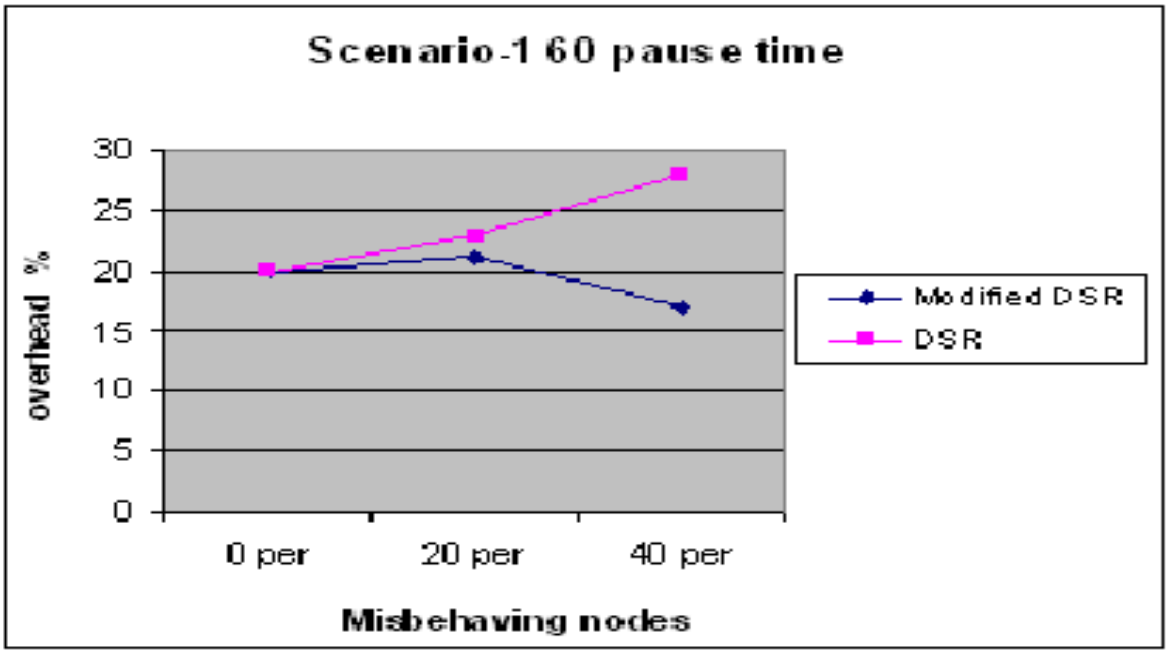

Figure 5.14: Overhead for scenario-1 pause time 60 seconds

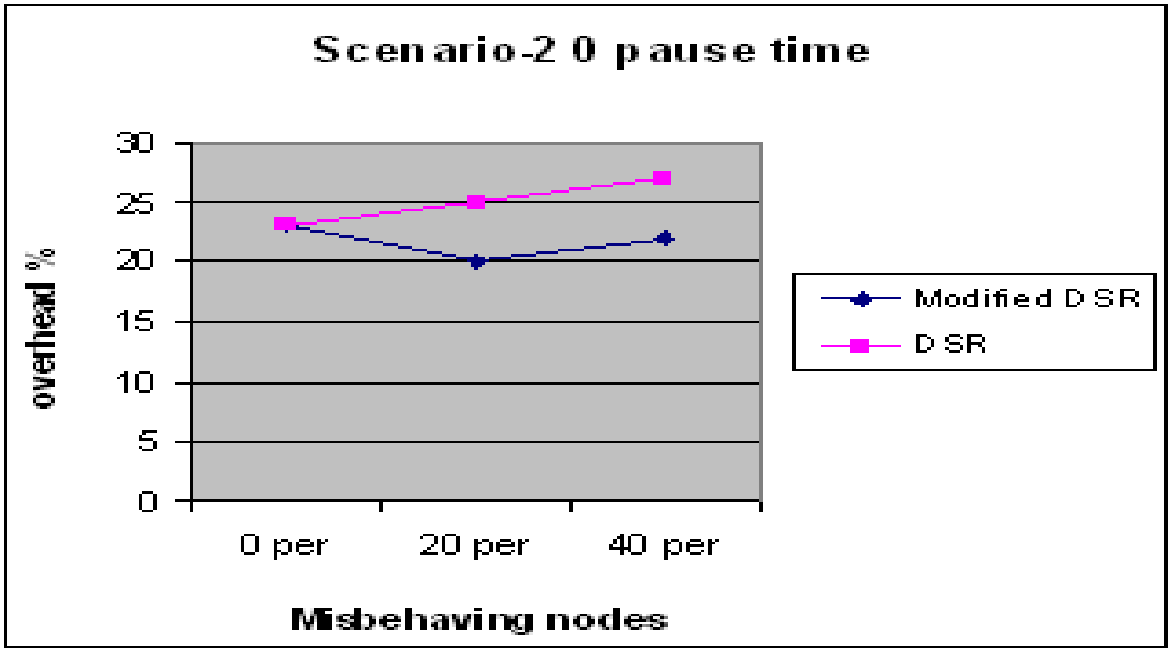

Figure 5.15: Overhead for Scenario-2 no pause time

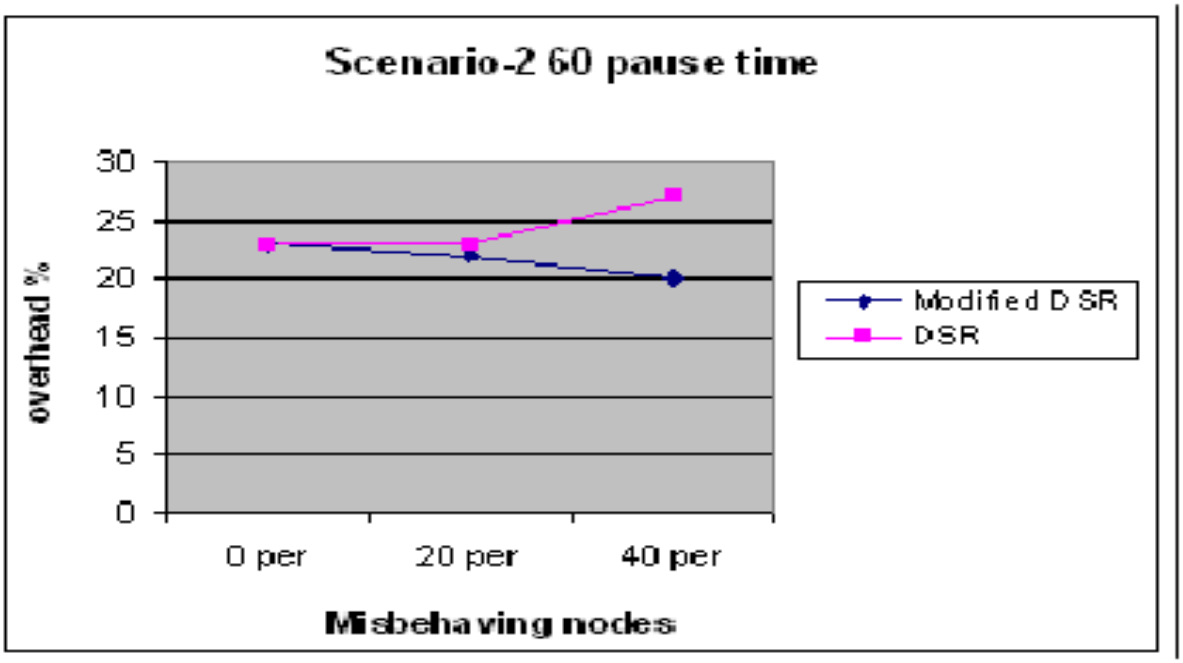

Figure 5.16: Overhead for scenario-2 pause time 60 seconds

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

After doing parametric study for different architecture of IDS system for adhoc wireless network we get the different mechanism to detect the malicious or selfish node. On that Watchdog is used in Stand Alone architecture for detect in malicious or selfish node. In Distributed and Cooperative architecture we have CONFIDANT Protocol, Probing algorithm mechanism used for detection. Stand Alone architecture Watchdog mechanism made for forwarded packet drop misbehavior done by node.

In the forwarded packet Drop misbehavior we have to maintain rating for every node and according to rating we can identify malicious or selfish node. After detection we have to select the path for send packet that not contain in malicious or selfish node that is done by the pathrater mechanism. Path is chosen by packet properly means not contain malicious or selfish node then network performance increase and also provided the reliability.

6.2 Future Scope

Future work is to implement wormhole attack with higher mobility. Also implement different other type attack done by misbehavior node and detect that node. After detecting node try to improve the performance of network.

Appendix A

Source Code

A.1 Generate Packet Drop Misbehavior In NS-2

```
set val(chan) Channel/WirelessChannel
    set val(prop) Propagation/TwoRayGround
    set val(netif) Phy/WirelessPhy
    set val(mac) Mac/80211
    set val(ifq) CMUPriQueue
    set val(ll) LL
    set val(ant) Antenna/OmniAntenna
    set val(x) 670 ;- X dimension of the topography
    set val(y) 670 ;- Y dimension of the topography
    set val(ifqlen) 50 ;- max packet in ifq
    set val(seed) 0.0
    set val(adhocRouting) DSR
    set val(nn) 50 ;- how many nodes are simulated
    set val(cp) "../mobility/scene/cbr-50-test"
    set val(sc) "../mobility/scene/scen-50-test"
    set val(stop) 400.0 ;- simulation time
```

set ns [new Simulator] ;-create simulator instance

set topo [new Topography]; -setup topography object

-create trace object for ns and nam

set tracefd [open no50.tr w]

set namtrace[open no50.nam w]

$*ns* trace-all $*tracefd*

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

$*topo* loadflatgrid $*val*(*x*) $*val*(*y*);-define topology

set god [create-god $*val*(*nn*)];-Create God, define how node should be created

set chan1 [new $*val*(*chan*)]

set chan2 [new $*val*(*chan*)]

-global node setting

$*ns* node-config -adhocRouting $*val*(adhocRouting)

-llType $*val*(ll)

-macType $*val*(mac)

-ifqType $*val*(ifq)

-ifqLen $*val*(ifqlen)

-antType $*val*(ant)

-propType $*val*(prop)

-phyType $*val*(netif)

-topoInstance $*topo*

-agentTrace OFF

-routerTrace ON

-macTrace OFF

-movementTrace OFF

-channel $*chan*1

for { set i 0 } { $*i* < $*val*(nn) } {incr i}
{
set node($*i*) [$*ns* node]
$*node*($*i*) random-motion 0 ;- disable random motion
}

puts “Loading connection pattern...”
source $*val*(*cp*)
puts “Loading scenario file...”
source $*val*(*sc*)

for {set i 0} $*i* < $*val*(nn)} {incr i}
{
$*ns* initialnodepos $*node*($*i*) 20
}

-Tell nodes when the simulation ends
for {set i 0} { i < $*val*(nn) } {incr i}
{
$*ns* at $*val*(stop) “$*node*($*i*) reset”;
}

proc finish {}
{
puts “hello” global ns tracefd namtrace
$*ns* flush-trace
lose $*tracefd*

close $*namtrace*

exit(0);

}

$*ns* at $*val*(stop).0002

puts " NS EXITING..." ;

$*ns* halt

puts $*tracefd* "M 0.0 nn $*val*(nn) x $*val*(x) y $*val*(y) rp $*val*(adhocRouting)"

puts $*tracefd* "M 0.0 sc $*val*(sc) cp $*val*(cp)seed $*val*(seed)"

puts $*tracefd* "M 0.0 prop $*val*(prop) ant $*val*(ant)"

puts "Starting Simulation..."

$*ns* at $*val*(stop).0002 "finish"

References

[1] S. Martiet, "Mitigating routing misbehavior in mobile ad hoc networks," *ACM Mobicom*, pp. 255–65, Auguest 2000.

[2] C. Murthy and B. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*. New Delhi: Prentice Hall India, second ed., 2005.

[3] H. yang, "Security in mobile ad hoc networks: Challenges and solutions," *IEEE Wireless Communications magazine*, October 2000.

[4] K. Inkinen, "New secure routing in ad hoc networks," tech. rep., Helsinki University of Technology. kai.inkinen@hut.fi.

[5] D. O. Patroklos G. Argyroudis, "Secure routing for mobile ad hoc networks,"

[6] E. J. Caballero, "Vulnerabilities of intrusion detection systems in mobile ad-hoc networks - the routing problem," erjica@gmail.com.

[7] B. A. Jean-Marie Orset and A. Cavalli, "An efsm-based intrusion detection system for ad hoc networks," *Institut National des Telecommunications GET-INT*. Evry, France fjean-marie.orset, baptiste.alcalde, ana.cavallig@int-evry.fr.

[8] S. S. Frank Kargl, Andreas Klenk and M. Weber, "Advanced detection of selfish or malicious nodes in ad hoc networks," Auguest 2004.

[9] R. D. Ningrinla Marchang, "Intrusion detection system for wireless networks," *Collaborative techniques for intrusion detection in mobile ad-hoc networks*, pp. 508–523, June 2008.

[10] Y. X. G. S. Bo Sun, Osborne L, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *Wireless Communications, IEEE*, vol. 14, pp. 56–63, October 2007.

[11] X. Wang, "Intrusion detection techniques in wireless ad hoc networks," *Computer Software and Applications Conference*, vol. 2, pp. 347–349, Sepetemer 2006. COMPSAC apos;06. 30th Annual International.

[12] T. W. Mike Just, Evangelos Kranakis, "Resisting malicious packet dropping in wireless ad hoc networks,"

[13] K. Fall and K. Varadhan, *The ns Manual (formerly ns Notes and Documentation)*. UC Berkeley, LBL, USC/ISI, and Xerox PARC. http://www.isi.edu/nsnam/ns/ns-documentation.html.

[14] *NS by Example*. http://nile.wpi.edu/NS.

[15] C. MonarchProject, "The cmu monarch projects wireless and mobilty extensions to ns.," *Collaborative techniques for intrusion detection in mobile ad-hoc networks*, October 1999. http://www.monarch.cs.cmu.edu/cmu-ns.html.

[16] *Ns mailing list*. ns-users@isi.edu.

Index