DECEPTION BASED HIERARCHICAL INTRUSION DETECTION SYSTEM FOR MOBILE AD-HOC NETWORKS

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of
INTEGRATED DUAL DEGREE
in
COMPUTER SCIENCE AND ENGINEERING
(With Specialization in Information Technology)

By

RADHIKA GOEL

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work is being presented in the dissertation work entitled " **Deception Based Hierarchical Intrusion Detection System for Mobile Ad-hoc Networks**" towards the partial fulfillment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** submitted to the **Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from May, 2010 to May, 2011 under the guidance and provision of **Prof. R. C. Joshi & Dr. Anjali Sardana, Department of Electronics and Computer Engineering, IIT Roorkee**.

I have not submitted the matter embodied in this dissertation work for the award of any other degree and diploma.

Date: 9th June, 2011

Place: IIT Roorkee

(Radhika Goel)

CERTIFICATE

This to certify that the work contained in the dissertation entitled "**Deception Based Hierarchical Intrusion Detection System for Mobile Ad-hoc Networks**" by Radhika Goel of Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology), has not been submitted elsewhere for a degree or diploma to the best of my knowledge.

Date: 9th June,2011

Place: IIT Roorkee

Prof. R. C. Joshi
Professor,
E&CE Department
IIT Roorkee, India

Anjali

Dr. Anjali Saradna
Assistant Professor,
E&CE Department
IIT Roorkee, India

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guides and mentors Prof. R. C. Joshi and Dr. Anjali Sardana, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for their trust in my work, their able guidance, regular source of encouragement and assistance throughout this dissertation work. I would state that the dissertation work would not have been in the present shape without their inspirational support and I consider myself fortunate to have done my dissertation under them.

I also extend my sincere thanks to Dr. S.N. Sinha, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities for the work.

I would like to thank all my friends, my sister Anuradha, members of Information Security Research Group (ISRG) and Mr. Raj Khapi, Information Security Lab In-charge, who supported and encouraged me to finish this work.

Finally, I would like to say that I am indebted to my parents for everything that they have given to me. I thank them for sacrifices they made so that I could grow up in a learning environment. They have always stood by me in everything I have done, providing constant support, encouragement, and love.

Radhika

RADHIKA GOEL

LIST OF FIGURES AND TABLES

ABSTRACT

The increasing usage of mobile computing devices has led to rapid emergence of Mobile Ad-Hoc Networks (MANETs) and potential threats to them. Even though a spectrum of Intrusion Detection Systems (IDS) exists for MANETS, the lack of knowledge about the exploitation methods used to compromise ad-hoc networks, is threatening the free and easy usage of ad-hoc networks.

Recently, the use of deceptive mechanisms for knowledge acquisition and intrusion detection has become very common in wired and infrastructure based wireless networks. They have traffic concentration and control points such as switches, routers, or gate ways where wired/wireless resources are deliberately deployed to lure and capture the attackers. MANET doesn't have such concentration or control points, therefore no proper architecture has been proposed till now for use of deceptive techniques in MANETs. However, the specific features of deception techniques like reliability, control over deployed resources and their luring capabilities can be used to overcome the limitations of earlier IDS used in general ad-hoc environment.

In this dissertation, we have combined detective techniques (Misuse and Anomaly based detection) and deceptive approaches to develop the first deception based hierarchical intrusion detection system to counter network based intrusions in ad-hoc environment. We propose the use of a deceptive, trusted and controlled mobile network in the vicinity of real production ad-hoc network as a trap to lure and deviate the attention of attackers. We have coined the term – HoneyMANET for this decoy network that monitors and test the maliciousness of foreign nodes crossing by. It is a made of trusted nodes, named as honeynodes, which move and generate data under the control of hidden hierarchical management and is open to foreign nodes for joining (with or without authentication, according to the security policies). Three different kinds of profiles – local, personal and global are generated for complete security of network from different kinds of old and new attacks. An unsupervised intrusion detection module is developed which uses the behavior of trusted honeynodes for reliable anomaly and misuse detection.

The tactic environment of HoneyMANET and the working of its four modules – deception, monitoring and logging, collection and integration, and intrusion detection modules have been simulated using Network Simulator. Simulations are done to find different design parameters of honeyMANET - free movement zones, number, speed and data generation rate of honeynodes, to make a robust IDS. Different kinds of localized and globally distributed attacks, with varying rate and number of attackers, are launched to test the robustness of proposed model in different attack scenarios. Simulation results show that the attack detection efficiency of HoneyMANET is high and mostly remains at value 1, independent of type of attack or number of attackers in the network. The false alarm rate is also low (mostly remaining at value 0). This is a great achievement as compared to IDS of general ad-hoc network where detection rate decreases as number of attacker increases in the network. HoneyMANET's use in evaluation of impact of different attacks on networks is also shown. Route Request flooding attack is shown to affect the network more drastically than packet dropping attacks. Simple packet drop attack is less severe than black hole attack, but as the number of attackers' increases, its effect is almost same as that of black hole attack. It has been shown that HoneyMANET gives us both localized and global overview of activities taking place in the network and is a reliable, robust and efficient intrusion detection system.

CONTENTS

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Introduction

Mobile Ad-Hoc Networks – MANETs are networks whose nodes are free to move randomly while being able to communicate with each other without the help of an existing network infrastructure. It is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies. Nodes of MANET behave as routers and take part in route discovery and maintenance processes to establish reliable routes for each other.

In recent years, the use of mobile computing devices, which mainly include laptops, personal digital assistants (PDAs) and handheld digital devices, has increased exponentially. This has lead to emergence of ubiquitous computing in which numerous users utilize several electronic platforms at the same time to access information. MANETs provide the interconnecting network between these platforms. The unique characteristics of MANETS like decentralized nature, scalable setup and dynamically changing topology make ad-hoc networks ideal for a variety of applications ranging from front-line zones (military, industrial and natural) to data collection(machinery analysis, bio-sensing). With the increasing usage of MANET in commercial/business applications its security has become a great challenge [1] [2].

In Manets, as all the network nodes act as routers and take part in route discovery and maintenance process, any malicious node can cause deliberate harm to ad-hoc network. Dynamism of ad-hoc networks and lack of central management facility make it extremely difficult to detect the attacks and the attackers. MANETs suffer from different kinds of attacks like passive eavesdropping, spoofing, information leakage, message distortion and DoS attacks [3]. Some attacks specific to MANETs are Gray Hole attack, Black Hole attack, Impersonation, Modification attack, attacks against routing tables and selfish misbehaving.

1.2 Motivation

Many researchers have proposed various preventive and detective mechanisms or intrusion detection systems (IDS) for securing MANETs [4]. However, because of limitations of dynamic environment of mobile ad-hoc network - Unreliable wireless links, limited bandwidth and power, lack of any efficient trust model to test authenticity of node participating in decision making process, lack of secure boundaries and physical protection of mobile nodes, dynamically changing topology, the absence of a certification authority, and the lack of a centralized monitoring or management point - their effectiveness is limited:

1. Reliability: IDS solutions require production nodes' participation in traffic sniffing, information sharing and decision making processes. These nodes can be malicious, selfish or benign. Because of absence of any line of defense, which distinguishes nodes as trusted and non-trusted, malicious and selfish nodes can poison the whole data [1] and thus, can make the whole decision making process ineffective.

2. Efficiency: Since most of the IDS are cooperative in nature, the increase in ratio of bad nodes in the network decreases the ratio of trusted decision making nodes in the network, decreasing the attack detection efficiency of IDS.

3. Robustness: It becomes extremely difficult by individual nodes running IDS to monitor the traffic in a highly dynamic and large scale ad hoc network [2] without central management which limits the detection of distributed attacks by them.

4. Limited Resources: The limited battery power of mobile nodes limits their intelligence sophistication and also causes some nodes to behave in a selfish manner during the cooperative intrusion detection processes.

Recently, the use of deceptive mechanisms [5] for information security and knowledge acquisition about different attacks and attack mechanisms has become very common in wired and infrastructure based wireless networks [5]. Wired and infrastructure based wireless networks have traffic concentration and control points such as switches, routers, or gate ways where wired/wireless resources are deliberately deployed to lure and capture the attackers. MANET doesn't have such concentration and control points, therefore no proper architecture has been proposed till now for use of deceptive techniques in MANETs. Seeing the increasing success of deceptive mechanisms in securing different network domains - wired and infrastructure based wireless networks, in this work, we propose the first deception based hierarchical IDS for raising the security bar of ad-hoc environment.

1.3 Problem Statement

The objective of this work is to design and develop an efficient, reliable and robust Deception based Hierarchical Intrusion Detection System to detect network based intrusions in mobile ad-hoc environment.

The problem can be further sub-divided into following sub-problems:

1. Designing deceptive IDS for Mobile ad-hoc environment.
2. Evaluating the impact of different attacks on MANET using the proposed deception based IDS.
3. Assessing the reliability, efficiency and robustness of deceptive measures in detecting new and old network layer attacks.

1.4 Organization of Dissertation

This report comprises of seven chapters including this chapter that introduces the topic and states the problem. The rest of the dissertation report is organized as follows:

Chapter 2 provides a detailed literature review on the existing security solutions for the mobile ad hoc networks and analyzes the research gaps of them. The other topics include characteristics and vulnerabilities of ad-hoc networks, classification of security attacks prevalent in MANET environment, security issues in MANETs and generic IDS models and architectures.

Chapter 3 provides a detailed description of the proposed scheme for detecting intrusions in ad-hoc environment using deception techniques. The different IDS modules of proposed hierarchical model are discussed along with the deployment scenarios. The advantages of using the proposed deception based IDS over earlier IDS used in general ad-hoc environment are given.

Chapter 4 gives the description of simulations done to find the different design parameters used to make a robust IDS. Simulation scenarios used to test the robustness and efficiency of proposed scheme are given. The evaluation criterion used to measure efficiency of proposed model and to evaluate the impact of different attacks on network is also given.

Chapter 5 includes results and discussion. The attack detection efficiency of the proposed model is measured and reasoned out under varying attack scenarios. Finally, a comparison between reliability and efficiency of proposed approach with earlier IDS used in general ad-hoc networks is given.

Chapter 6 provides as analysis on how impact of different attack on network can be evaluated using the proposed technique.

Chapter 7 concludes the work giving the important features and advantages of the proposed scheme. The directions for future work are also given.

CHAPTER 2
LITERATURE REVIEW

2.1 Mobile Ad- Hoc Networks

A Mobile Ad hoc Network (MANET) is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies to communicate with each other without the help of an existing network infrastructure. In the mobile ad hoc network, nodes can directly communicate with all the other nodes within their radio ranges or intermediate nodes can act as routers.

2.1.1 Characteristics

The mobile ad hoc network has the following typical features, because of which they are more vulnerable to malicious behaviors than the traditional wired networks:

1. **Constantly changing topology**: Due to the continuous motion of nodes, the topology of the MANET changes constantly. Nodes continuously move into and out of the radio range of other nodes continuously changing the routing information.
2. **Energy-constrained operation**: Some or all of the nodes in a MANET may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation. Such optimizations may lead to selfish behaviors by nodes. Selfish and failed nodes (battery failed cases) affect the performance of network.
3. **Unreliability of wireless links between nodes:** Because of the limited energy supply for the wireless nodes and the mobility of the nodes, the wireless links between mobile nodes in the ad hoc network are not consistent for the communication participants.
4. **Limited physical security**: Mobile wireless networks are generally more prone to physical security threats than are wired networks. MANET has more possibility of eavesdropping, spoofing and denial-of-service attacks.

2.1.2 Mobility Patterns in MANETs

There can be various real world situations in which MANETS could be used. Here we have explained node's mobility in four different scenarios - MANETs in general, in battlefields, on Freeways and in urban street areas.

In different kinds of MANETs, network nodes can have different spatial and temporal dependencies. Spatial dependency exists when two nodes are dependent in their motion. If two nodes are moving in same direction then they have high spatial dependency. Temporal dependency exists when current velocity (magnitude and direction) of a node is related to its previous velocity. Nodes having same velocity have high temporal dependency.

1. **MANETs in general:** In general, network nodes can have no spatial and temporal dependencies. They can move randomly in all different directions, with all different allowable velocities and with different pause times in between. Figures 2.1 illustrate the example of a topography showing the movement of nodes in random fashion..
2. **MANETS in military battlefields:** In military battlefields, mobile nodes usually exist in groups. Here each group has a group leader (at logical center of group) that determines the group's motion behavior. Initially, each member of the group is uniformly distributed in the neighborhood of the group leader. Subsequently, at each instant, every node has speed and direction that is derived by randomly deviating from that of the group leader. Given below, in figure 2.1, is an example topography showing the movement of nodes in military battlefield.
3. **MANETS on Freeways:** MANETs are used in exchanging traffic status or tracking a vehicle on freeways. On freeways, mobile nodes are geographically restricted to their lanes and their velocities are temporally dependent on their previous velocities. Therefore, in such MANETs, high spatial and temporal dependencies exist. Given below, in figure 2.1, is an example topography showing the movement of nodes on a freeway.
4. **Manets in city streets:** The movement pattern of mobile nodes on streets in an urban area moving along the grid of horizontal and vertical streets is shown in Figure 2.1. At an intersection of a horizontal and a vertical street, the mobile node can turn left, right or go straight with certain probability. Nodes in such environment have spatial and temporal dependencies.

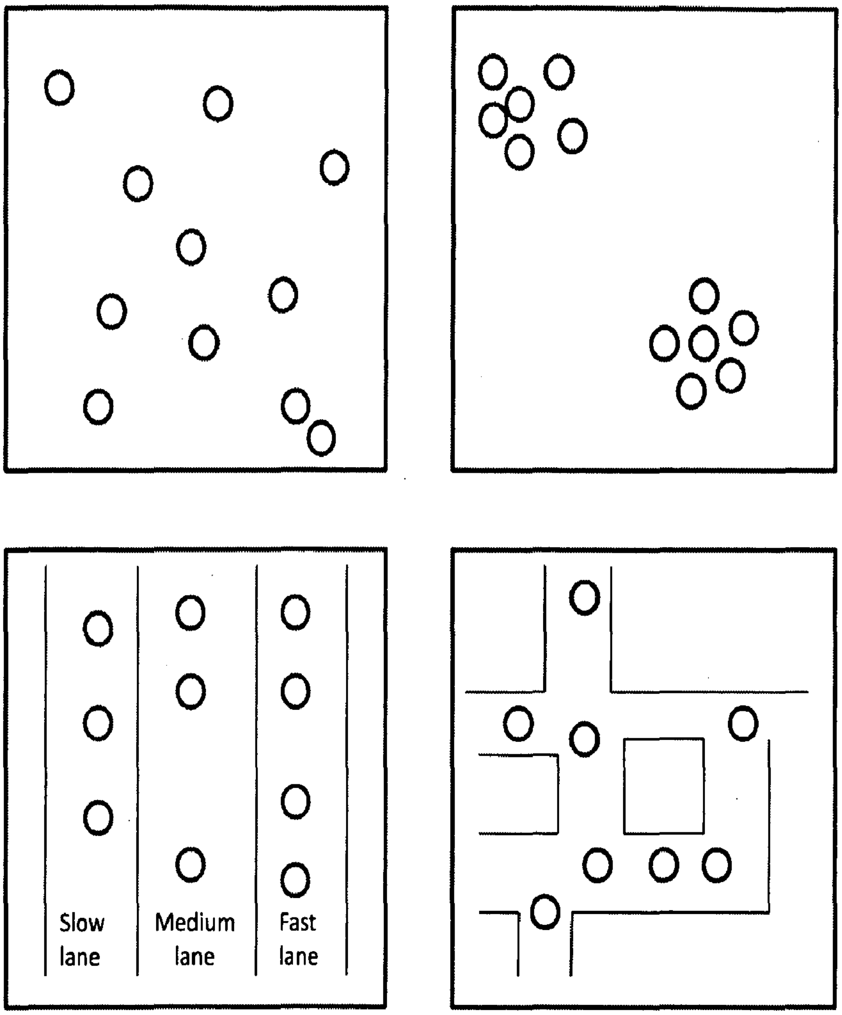

Figure 2.1 Example MANET Topologies: in general (top left), in battlefield (top right), on freeways (bottom left) and on streets (bottom right)

2.1.3 Service provisioning

Service provisioning in MANET environment is different from that over fixed network. In MANETs service provisioning is opportunistic. There are no fixed, well-known service providers. Any node can be a service provider for its own benefit and for as long as it participates in the MANET.

Two type of service provisioning application models are possible in MANET environment - Client-Server or Client-Client type of application model.

1. **Client-Server type of applications**: Mobile servers being nodes in the MANET offer their services to other MANET nodes, acting as clients, for profit or for non-profit reasons. For example, in a Vehicular Ad Hoc Network (VANET), a car can offer

navigation service to other cars in the network. A car using this service will be paying for the service as long as it can reach the serving vehicle via the multi-hop network (e.g. cents/packet). In such commercial-purpose Mobile Ad Hoc networks, both the mobile clients and mobile servers may act selfishly or maliciously to maximize their own profit.

2. **Client-Client type of applications:** Any node in MANET makes a connection with any other node in network providing services to each other. Client-client type of applications in MANET can include industrial and commercial applications involving cooperative mobile data exchange.

2.2 Attacks in MANETs

2.2.1 Vulnerability of MANET Protocol Stack

There are differences between the protocol stack of MANET, TCP/IP and OSI model. Figure 2.2 shows the protocol stacks for MANET, TCP/IP and OSI model. The MANET protocol stack consists of five layers: physical layer, data link layer, network layer, transport layer and application layer. The network layer is divided into two parts: network and ad hoc routing. The protocol used in the network part is generally Internet Protocol (IP). For ad hoc routing, mobile nodes use various proactive and reactive ad hoc routing protocols to route packets [6].

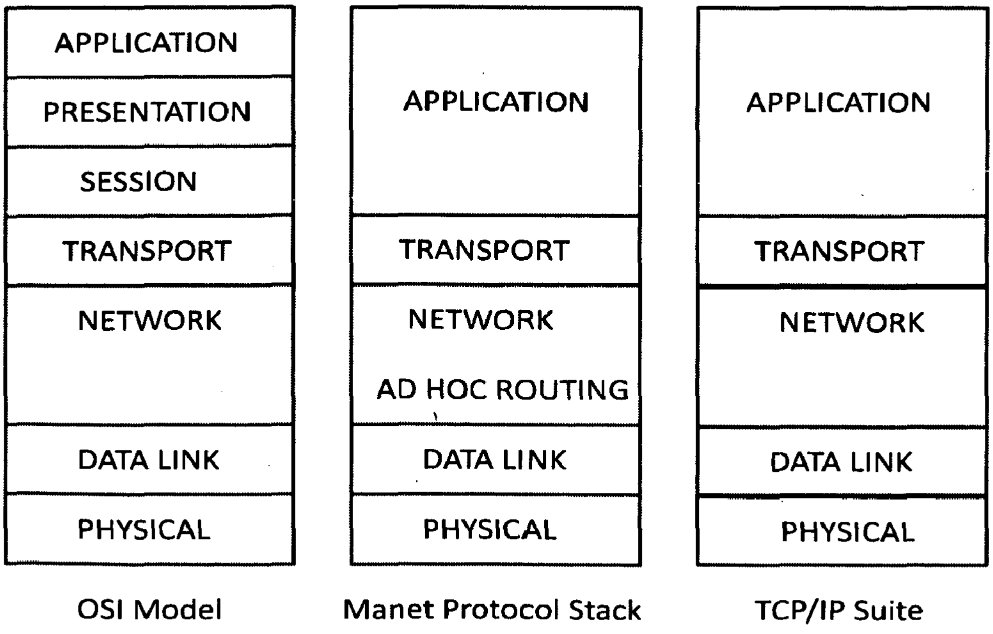

Figure 2.2 Protocol Stacks

Table 2.1 shows various security attacks on MANET Protocol Stack. There also exist some multi-layer attacks in MANETs such as Denial of service (DoS) attack. Denial of service (DoS) is an attack that can be launched from several layers in which legitimate users are denied services. An attacker can employ signal jamming at the physical layer, which disrupts normal communications. At the link layer, malicious nodes can occupy channels through the capture effect, which takes advantage of the binary exponential scheme in MAC protocols and prevents other nodes from channel access. At the network layer, the routing process can be interrupted through routing control packet modification, selective dropping, table overflow, or poisoning. At the transport and application layers, SYN flooding, session hijacking, and malicious programs can cause DoS attacks.

Table 2.1 Security Attacks on MANET Protocol Stack

Layer	Attacks Examples
Application layer	Repudiation and data corruption
Transport layer	Session Hijacking and SYN Flooding
Network layer	Black Hole, Byzantine, Flooding attack, Grey Hole and Sleep Deprivation attack.
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11) and WEP weakness
Physical layer	Jamming, interceptions and eavesdropping

2.2.2 Classification of Security attacks in MANETs

Attacks on ad hoc wireless networks can be classified as passive and active attacks, depending on whether the normal operations of the network are disrupted or not. Both passive and active attacks can be made on any layer of the network protocol stack.

1. **Passive attacks:** A passive attack does not disrupt the normal operations of the network. The attacker snoops the data exchanged in the network without altering it violating the .requirement of confidentiality. Detection of passive attack is very difficult since the

operations of the network itself don't get affected. Preventive cryptographic techniques are used to secure the confidentiality of data exchanged.

2. **Active attacks**: An active attack attempts to alter or destroy the data being exchanged in the network, disrupting the normal functioning of the network. Active attack involves information interruption, modification, or fabrication. Active attacks can be internal or external, according to the domain of the attack. External attacks are carried out by nodes that do not belong to the domain of the network. Internal attacks are from compromised nodes which are actually part of the network. Internal attacks are more severe when compared with outside attacks since the insider knows valuable and secret information and possesses privileged access rights.

Table 2.2 shows the general taxonomy of security attacks [1][2][3] against MANET. Examples of passive attacks are eavesdropping, traffic analysis, and traffic monitoring. Examples of active attacks include jamming, impersonating, modification, denial of service (DoS), and message replay.

Table 2.2 Security Attacks Classification

Passive Attacks	Eavesdropping, traffic analysis, monitoring
Active Attacks	Jamming, spoofing, modification, replaying, DoS

2.2.3 Network Layer Attacks

Currently routing security is one of the hottest research areas in MANET [3]. In routing mechanisms of ad hoc networks, three layers namely physical, MAC and network layers play major roles, out of which the foremost concerned issue is to protect the route and data forwarding operations of the network layer. Without any appropriate security solution, the malicious nodes in the network can readily act as routers. This disturbs the network operation from correct delivering of the packets. Therefore, this work focuses on mainly network layer attacks. The variety of attacks in the network layer includes not forwarding the packets or adding and modifying some parameters of routing messages; such as sequence number and hop count.

Some examples of attacks on network layer are Black Hole, Byzantine, Flooding, Grey Hole and Sleep Deprivation attack.

1. **Byzantine attack**: In this attack, a compromised intermediate node works alone, or a set of compromised intermediate nodes work in collusion and carry out the attacks. They can generate false alarm messages to hinder the working of defense mechanism used, create routing loops, forward packets through non-optimal paths, or selectively drop packets, which result in disruption or degradation of the routing services. These kinds of failures are not easy to detect, since the network seems to be operating very normally in the view of the user.
2. **Route Request Flooding attack (RREQ Flooding attack):** In this attack, a malicious node deliberately tries to consume the resources (e.g. battery power, bandwidth) of other nodes in the network by flooding the network with unnecessary RREQ packets.
3. **Simple Packet Dropping:** In this attack a malicious node drops all the data packet it receives from other nodes for forwarding. A node may drop packets due to several reasons: due to network congestion, due to lack of energy resources or due to malicious nature. When a node does not participate in packet routing mechanism (for packets of other nodes) because of its own resource constraints while consuming network resources, it is seen as selfish behavior. It is called an attack, when the node drops the data packets for malicious intentions.
4. **Black hole attack:** It is a severe form of packet dropping attack in which attacker attracts all the network traffic towards it by falsely advertising itself as having a good path (e.g., shortest path or most stable path) to the destination node during the path finding process. The intension of the malicious nodes can be to hinder the path finding process or to interrupt all the data packets being sent to the concerned destination node.
5. **Gray Hole:** The gray hole attack is a special case of a black hole attack in which instead of dropping all the intercepted packets, an attacker drops packets with a certain probability. This attack is more difficult to detect than the black hole attack. A gray hole may exhibit its malicious behavior in different ways. It may drop packets coming from (or destined to) certain specific node(s) in the network while forwarding all the packets for other nodes. Another type of gray hole node may behave maliciously for some time duration by dropping packets but may switch to normal behavior later. A gray hole may

also exhibit a behavior which is a combination of the above two, thereby making its detection even more difficult.

6. **Sleep Deprivation (SD) attack by flooding useless data:** An attacker floods a specific route by sending data packets through it at high rate or directing all the traffic passing coming to it towards a specific route, consuming resources of nodes in the path. A vicious node can also establish connections with some other nodes in ad hoc networks, then it sends a lot of useless data packets to the destination node consuming all its resources and time. It is a kind of denial of service attack in which an attacker interacts with the node in a manner that appears to be legitimate; but where the purpose of interaction is to keep the victim node out of its power conserving sleep mode.

2.3 Security in General

2.3.1 Security attributes

Security is the combination of processes, procedures, and systems used to ensure confidentiality, authentication, integrity, availability, access control, and non-repudiation.

1. **Confidentiality** is to keep the information sent unreadable to unauthorized users or nodes. MANET uses an open medium, so usually all nodes within the direct transmission range can obtain the data. One way to keep information confidential is to encrypt the data and another technique is to use directional antennas.

2. **Authentication** is to be able to identify a node or a user and to be able to prevent impersonation. In wired networks and infrastructure-based wireless networks, it is possible to implement a central authority at a point such as a router, base station, or access point. But there is no central authority in MANET, and it is much more difficult to authenticate an entity.

3. **Integrity** is to be able to keep the message sent from being illegally altered or destroyed in the transmission. When the data is sent through the wireless medium, the data can be modified or deleted by malicious attackers. The malicious attackers can also resend it, which is called a replay attack.

4. **Non-repudiation** is related to a fact that if an entity sends a message, the entity cannot deny that the message was sent by it. By producing a signature for the message, the entity cannot later deny the message. In public key cryptography, a node A signs the message using its private key. All other nodes can verify the signed message by using A's public key and A cannot deny that its signature is attached to the message.

5. **Availability** is to keep the network service or resources available to legitimate users. It ensures the survivability of the network despite malicious incidents.

6. **Access control** is to prevent unauthorized use of network services and system resources. Access control is tied to authentication attributes. In general, access control is the most commonly thought of service in both network communications and individual computer systems.

2.3.2 Security Mechanisms

Intrusion of a computing system or a network means an attempt to break into or misuse it. The act of intrusion comprises a set of actions that attempt to compromise the integrity, confidentiality or availability of the victim system. Various preventive, detective and deceptive measures are used in order to secure our computers or networks against intrusions.

1. **Preventive mechanisms:** The conventional approaches such as authentication, access control, encryption, and digital signature provide the first line of defense. Cryptographic primitives such as hash functions (message digests) can be used to enhance data integrity in transmission. Threshold cryptography is used to hide data by dividing it into a number of shares. Digital signatures are used to achieve data integrity and authentication services. The protection of the sensitive data on a physical device is enforced by security modules, such as tokens or a smart card that is accessible through PIN, passphrases, or biometrics.
2. **Detective mechanism:** Detective security measures act as the second wall of defense. Intrusion detection systems help to detect attacks in action and take appropriate actions for recovery before serious damage occurs. The first intrusion detection model was built by Dorothy E. Denning [7]. In 1987, Dinning proposed a model of a real-time intrusion-detection expert system which was able to detect break-ins, penetrations, and other forms

of computer abuses. The model is based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. The model includes profiles representing the normal behavior of subjects using statistical models and rules for acquiring knowledge about this behavior from audit records for detecting anomalous behavior.

3. **Deceptive Mechanisms:** Deception includes taking actions to deliberately mislead hackers, causing them to take (or not take) specific actions that aid computer security. Hackers rely heavily upon network data, and there is great potential for using network data to deceive them. Lance Spitzner [8] developed the concept of using honeypots for deception in wired domain. Later, wireless honeypots were also designed for use in infrastructure/access points based wireless networks. Fake wireless resources are deployed to lure the attacker and to gather knowledge about real statistics of attacks like layer being attacked, attack techniques used, the vulnerabilities of attacked layer, weakness in current encryption and cryptographic technique used, frequency of these attacks, the attacker's skill level, his goals and methods. They also act as intrusion detection systems and consume attacker's time on fake resources.

2.4 Intrusion Detection as Active Network Defense in MANETs

2.4.1 Intrusion Detection System

An Intrusion Detection System (IDS) is a defense system that implements the functions of intrusion detection, intruder detection and tries to mitigate those attacks. An IDS achieves these goals by continuously monitoring the network (surveillance), collecting and analyzing audit data from variety of sources, identifying unusual or misuse activities, and initializing appropriate responses for detected activities. In general, an IDS consists of 4 subsystems (Figure 2.3):

1. **Monitoring and Logging Module**: Different parts of the network/system are monitored and audits are created. The choice of audit data sources depends on the type of intrusions, the type of detection mechanism used, and the type of the system in which the detection is being performed. For Host based intrusion detection systems - system calls, syslog logs generated by the Operating System (OS) kernel and other audit information (e.g., CPU

usage, I/O activity) are logged. To monitor network based intrusions- traffic flow features are logged.

2. **Data Collection and Integration Module**: The audit data collection module in the audit subsystem takes care of logged information, collects audit data from different sources, and transforms them into a common format before sending it to the detection engine. The audit storage/pre-process module can also be used to deposit audit data for later reference or offline analysis.
3. **Intrusion detection Module:** The detection engine is a place where the collected audit data is analyzed and suspicious activities are identified. If different detection engines are used, it is necessary to correlate results from different engines. Information about events that are classified as intrusive or anomalous by the detection engine is then sent to the response module. Two types of widely used detection methodologies are: anomaly detection and misuse based detection.
4. **Response Module:** Based on the knowledge of the type of intrusion, network protocols, applications in use, confidence in the evidence and pre-programmed rules in the response policy database, it is decided how to automatically respond to different events. The automated response mechanisms range from passive notification to active intruder traceback and filtering.

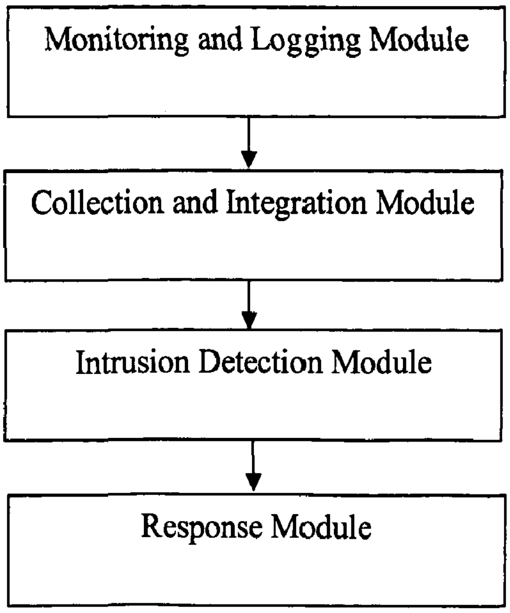
Monitoring and Logging Module
Collection and Integration Module
Intrusion Detection Module
Response Module

Figure 2.3 Intrusion Detection System' Modules

2.4.2 Classification of IDS

IDS can be classified on basis of type of intrusions (network-based or host-based IDS), on basis of detection mythology (Anomaly or Misuse based IDS), architecture used (stand alone, distributed and cooperative, hierarchical and mobile agents based IDS) and timeliness of the audit data being gathered and processed (offline or online IDS).

2.4.2.1 Host-based and Network-based IDS

Host-based systems (HIDS): They are concerned with what is happening on each individual host. They are able to detect actions such as repeated failed access attempts or changes to critical system files, and normally operate by accessing log files or monitoring real-time system usage. In order for a HIDS to function, clients have to be installed on every host in the network. These clients reside on the hosts as processes and perform analysis on the audit data gathered locally, at the expense of the already limited resources of the hosts.

Network Based IDS (NIDS): Network based intrusion detection system looks for maliciousness or suspicious traffic patterns in network traffic. Network adapter card running in promiscuous mode allows monitoring and analysis of individual raw network packets flowing through a network in real-time. They are able to look at the payload within a packet, to see which particular host application is being accessed and to raise alerts when attacker tries to exploit a bug in such code. NIDS are typically host-independent but can also be a software package installed on dedicated workstation.

2.4.2.2 Anomaly detection and Misuse detection

Anomaly detection bases its idea on statistical behavior modeling. Anomaly detectors look for behavior that deviates from normal behavior [9]. A typical anomaly detection system takes in audit data for analysis. The audit data is transformed to a format statistically comparable to the profile of a normal user. The user's profile is generated dynamically by the system (usually using a baseline rule laid by the system administrator) initially and subsequently updated based on the user's usage. Thresholds are normally always associated to all the profiles. If any comparison between the audit data and the user's profile resulted in deviation crossing a threshold set, an alarm of intrusion is declared. This type of detection systems is well suited to detect unknown or previously not encountered attacks.

Misuse detection bases its idea on precedence and rules. Misuse detectors look for behavior that matches a known attack scenario [9]. A typical misuse detection system takes in audit data for analysis and compares the data to large databases of attack signatures. The attack signatures are normally specified as rules with respect to timing information and are also referred to as known attack patterns. If any comparison between the audit data and the known attack patterns described resulted in a match, an alarm of intrusion is sounded. This type of detection systems is useful in networks with highly dynamic behavioral patterns. However, it has limited usage and cannot identify new attacks because of limited rul e base.

2.4.2.3 Online Detection and Offline Detection

When a system is performing intrusion detection in online mode, the audit data is processed real-time continuously. A host-based system will gather information about a host as long as the host is connected to the network. A network-based system will monitor the network traffic of the hosts throughout the time they are connected. Any intrusion detected is immediately notified to the host.

When a system is performing intrusion detection in offline mode, the audit data is not processed real-time. A host-based system will gather information about a host even if it is not connected to the network. Even if the host is connected, detection is done as scheduled by the system. A network-based system will monitor the network traffic of the hosts periodically, typically for an entire hour or day. Any intrusion detected is still immediately notified to the host but a delay is expected. A typical technique of an offline intrusion detection system is data mining.

2.4.2.4 Classification based on architecture

The various IDS based solutions proposed till now are categorized in mainly the following four classes [4] - Stand-alone IDSs, Distributed and Cooperative IDSs, Hierarchical IDSs and Mobile Agent based IDSs.

1. **Stand-alone Network Architecture:** IDS is executed independently for each node, and the necessary decision taken for that node is based on the data collected locally. There is no interaction among network nodes in such systems and therefore each node

has no knowledge of the position of other nodes in that network and no alert information crosses the network. Due to the fact that exclusive node information is not enough to detect intrusions, such architecture is not appreciated much for intrusion detection by present day IDS. While the effectiveness of this solution is limited, this architecture may be suitable in an environment where not all nodes are capable of running an IDS or have an IDS installed.

2. **Distributed and Collaborative Architecture:** In this architecture, every node in the mobile ad hoc network captures the data around it and detects intrusions locally and independently. The detection is done by the built-in IDS agent at each node. Neighboring nodes also share their investigation results with each other when detected anomaly is difficult to be evident of an intrusion by a single node. However, the cooperative and distributive IDS architectures are generally susceptible to attacks from Byzantine nodes, which could independently make false claims of detecting an attack from a correct node with strong evidence, thus making it difficult to derive a distributed consensus.

3. **Hierarchical IDS** are cluster-based intrusion detection systems for ad hoc networks. It is an extended version of the distributed and collaborative IDS architecture. This architecture is used in multi-layered network infrastructures where the network is divided into clusters. In this architecture, MANETs are organized into a number of clusters and one node per cluster is selected as cluster head for monitoring and analysis purposes. These cluster heads, in some sense, act as control points which are similar to switches, routers, or gate ways in wired or infrastructure based wireless networks. The main advantage of hierarchical architecture is that overall packet-monitoring task is limited to a small subset of nodes, thus conserving power and processing capabilities for many nodes in the ad hoc network. The disadvantage is that any Byzantine cluster head nodes can potentially reroute, modify, or drop packets transmitted by cluster member nodes, as well as any packets routed through the cluster head on the virtual backbone.

4. **Mobile agent based IDS architecture** uses mobile agents to perform specific task on nodes on behalf of their parent nodes. In distributed and cooperative architecture or in hierarchical architecture, mobile agents are used for intrusion detection tasks. There are

several advantages of using mobile agents [10] for intrusion detection like distribution of the intrusion detection tasks and continuous working even in absence of parent node.

2.4.3 Evolution of Intrusion Detection in MANETs

Intrusion detection in MANET is addressed by various researchers and has been a major research area. The detective mechanisms used in ad-hoc networks include incentive based systems that provide incentives to nodes to comply with protocol rules, secure routing techniques which make attack specific changes in routing protocols, watchdog technique based systems which monitors the behavior of all the surrounding nodes, reputation based systems that maintains a trust and reputation table for all nodes in a sub-network and systems to detect faulty and/or misbehaving nodes, report them and exclude them from the network.

In 2000, S. Marti, T. J. Giuli, K. Lai and M. Baker [11] proposed the "watchdog and pathrater" scheme that is used to detect & mitigate the effect of nodes that do not forward packets. These two techniques were added to the standard routing protocol DSR in ad hoc networks. Watchdog determines misbehavior by copying packets to be forwarded into a buffer and monitoring the behavior of the adjacent node to these packets. Watchdog promiscuously snoops to decide if the adjacent node forwards the packets without modifications or not. If the packets that are snooped match with the observing node's buffer, then they are discarded; whereas packets that stay in the buffer beyond a timeout period without any successful match are flagged as having been dropped or modified. The node responsible for forwarding the packet is then noted as being suspicious. If the number of violations becomes greater than a certain predetermined threshold, the violating node is marked as being malicious. Information about malicious nodes is passed to the Pathrater component for inclusion in path rating evaluation. Pathrater on an individual node works to rate all of the known nodes in a particular network with respect to their reliabilities. Ratings are made and updated from a particular node's perspective. Nodes start with a neutral rating that is modified over time based on observed reliable or unreliable behavior during packet routing. Nodes that are observed by watchdog to have misbehaved are given an immediate rating of -100. This way Pathrater helps in finding the possible routes excluding the misbehaving nodes.

Lee, and Zhang [12][13] proposed a distributed and cooperative "intrusion detection (ID) and response system". In this proposed architecture model, each node was responsible for detecting signs of intrusion locally and independently, but neighboring nodes can collaboratively investigate in a broader range. Individual IDS agents are placed on each and every node. Each the IDS agent runs independently and monitors local activities (user and systems activities, and communication activities within the radio range). The agent detects intrusion from local traces and initiates response. If anomaly is detected in the local data, or if the evidence is inconclusive, then a majority voting is taken from neighboring nodes for intrusion detection. These individual IDS agents collectively form the IDS system to defend the wireless ad-hoc network.

The CONFIDANT (Cooperation of Nodes, Fairness In Dynamic Ad-hoc Networks) system proposed in 2002 [14], is based on direct observations and on second-hand information from other nodes, and is updated according to a Bayesian estimation. CONFIDANT consists of Monitoring System, Reputation System, Trust Manager and Path Manager. Their tasks are divided into two sections: the process to handle its own observations and the process to handle reports from trusted nodes. It is based on distributed and cooperative architecture. Its nodes cooperate and share alarm messages with other nodes in the wireless ad-hoc network that are in a node's friend list. If a node is observed to behave in a cooperative fashion, then positive reputation is assigned to it, otherwise a negative reputation is assigned to it. The alarm messages are evaluated based on their trustworthiness using Bayesian estimations.

Michiardi and Molva, in [15], proposed the CORE system (A Collaborative Reputation Mechanism to enforce node cooperation in MANETs), which uses game theoretic analysis to model reputation. Members that have a good reputation, because they helpfully contribute to the community life, can use the resources, while members with a bad reputation, because they refuse to cooperate, are gradually excluded from the community. In CORE [15], the term "subjective reputation" is used to represent the reputation calculated from direct observations' using a weighted mean of the observations rating factors, giving more relevance to the past observations. It also used indirect reputation exchanges from other nodes to obtain a global reputation value. The difference between CORE and CONFIDANT is that CORE only allows positive reports to pass through, but CONFIDANT allows negative reports also. This means that CORE prevents false reports. Therefore, it prevents a DoS attack which CONFIDANT cannot do.

In 2002, Albers et al. [16] proposed another distributed and collaborative architecture of IDS by using mobile agents. A Local Intrusion Detection System (LIDS) is implemented on every node for local concern, which can be extended for global concern by cooperating with other LIDS. Two types of data are exchanged among LIDS: security data (to obtain complementary information from collaborating nodes) and intrusion alerts (to inform others of locally detected intrusion). In order to analyze the possible intrusions, data must be obtained from what the LIDS detects on, along with additional information from other nodes. Once the local intrusion is detected, the LIDS initiates a response and informs the other nodes in the network. Upon receiving an alert, the LIDS can protect itself against the intrusion.

In 2003, Kachirski and Guha [17] proposed a hierarchical intrusion detection system based on mobile agent technology. The system can be divided into three main modules, each of which represents a mobile agent with certain functionality : monitoring, decision-making and initiating a response. Monitoring agents perform the network monitoring and host monitoring. Host monitoring agents are present on every node. A distributed algorithm is used to dynamically divide the mobile network into clusters and assigning only a few nodes – cluster heads, to host sensors that monitor network packets and agents that make decisions. Action agent resides on all network nodes, which initiate a response, such as terminating the process or blocking the node from the network, if it meets intrusion activities where it lives. The decision agent is run only on certain nodes, mostly at the nodes that run network monitoring agents. If the local detection agent cannot make a decision on its own due to insufficient evidence of an intrusion, it will report to this decision agent in order to investigate deeply on the suspected node.

Huang and Lee [18], in 2003, proposed another cluster-based cooperative intrusion detection system. This IDS was not only capable of detecting an intrusion but also revealed the type of attack and the attacker using statistical anomaly detection and identification rules.

In 2003, Bo Sun, Kui Wu and Udo W. Pooch [19] introduced a Hierarchical Zone Based IDS (ZBIDS). In this system, the MANET is divided into non-overlapping zones. The nodes can be categorized into two types: the intra-zone node and the inter-zone node (or a gateway node). Others components on the system are data collection module and detection engine, local aggregation and correlation (LACE), and global aggregation and correlation (GACE). The data collection and the detection engine are responsible for collecting local audit data (for instance, system call activities, and system log files) and analyzing collected data for any sign of intrusion respectively. The remainder, LACE module is responsible for combining the results of these local detection engines and generating alerts if any abnormal behavior is detected. These alerts are broadcasted to other nodes within the same zone. However, for the GACE, its functionality depends on the type of the node. If the node is an intra-zone node, it only sends the generated alerts to the inter-zone nodes. Thus, if the node is an inter-zone node, it receives alerts from other intra-zone nodes, aggregates and correlates those alerts with its own alerts, and then generates alarms. The intrusion response module is responsible for handling the alarms generated from the GACE.

In 2005, Sterne et al. [20] also proposed a clustering based dynamic intrusion detection hierarchy in which every node has the task to monitor, log, analyze, respond, and alert or report to cluster heads. Then cluster heads perform the data integration, data filtering, detection of intrusion, and security management. The architecture uses dynamic hierarchy in which detection data is acquired at the leaves and is incrementally aggregated, reduced, and analyzed as it flows upward toward the root. The nodes at the top are responsible for security management functions.

In 2005, Ioanna Stamouli [21] propossed RIDAN architecture which uses timed finite state machine to formally define attack against the AODV routing process. It uses a knowledge based methodology to detect the intrusion. RIDAN operates locally in every participating node and observe the network traffic. This model can able to detect resource consumption attack, sequence number attack and dropping routing packet attack.

In 2006, A. Karygiannis, E. Antonakakis, and A. Apostolopoulos [22] proposed a method to detect the critical nodes for MANET. Critical node is a node whose failure or malicious behavior disconnects or significantly degrades the performance of the network. After identification of critical node, these nodes are continuously monitored. To detect the critical node they used a vertex cut and edge cut approach.

In 2006, S. Bose, P. Yogesh and A. Kanan [23] proposed a "Neural network approach for anomaly intrusion detection in ad hoc network using mobile agents". They used the user log file data obtained from local host for training the neural network for the purpose of intrusion detection.

In Sept 2006, Xia Wang [24] proposed end to end wormhole detection method in wireless ad hoc networks. They used AODV protocol. In the route discovery process, the sender sets the Destination-only flag such that only the destination can be able to respond to the ROUTE REQUEST packet. Once the ROUTE REQUEST packet reaches to the destination, it responds by sending a ROUTE REPLY with its current position. The sender retrieves the receiver's position from the ROUTE REPLY packet and estimates the lower bound of hops between the sender and the receiver. If the received route is shorter than the estimated shortest path, the corresponding route will be discarded. Otherwise, the sender will select the shortest path corresponding to the estimation. After the detection of wormhole by sender, it temporarily enables the path with wormhole and send the TRACE packet to the receiver through this path. This TRACE packet is forwarded by each intermediate node through the route with wormhole. When any node on the route receives the TRACE packet, it replies to sender by sending its current position and hop count to the destination node. Then the sender can calculate the increase of hop count at each node using the received position. If the increase of hop count at one node is not one comparing to its previous hop, then this node and its previous hop node are identified as the wormhole.

In the year 2007, R. Ranjana and M. Rajaram [25] proposed another model which does not perform any change in underlying protocol and used additional security component to detect fabrication attack, resource consumption attack and packet dropping attack.

In June 2008, Ningrinla Marching and Raja Datta [26] proposed collaborative technique for Intrusion detection in MANET. In this, they proposed two intrusion detection techniques for mobile ad-hoc networks, which use collaborative efforts of nodes in a neighborhood to detect a malicious node in that neighborhood. The first technique is designed for detection of malicious nodes in a neighborhood of nodes in which each pair of nodes in the neighborhood are within radio range of each other. Such a neighborhood of nodes is known as a clique. The second technique is designed for detection of malicious nodes in a neighborhood of nodes, in which each pair of nodes may not be in radio range of each other but where there is a node among them which has all the other nodes in its one-hop vicinity. Both techniques use message passing between the nodes. A node called the monitor node initiates the detection process. Based on the messages that it receives during the detection process, each node determines the nodes it suspects to be malicious and send votes to the monitor node. The monitor node upon inspecting the votes determines the malicious nodes from among the suspected nodes. Their IDS is independent of any routing protocol.

S.Madhavi and Dr. Tai Hoon Kim [27] developed another IDS for MANETs in 2008. In their work the author define the monitor node whose job is to detect misbehaving node. They also describe the algorithm for detecting the packet dropping and packet delaying attack.

In 2009, S .Sen proposed a "grammatical evolution approach for intrusion detection in mobile ad hoc networks" [28]. They use artificial intelligence based learning technique to explore design space. The grammatical evolution technique inspired by natural evolution is explored to detect known attacks on MANETs such as DOS attacks and route disruption attacks. Intrusion detection programs are evolved for each attack and distributed to each node on the network.

2.5 Research Gaps

A series of security solutions have been proposed. However, because of limitations of dynamic environment of mobile ad-hoc network- their effectiveness is limited. The following research gaps were identified after critical literature review:

1. The preventive approaches are not capable to defend against all attacks. They cannot guarantee availability for example, it cannot prevent radio jamming. The secure distribution of signatures may be difficult, due to the properties of wireless communication and mobile nodes that operate in disconnect mode. Due to increasing instances of new security threats with the constant emergence of new technology and applications in MANETs, preventive measures are unable to eliminate all possible intrusions.

2. The anomaly and misuse detection techniques are mostly limited to specific routing protocols and attacks:

a. The lack of a single standardized routing protocol makes it difficult to define generalized attack signatures for wireless mobile environment. Signatures are defined from the characteristics, vulnerabilities and the working topologies of the routing protocol. The lack of understanding of new applications that are being developed for the wireless mobile environment also add to the difficulty in defining attack signatures.

b. Due to the bandwidth limitations, battery constraints and frequent disconnects, users often adopt new operations modes such as disconnected operations. A node that sends out false routing information could be a compromised node or merely one that is temporarily out of sync due to rigorous physical movement. Existing anomaly detection methods may find it increasingly difficult to differentiate false alarms from real intrusions.

3. The wireless ad-hoc environment does not have traffic concentration points like switches, routers and gateways where the IDS can collect audit data for the entire network. Watchdog technique, that is overhearing the packets received and transmitted by the neighboring nodes to detect maliciousness, has been used in all of the discussed IDSs. These techniques rely on only partial, localized audit data collected from the host and from communication activities taking place within the radio range. In case of collisions, it cannot work correctly and leads to wrong accusations. When each node has a different transfer range or implements directional antennas, the Watchdog cannot monitor the neighboring nodes accurately.

4. The limitations of Stand-alone IDS are:

   a. Firstly nodes are able to detect misbehaving nodes only upto a certain extent based on their local observations.

   b. Secondly, they are unable to deal with or punish them in any way. Misbehaving nodes can continue to use network resources and continue their usual behaviors.

5. IDS solution requires production nodes of network to actively participate in traffic sniffing, information sharing and decision making processes. These production nodes

can be malicious, selfish or benign. In MANETs, there is no a priori trusted subset of nodes to support the network functionalities; therefore collaboration between strange nodes is not fully utilized because of the potential risk of such collaboration.

a. When distributed systems (cooperative, hierarchical or agent based systems) are used, nodes not only detect malicious nodes within transmission range, but also report these malicious nodes to other network nodes. Attackers use this opportunity to send false alarm messages accusing legitimate node as intruder.

b. The distributed systems using majority voting has their own limitations. All nodes are equally trusted and are allowed to send alarm messages to each other about intrusions. According to majority votes received, individual node determines whether the network is under intrusion or not. They are based on belief that a malicious node will have no incentive to send report that network is under intrusion. However, they are designed to detect intrusions only and not the intruder. These systems required the individual's response mechanism demanding the re-authentication of whole network.

c. Reputation and trust systems exchanges reputation values about different nodes and assign trust levels/values based on their past behavior and on recommendations received from other nodes. Alarm messages from only trusted nodes are then used for analysis. However, these systems are vulnerable to false accusation from second-hand reputation exchanges. A group of malicious nodes can exploit the estimation mechanism to make a node look bad or good by sending false reputation values. CONFIDANT uses both positive and negative values of reputation. It is vulnerable to malicious spreading of false reputation values which can make a good node look bad. CORE overcome this problem by making use of only positive value, but does not address the issue of collusion to create false praise. Therefore, trust relationships in MANETs are established, evolved, propagated and expired on the fly (no infrastructure) and are very susceptible to attacks, as the whole environment is vulnerable due to the shared wireless medium.

Because of absence of any line of defense, which distinguishes nodes as trusted and non-trusted, malicious and selfish nodes can poison the whole data and thus can make the whole decision making process ineffective. As the ratio of bad nodes in the network increases, the ratio of trusted decision making nodes in the network decreases, decreasing the attack detection efficiency of IDS.

6. The lack of any central management facility limits the detection of distributed attacks by IDS.

    a. Although different network nodes exchange alarm messages with each other, they all make their own individual decision regarding intrusion and intruder. They make individual responses based on local observations and global alarms received. Individual nodes observe the failure that occurs in itself or in its nearby locality, this short-range observation cannot present the big network picture to make convincing conclusion about occurrences of globally distributed attacks.

    b. When the attackers change their attack pattern and their attack target in different periods of time, the absence of centralized management machinery makes the detection of attacks a very difficult problem for individual nodes. It becomes extremely difficult to monitor the traffic in a highly dynamic and large scale ad hoc network and leads to false positive and false negative alarms.

7. The resource constraint constitutes another challenge to IDS

    a. The wireless channel is bandwidth-constrained and shared among multiple networking entities. Continuous information exchange between cooperative nodes consumes a lot of network bandwidth.

    b. IDS accuracy itself is a critical issue. In MANETs, the IDS monitor the activities and analyze and compare them against the security rules and accordingly generate the alarm. The limited battery power of mobile nodes limits their intelligence and also causes some nodes to behave in a selfish manner during cooperative intrusion detection processes.

c. The other power consuming functionalities of mobile nodes like LCDs can be switched off when not in use, but IDS runs at these nodes even when they are not doing anything. Lots of redundant data is collected by continuous monitoring. This leads to processing power wastage in filtering. The continuous processing done by production nodes for security slowly depletes the energy available for the useful production work.

8. The hierarchical IDS suffers from dynamically cluster head selection overheads. As the physical network arrangement changes, cluster membership is dynamically updated – the algorithms used to make the clusters and chose the cluster heads are needed to run several times according to the changing network topology.

9. In distributed IDS systems, the working and vulnerabilities of IDS are known to all. An IDS system may detect attacks on mobile hosts, but an attacker can attack IDS system itself. A knowledgeable attacker can bypass the security rules of IDS, so protection of IDS against attacks is required. In mobile agent based systems, the agents themselves may be the primary target of an attack.

10. Deceptive measures are used for only wired networks or for infrastructure based- wireless networks (Access points based wireless networks) only where traffic concentration or control points like gateways, routers and access points exist. Many authors have suggested that deceptive measures can be effective in ad-hoc environment also, but because of absence of any such concentration points, till now no proper architecture model has been proposed.

In this work, we have tried to address some of the research gaps discussed above using deceptive techniques. We have combined detective techniques (Misuse and Anomaly based detection engines) and deceptive approaches to develop a deception based hierarchical intrusion detection system to counter network layer attacks in ad-hoc environment.

CHAPTER 3
PROPOSED DECEPTION BASED HIERARCHICAL IDS FOR MANETs

In this work, we propose the use of a deceptive, trusted and controlled mobile network in the vicinity of real production ad-hoc network as a trap for attackers. We have coined the term – HoneyMANET for this decoy mobile ad-hoc network that acts as a bait to lure the attackers. It is made of many trusted nodes called honeynodes that are always cooperating, communicating and sharing data under hidden central management to capture the network based intrusions. It works by deviating the attention of attackers from real production network and is open to foreign nodes for joining (with or without authentication according to the security policies). This chapter gives the complete architecture, design and working of HoneyMANET.

3.1 System Architecture

The overall architecture of HoneyMANET system proposed in our dissertation includes four modules – Deception Module, Monitoring and Logging module, Collection and Integration Module and Intrusion Detection Module as shown in Figure 3.1.

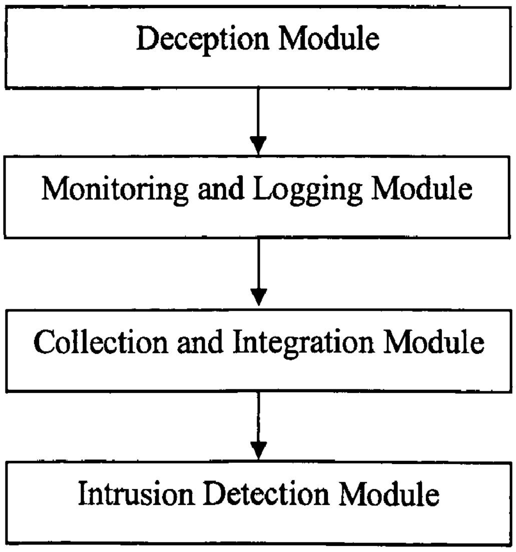
Deception Module
Monitoring and Logging Module
Collection and Integration Module
Intrusion Detection Module

Figure 3.1 HoneyMANET's Modules

The proposed special trusted nodes are called honeynodes. They make the initial mobile network to lure the attackers. They are deployed in or around the arena where we want to detect intrusions. They are multifunctional- they are mobile and they generate fake traffic to create deception of a real ad-hoc network, they act like traffic sensors/sniffers for monitoring the activities of foreign nodes crossing by, they have their own backend communication network for relaying the monitored traffic and have intrusion detection intelligence (IDS).

There are two different networks in HoneyMANET: Frontend wireless ad-hoc network and backend hidden and controlled mobile wireless network. Front-end network is made of mobile honeynodes and foreign nodes (Malicious/Neutral/Benign production nodes) that join the network. The back-end network is made of only honeynodes. The deception module is implemented as the front-end network. The other three modules – monitoring and logging, collection and integration, and intrusion detection intelligence are implemented in back-end network using honeynodes. It is this backend controlled mobile network – the backbone of HoneyMANET that detects the attacks and captures the attackers.

3.1.1 Deception Module

HoneyMANET's frontend mobile ad-hoc network is the deception module created to lure the attackers. The frontend network sees the whole network as one and is configured in a flat network kind of architecture [4], where all nodes – honeynodes and foreign nodes are considered to be equal and may participate in routing functions. Honeynodes in this module do not have any production function except generation of fake traffic and have mobility pattern according to application for which they are deployed, for example, when deployed on freeways, they follow the mobility model of freeway mobile nodes. This network looks like any other general ad-hoc network and have security and authentication policies that are visible to the world and to the attacker, and are used to lure and trap the attackers.

3.1.2 Monitoring and Logging Module

As member of hidden back-end network, honeynodes act like traffic sensor and continuously monitor and log all the network layered activities of the frontend ad-hoc network. Network adapter card allow mobile nodes to listen in monitor mode all the network traffic without even associating with any network. Softwares such as as KisMAC or Kismet in combination with packet analyzers such as wireshark or Tcpdump provide a user interface for passive wireless network monitoring. The wireless traffic is collected in PCAP format. The effectiveness of this module lies in sniffing maximum amount of network traffic, covering maximum arena possible, with minimum number of honeynodes.

3.1.3 Collection and Integration Module

The back-end network consisting of only honeynodes relays the network traffic sniffed to specific intrusion detection engines for analysis. The backend network follows multi-layered network architecture [4] where all nodes are not considered equal. It sees the network as divided into zones for hierarchical management. Each zone in HoneyMANET's backend network is supposed to have a zone head. Each mobile honeynode has the responsibility of monitoring traffic of its own zone. At backend network, sensed data from different zones is collected at their zone heads and is integrated and analyzed for traces of local intrusion. The zone-head of central square region (of whole topology), is assigned the function of central collecting nodes. This node collects data from all other region heads for global analysis. The zone heads and central collecting node are given high memory and processing resources to integrate the collected data and to perform the intrusion detection tasks. The whole data collection process is made as obscure as possible from the outside world.

Table 3.1 gives the location, functionality and number of different kinds of honeynodes as member of back-end network.

Table 3.2 gives the differences between front-end network and back-end network.

Figure 3.2 shows the entire HoneyMANET consisting of front-end and backend network

Table 3.1 Honeynodes

Honeynodes	Location	Functionalities	Number
Localised honeynodes	Mobile within their defined zone	They roam, sniff the data from the frontend network in promiscuous/monitor mode, generate fake data into the frontend network to deceive the attacker.	Depend on the illusion we want to create in each zone.
Zone-heads honeynodes (Special kind of localised honeynodes)	Mobility within their defined zone	Along with the functions of localized honeynode, they also collect data from localized honeynodes within their regions and do the local analysis.	Depend on number of zones we have –One for each zone
Central Collecting nodes (Special kind of Zone-head honeynode)	Mobility within central zone	Along with functions of zone-head honeynode, it also collects global data from zone-heads of all zones and does the global analysis.	One for whole network

Table 3.2 Frontend Network Vs Backend Network

Front-end Mobile Ad-hoc Network	Backend-hidden Mobile Network
It has a flat network architecture	It has a hierarchical network architecture
It is a deception of an ad-hoc mobile network to lure the attacker	It is the hidden, controlled and planned mobile network deployed for security purposes
It is made up of mobile honeynodes and foreign nodes	It is made up of only mobile honeynodes
Honeynodes's function is the emulation of real network by generating fake traffic and by its mobility.	Honeynodes's function are surveillance, relaying and Intrusion Detection
The ids and other security solutions implemented in	Its intrusion detection mechanism is known only to

it involves all network nodes – honeynodes and foreign nodes in decision making process like any general ad-hoc network.	honeynodes and involves only honeynodes for decision making process.

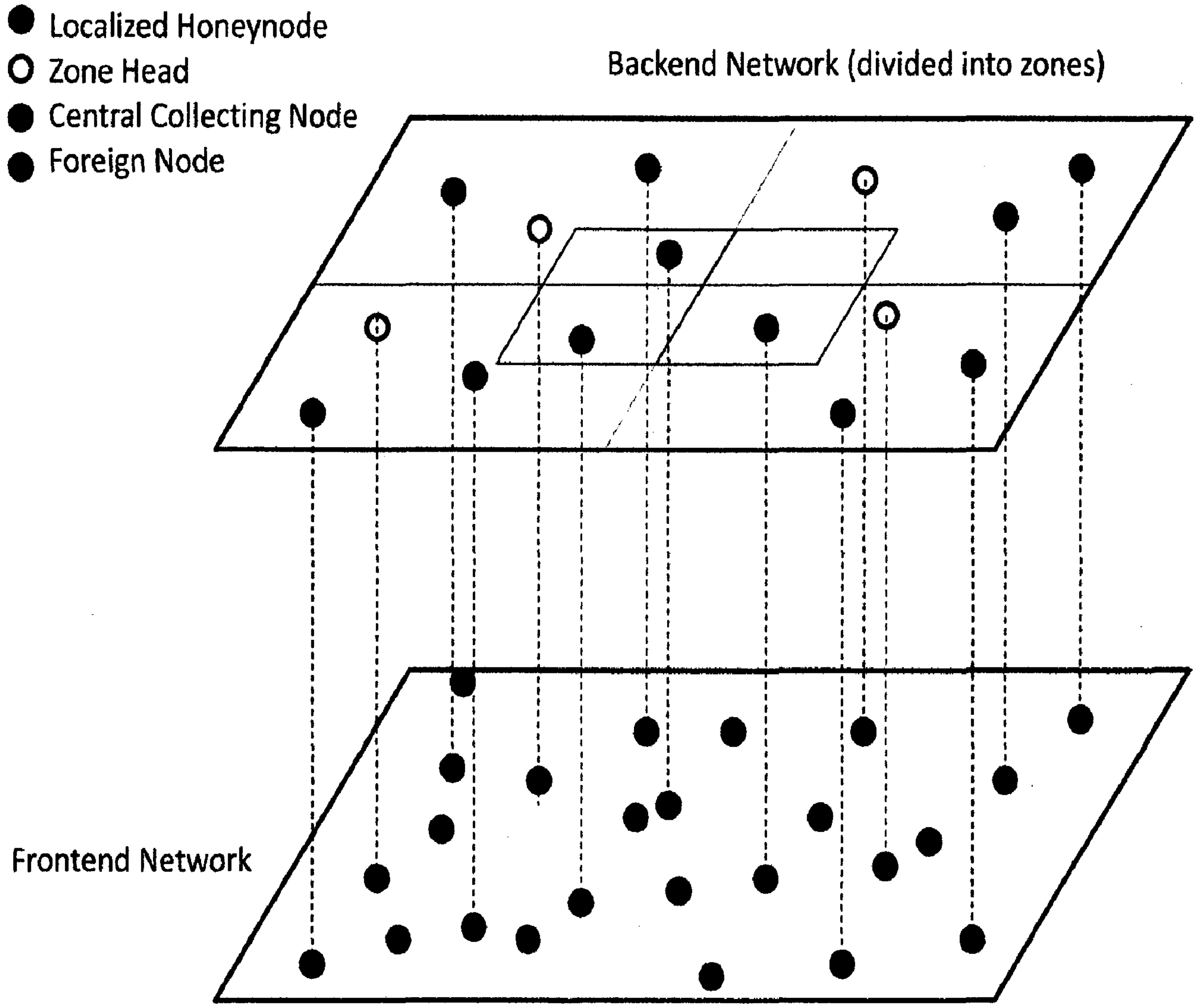

Figure 3.2 HoneyMANET's Design

3.1.4 Intrusion Detection Module

The HoneyMANET consisting of just Honeynodes is said to be in IDLE state. When a foreign node joins the network, it triggers the HoneyMANET and set it into ACTIVE state. The new node that joins the network is considered suspicious. This node will then be marked as – malicious, selfish, neutral or benign by the HoneyMANET based on the behavior of node in HoneyMANET environment as shown in Table 3.3.

HoneyMANET, uses a combination of anomaly and misuse based intrusion detection techniques for generalized intrusion detection.

Table 3.3 General Event Action Plan

Event (Trigger by Analysis Module)	Action- mark behavior as	Threat Level/Value
Node has just joined the network	Suspicious	0
Behave normally- no harm, contribute and consume resources	Benign	0
Don't do anything- no harm, no resource consumption/contribution- (when intensions are not known, might be eavesdropping)	Neutral	1
Don't cause harm to other nodes, only consume network resources	Selfish	2
Cause harm to other nodes, consume network resources	Malicious	3

The steps involved in generalized intrusion detection are given as:

Step 1: From the integrated database, rule traces describing the behavior of nodes and of entire network are generated. Rule Traces are sent at end of T1 interval to hybrid Anomaly and Misuse based Intrusion Detection module for analysis.

Step 2: In the anomaly detection engine, if the rule trace deviates from the normal profiles then the anomaly is detected.

Step 3: The rule trace is then sent to online signature based intrusion detection engine, which matches the rule trace with existing rule base consecutively for 't' T1 time intervals. If the match is found for 't' T1 time intervals then the attack is marked to be identified.

Step 5: If the attacker is not identified in misuse detection engine then rule trace is sent for confirmation of attack phase. Offline analysis on globally and locally collected databases is done to find the intruders.

Step 6: If only the ‘t’ T1 time interval threshold is passed (without match), then rule trace is sent to existing knowledge database as new attack rule.

Step 7: The left out rule traces are eliminated as false positives.

Figure 3.3 gives the flowchart of overall hybrid Anomaly and Misuse based intrusion detection module:

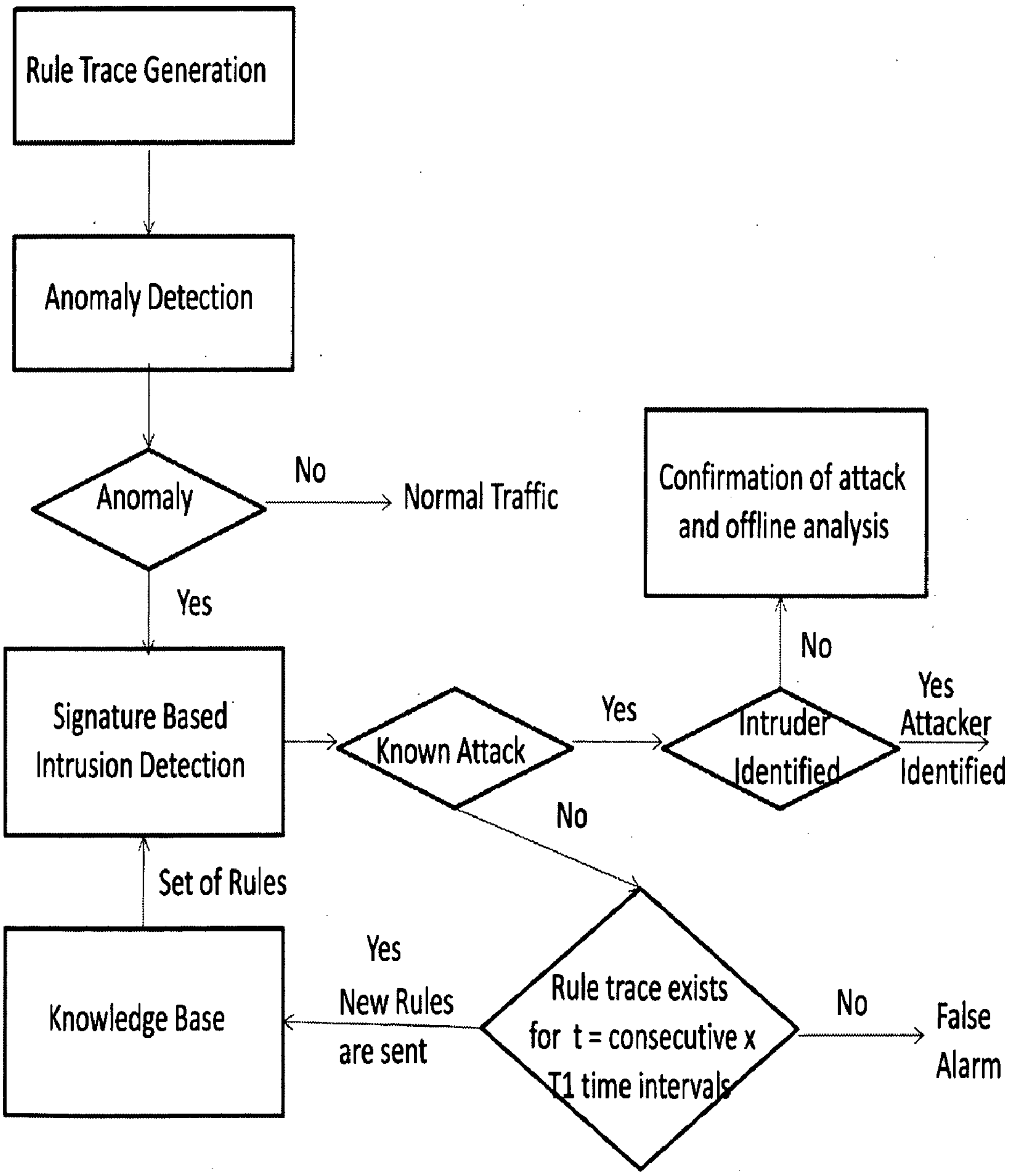

Figure 3.3 HoneyMANET’s Hybrid IDS

3.2 Rule Trace Generation

Two types of rule traces are generated. For each node in the network, rule traces showing a node's behavior in the network for T1 time interval are created (using the integrated database of T1 time interval). These rule traces consist of primary and secondary parameters. Rule traces for an entire network describing the behavior of network globally after a time interval T1 are also generated. These rule traces consist of global parameters.

3.2.1 Rule Trace Generation for a Node

The primary parameters are: {PS, PRF, PSF, PR, RR, RP, RE}, where PS = Packets send by node as sender, PRF = Packets received by node to forward, PSF = Packets forwarded by node, PR = Packets received by a node as destination, Control Packets generated by a node (Route Requests - RR, Route Reply - RP, Route Error - RE). The following pseudocode shows the rule trace formation for a node from data packets. Node[i][j] represents node 'j' belonging to zone/area 'i', last_hop is the last hop information from the packet, and source is the packet sender. Pseudo code for extracting primary parameters from data packets is given in Figure 3.4.

```
Do for each area
Do For each packet belonging to area i
    if ( last_hop == source && Packet_type == data_packet ) Node[i][last_hop].PS++;
    else if (last_hop == source && Packet_type == Route_Request) Node[i][last_hop].RR++;
    else if (last_hop == source && Packet_type == Route_Reply) Node[i][last_hop].RP++;
    elseif(last_hop == source && Packet_type == Route_Error) Node[i][last_hop].RE++;
    else if (Packet_type == data_packet) Node[i][last_hop].PSF++;
    Next_node = receiving node in packet path;
    if (Next_node != destination && Packet_type == data_packet) Node[i][next_node].PRF++;
    else if (Packet_type == data_packet ) Node[i][next_node].PR++;
```

Figure 3.4 Pseudo code for extracting primary parameters

From the primary parameters the following six secondary parameters as given in Table 3.4, are derived: Node[i][j].parameter[k][m] represents secondary parameter 'k' value at 'm' time interval ('0' representing current time) for a node 'j' belonging to area 'i'.

Table 3.4 Secondary Parameters

Node[i][j].parameter[0][0] = Data send rate of node = (Bytes send by node as sender / T1)
Node[i][j].parameter[1][0] = Data Packet Drop ratio of node = ((Packets received by node to forward - Packets forwarded by node) / Packets received by node to forward)
Node[i][j].parameter[2][0] = Route Request send ratio of node = (Route Request send / Data Packets Sent)
Node[i][j].parameter[3][0] = Error Generation rate of node = (Route error generated / T1)
Node[i][j].parameter[4][0] = Route Reply Send Ratio = (Route Reply sent / Packets Received)
Node[i][j].parameter[5][0] = Rate at which packets are sent for this node to forward = (Bytes received by node to forward / Time)
Node[i][j].parameter[6][0] = Data delivery ratio as sender = (Packets received at destination / Packets send by node as sender)
Node[i][j].parameter[7][0] = Data delivery ratio as receiver = (Packets received by this node / Packets sent for this node)

3.2.2 Rule Traces for the network

The three global Parameters, as given in Table 3.5, create the network's rule trace after each T1 time interval. Global.Parameter[i][t] represent global Parameter 'i' at time instant 't', where t = 0 represent current time.

Table 3.5 Global Parameters

Global.Parameter[0][0] = Throughput rate = $\sum_{i,j}$(Bytes received by Node[i][j]) /T1
Global.Parameter[1][0] = Packet drop rate = $\sum_{i,j}$(Bytes received by Node[i][j] to forward – Bytes forwarded by Node [i][j]) /T1
Global.Parameter[2][0] = Control Packet Overhead = Control / Data Packets of whole network in T1

3.3 Unsupervised Anomaly Detection

Anomaly detection techniques work by observing the normal network for long training phases and then building a normal profile of network behavior. Any network activity deviating from this normal profile is then detected as anomaly. The honeynodes in our HoneyMANT have always 100% trust factor associated with them. They always behave normally in the network. Therefore, their average behavior at any instant of time is seen as normal network behavior. At any instant of time, the behavior of any foreign node can be compared with the behavior of honeynode to detect anomaly. This possibility of using trusted nodes behavior at any time to find anomalies obviates the need to have a training phase to generate normal profile. Thus, we have an unsupervised anomaly detection engine.

Step 1: After each time unit T1, three types of normal profiles are created: Local Profile for a zone, Personal Profile for a node and Global Profile for the network using the rule traces.

Step 2: Chi-square Test is conducted to find the deviation of rule traces from normal profile.

3.3.1 Local Profile Generation for each zone

Local Profile is created taking the average and variance of rule traces (described by primary and secondary parameters) of honeynodes belonging to that zone (defining the normal zone behavior) at that instant. The rule traces of foreign nodes are then compared with this local profile for local intrusion detections. Equations 1 and 2 are used for the creation of local profile for a zone 'i'. Expected[i][k] represents the average value of parameter 'k' for zone 'i' , variance[i][k] gives the variance in that parameter and honey[i][j].parameter[k][0] represents the value of parameter 'k' at current time (t=0) for honeynode 'j' belonging to zone 'i'. Parameter 'k' is varied from 0 to 5.

$$\text{Expected}[i][k] = \frac{\sum_j (\text{honey}[i][j].\text{parameter}[k][0])}{\sum_j (j)} \tag{3.1}$$

$$\text{Variance}[i][k] = \frac{\sum_j (\text{honey}[i][j].\text{parameter}[k][0] - \text{Expected}\ [i][k])^2}{\sum_j (j)} \tag{3.2}$$

3.3.2 Personal Profile Generation of each node

Personal Profile captures the behavior of a node like sender's delivery ratios and receiver's receiving ratio with time. This profile of each node is created over 't' T1 past time intervals. Comparison with this profile detects deviations from its own earlier profile and mark that as anomaly from its earlier behavior. Equation 3 and 4 are used for the creation of personal profile for a node 'j' belonging to zone 'i'. Expected[j][k] represents the average value of parameter 'k' for node 'j', variance[j][k] gives the variance in that parameter and Node[i][j].parameter[k][t] represents the value of parameter 'k' for the past 't' = 3T1 time intervals for a node 'j' belonging to zone 'i'. Parameter 'k' is varied from 6 to 7.

$$\text{Expected}[j][k] = \frac{\sum_t (\text{Node}[i][j].\text{parameter}[k][t])}{\sum_t (t)} \tag{3.3}$$

$$\text{Variance}[j][k] = \frac{\sum_t (\text{Node}[i][j].\text{parameter}[k][t] - \text{Expected}[j][k])^2}{\sum_t (t)} \tag{3.4}$$

3.3.3 Global Profile Generation for network

Global Profile captures global network behavior with time. This profile is created using global parameters- throughput rate, packet loss rate, control packets overhead of network. A deviation in this profile suggests anomaly from earlier global network behavior. Equation 5 and 6 are used for the creation of global profile for the network. GExpected[i] represents the average value of parameter 'k' for the entire network, variance[j][k] gives the variance in that parameter and

Global.parameter[k][t] represents the value of parameter 'k' for the past 't' = 3T1 time intervals for the network globally. Parameter 'k' is varied from 1 to 3.

$$\text{GExpected}[i] = \frac{\sum_t (\text{Global.Parameter}[i][t]\ )}{\sum_t (t)} \tag{3.5}$$

$$\text{GVariance}[i] = \frac{\sum_t (\text{Global.Parameter}[i][t] - \text{GExpected}[i]\ )^2}{\sum_t (t)} \tag{3.6}$$

3.3.4 Chi-Square Test:

Chi-sqaure tests are conducted to find the deviation of rule traces from normal profiles. Eq 7, 8 and 9 gives the Local Chi-Square Test conducted on rule traces of foreign nodes (Parameter 'k' is varied from 0 to 5), Personal Chi-Square test conducted on rule traces of all network nodes (Parameter 'k' is varied from 6 to 7) and Global Chi-Square test conducted for the entire network (Parameter 'k' is varied from 0 to 2).

$$\chi^2[j] = \frac{\sum_t (\text{Node}[i][j].\text{parameter}[k][0] - \text{Expected}[i][k]\ )^2}{\text{variance}[i][k]} \tag{3.7}$$

$$\chi^2[j] = \frac{\sum_t (\text{Node}[i][j].\text{parameter}[k][0] - \text{Expected}[j][k]\ )^2}{\text{variance}[j][k]} \tag{3.8}$$

$$\chi^2 = \frac{\sum_t (\text{Global.Parameter}[i][0] - \text{GExpected}[i]\ )^2}{\text{Gvariance}[i]} \tag{3.9}$$

Three different P values are used as shown in Table 3.6 to reject or accept the hypothesis. The value of P for which detection accuracy is high is then used for detection.

Table 3.6 P values for Chi Square Test

Test	Degree of Freedom	χ^2 value		
Local Chi-Square	6	12.59	16.81	22.46
Personal Chi-Sqaure	2	5.99	9.21	13.82
Global Chi-Square	3	7.82	11.34	16.27
P value (Probability)		0.05	0.01	0.001

3.4 Signature/Misuse based Intrusion Detection

Whenever an anomaly is detected, it is sent to Signature based IDS for identification using intruder identification rules specific to the known attack. To optimize the probability of identifying intruders correctly with a low level of false positives, it maintains a test sliding window (TSW) , in which 't' consecutive detections of an anomaly are required. If this detection threshold is passed then the anomaly is confirmed to be an attack. The rule base used for signature based detection is given in Table 3.7 and Table 3.8

Table 3.7 Rule base for Signature based Detection

Rule	Attack is constant for TSW (t = 3T1 time intervals)
Node's Route Request send ratio >> Expected	Flooding Attack Attacker
Node's Packet drop ratio >> Expected && Packets received by node to forward – PRF ~ Expected	Simple Packet Drop Attacker
Node's Packet drop ratio >> Expected && Packets received by node to forward – PRF >> Expected	Black Hole Attacker

Packets received by node to forward – PRF << Expected && Packets send by node as sender – PS >> or ~ Expected	Selfish Node (not participating in packet forwarding)
Packets received by node to forward – PRF << Expected && Packets send by node as sender – PS << Expected	Neutral Node (node may be eavesdropping)
Packets received by node to forward – PRF >> Expected && Node's Packet drop ratio ~ Expected	Node is victim of sleep deprivation attack
Node's Data delivery ratio as sender << Expected	Node is victim of Denial of Service Attack
Node's Data delivery ratio as receiver << Expected	
Global Throughput << Expected	Network is under Distributed Denial of Service Attack
Global Packet Drop rate >> Expected	
Global Control Packet Overhead >> Expected	

The meaning of different symbols used in Table 3.7 is given in Table 3.8.

Table 3.8 Meaning of Symbols used in Table 3.7

>>	Node[i][j].parameter[k][0] > (Expected[i][k] + c* std_dev[i][k]) (For time independent parameters) Node[i][j].parameter[k][0] > (Expected[j][k] + c* std_dev[j][k] (For time dependent parameters) Global.parameter[k][0] > (Expected[k] + c* std_dev[k] (For global time dependent parameters)
<<	Node[i][j].parameter[k][0] < (Expected[i][k] - c* std_dev[i][k]) (For time independent parameters) Node[i][j].parameter[k][0] < (Expected[j][k] - c* std_dev[j][k] (For time dependent parameters) Global.parameter[k][0] < (Expected[k] – c* std_dev[k] (For global time dependent parameters)
~	Node[i][j].parameter[k][0] < (Expected[i][k] + c* std_dev[i][k]) (For time independent

	parameters) && Node[i][j].parameter[k][0] > (Expected[i][k] - c* std_dev[i][k]) (For time independent parameters) Node[i][j].parameter[k][0] < (Expected[j][k] + c* std_dev[j][k] (For time dependent parameters) && Node[i][j].parameter[k][0] > (Expected[j][k] - c* std_dev[j][k] (For time dependent parameters) Global.parameter[k][0] < (Expected[k] + c* std_dev[k] (For global time dependent parameters) && Global.parameter[k][0] > (Expected[k] - c* std_dev[k] (For global time dependent parameters)

Different c values used for misuse detection depend on P value used for anomaly detection are given in Table 3.9.

Table 3.9 ‘c’ values for Misuse Identification

Degree of Freedom	1		
χ^2	3.84	6.64	10.83
C value = $(\chi^2)^{\frac{1}{2}}$	1.96	2.58	3.29

3.5 New Rule Generation and Attack Confirmation

3.5.1 New rule generation

If the misuse based detection engine detects the same anomalous kind of rule trace for TSW (consecutive ‘t’ time intervals) which does not match any of the rules from the rule base, then that rule trace is marked as new attack and is sent to existing knowledge base as new rule.

3.5.2 Confirmation of an attack and offline analysis phase

The hybrid intrusion detection module can be followed by a confirmation of attack phase or an offline analysis phase for the case when misuse based engines identifies the presence of a victim in the network but cannot identify the attacker node.

When anomaly is detected in the Packet delivery ratio of sender (victim is sender node) this can imply two things- either the receiver node is dead (in which case receiver node will not be receiving data from any node), or the path can be broken or the sender node can actually be the victim of selective DOS attack. Local honeynodes can confirm the reason for the anomaly in packet delivery ratio by sending data to the destination node and can check if their delivery is also failing. If delivery is not failing than the node is confirmed to be a victim of selective DOS attack.

When anomaly is detected in the Packet receiving ratio at the receiving end, this could imply either the path is broken, or the senders have stopped sending data or node is a victim of selective DOS attack. Again local honeynodes at sender end can send data to local honeynode at receiver region and can check their delivery to confirm the presence of any attack.

Offline analysis can be done on globally and locally stored database to detect the attacker in case a victim of an attack is detected. Global data is needed to be analyzed to trace back the attackers and their locations if distributed attack is launched.

3.6 DEPLOYMENT SCENARIOS for HoneyMANET

HoneyMANET can be deployed at isolated locations for research purposes and can also help to mitigate attacks at production networks when deployed with them. Table 3.10 and Figure 3.5 show the three different scenarios how HoneyMANET can be deployed. In first scenario, it is deployed at isolated locations for research purposes only. In second scenario, it is deployed in the neighborhood of production systems. In third scenario, HoneyMANET's network is overlapped with production network.

Table 3.10 Deployment Scenarios of HoneyMANET

	Scenario 1	Scenario 2	Scenario 3
Deployed as	Isolated HoneyMANET	In vicinity of Production network	Overlapping Production Network
Purpose	Research	Production HoneyMANET	Production HoneyMANET
Application	As a general random MANET	In battlefields	On Freeways and urban areas streets

1. HoneyMANET with honeynodes moving in random fashion can be deployed at isolated locations to gather research information about the motives and tactics of the attackers in ad-hoc environment. Such a HoneyMANET system does not add direct value to a specific organization. It is the knowledge that is extracted from gathered information which is used to protect against those threats.
2. In battlefield kind of environment, MANETS are generally closed (only authenticated nodes are allowed to join a group) and they may or may not have IDS agents at production nodes. In such scenarios, HoneyMANET can be deployed as another military group with same or less intense security policies in neighbourhood of production military groups. A legitimate node is expected to try and join only the military group it is authorized to join. Therefore, whenever a foreign node tries to join the HoneyMANET, it is marked as suspicious; its activities are monitored and analyzed. And if that foreign node succeeds in joining the HoneyMANET it is seen as breach in security. The security standards of production MANET groups are then upgraded. In such scenarios, HoneyMANET system works in conjunction with intrusion detection systems and helps in lessening the load, and providing a secure and dependable environment for real production military groups to work. It helps in deviating attention of the attackers, in evaluating how much effective various authentication and authorization policies are, and to capture the attackers and their attack mechanisms.
3. In freeways or urban streets kind of environment, HoneyMANET is deployed overlapping the production network. If the production network does not have any protection and is completely open to join, then any roaming users can try to connect to it. Government and private enterprises usually set up open MANETs to provide free access to information to users. HoneyMANET deployed overlapping such networks will help us in knowledge acquisition not only about attackers but also about the usage of MANETs and its different application in those scenarios. In such scenario, our frontend ad-hoc network will be the underlying production network which will be monitored by our honeynodes. It will help us to evaluate how many users joined the network, how many of them were just benign roaming users, how many were attacker, what was the purpose of users joining the network, and about sophistication of users/attackers.

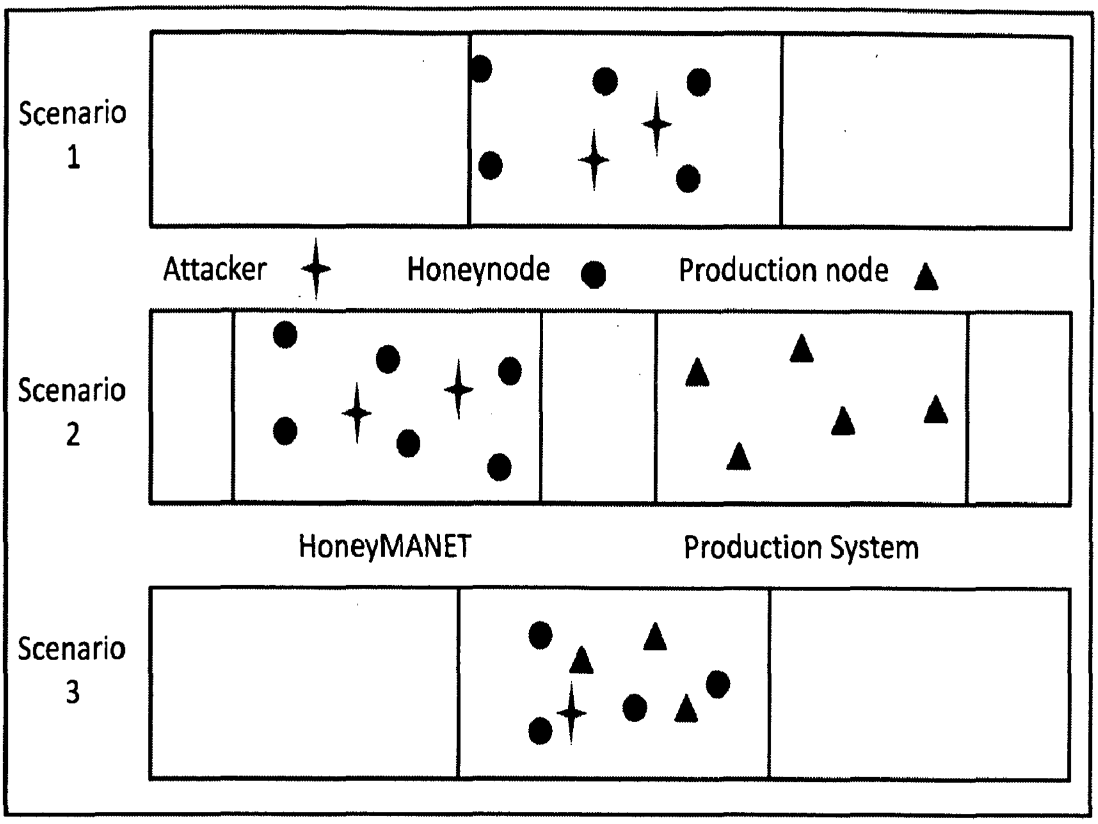

Figure 3.5 Deployment Scenarios of HoneyMANET

3.7 Summary

HoneyMANET has many advantages over IDS in general ad-hoc network. Table 3.11 compares implementing IDS in general ad-hoc environment with trusted and controlled HoneyMANET.

Table 3.11 IDS in General Ad-Hoc Network Vs. HoneyMANET

IDS in general ad-hoc environment	HoneyMANET – trusted and controlled mobile ad-hoc network
Detection and identification of distributed attacks are difficult because of lack of centralized monitoring and central management point	HoneyMANET has controlled hierarchical management.
IDS agents are installed on mobile production nodes which have constraints on processing	Traffic sensing and analysis fucntionalities are given to honeynodes which have no other production work than providing securtity and

capabilities and energy.	therefore can afford to have more intelligence and energy utilization for security purposes.
Data collected from Production nodes- which can be malicious or selfish or benign is used in decision making process. Therefore, there always remain trust issues with the analysis results that are obtained.	100% trusted Honeynodes are used. As the foreign nodes are not involved in decision making process, therefore there is no need to associate trust with them or with the data that is collected.
The detection efficiency degrades as more and more of malicious nodes participate in the decision making process.	The increase in number of malicious nodes doesn't degrade the detection efficiency as they are not involved in decision making process. The number of trusted nodes - honeynodes involved in decision process remains the same.
The IDS intelligence used for detection is known to all the production nodes and can be easily manipulated. IDS itself can be attacked.	The sensing and analysis is hidden from the foreign nodes. HoneyMANET has the advantage of deception and detection both.
Lot of redundant data is being communicated in the production network.	The data is communicated through backend network.
Fully random and dynamic network	Honeynodes control the randomness and congestion of network to some extent.

CHAPTER 4
IMPLEMENTATION DETAILS

The overall process model of HoneyMANET is simulated using Network Simulator- NS2 [29]. Firstly, simulations are done to find different HoneyMANET's design parameters like speed, number and data generation rate of honeynodes in front-end network. Then, using these parameters, the deceptive network topology is simulated in NS2. The logging and integration modules are also simulated. Different kinds of attacks scenarios are simulated to test the robustness and detection efficiency of HoneyMANET.

4.1 Simulated Network Topology

The Random Waypoint model is the most commonly used mobility model in research community for MANETs. Therefore, we have simulated honeyMANET as random waypoint model using NS2's node-movement generator and have conducted our experiments further in this environment. The simulated topology is given in Table 4.1.

```
./setdest [-n num_of_nodes] [-s speedType (uniform/normal distribution)] [-M maxspeed] [-p
pausetime] [-t simtime][-x maxx] [-y maxy] > [outdir/movement-file]
```

In our work, we have targeted commercial-purpose client-server type of MANETs scenarios. Using NS2 we have simulated a MANET network of 45 nodes spread over 1000x1000 meter sq region with single server (honeynode) providing service to rest of the nodes. At network layer, routing protocol used for simulation is Dynamic Source Routing Protocol [6]. Clients send constant bit rate (cbr) traffic at varying rates to server over UDP connections. Each packet is of size 1000 bytes.

Table 4.1 Simulated Network Topology

Mobility Model Used	Random Waypoint Model
Total Number of Nodes	45
Topology	1000x1000
Routing Protocol	DSR
Application Model	Client-Server Type
Traffic Type	CBR Traffic over UDP Protocol
Packet Size	1000 Bytes
Vulnerable Layer	Network Layer

4.2 Determining the Design parameters of HoneyMANET's front-end network

Honeynodes are multifunctional- they generate fake traffic (Emulation of real Network), they act like traffic sensors/sniffers (Surveillance), they have their own backend communication network (Relaying) and intrusion detection intelligence (IDS). In general mobile ad hoc networks (MANETs), production node's mobility and network traffic are uncontrolled factors that lead to its congestion, dynamism and randomness. In HoneyMANET, the mobility and traffic generation rate of Honeynodes is controlled and is decided such that they can perform any one or all the following tasks better:

1. Surveillance – The effectiveness of surveillance is estimated by measuring the percentage of network traffic sniffed by honeynodes (acting as traffic sensors). The percentage of traffic sniffed /scanned by Honeynodes should be as high as possible.
2. Emulation of real Network- Good deception means high randomness in mobility and traffic generation. It is needed that mobility and traffic generation of honeynodes should not be over controlled and should give a feel of real ad-hoc network.

3. Relaying- For relaying the sniffed traffic successfully to common intrusion detection engine we need controlled – less random and lowly congested network, to reduce/minimize the cost of communication and to have less packet loss.
4. Intrusion Detection- Attacks in an ad-hoc network can occur globally or they can have only local effect. To detect, identify and mitigate attacks at their initial stages (before affecting the whole of the network), an overview of network activities both at specific regions/zones and at global level is needed.

Experiments are conducted to find the appropriate value for the following design parameters of front-end network:

1. Initial Mobility or randomness of Honeynodes
2. Initial Congestion Level (traffic generation rate) of HoneyMANET

4.2.1 Mobility and Randomness of Honeynodes

Randomness of network depends on - number of network nodes, area coverage of network nodes, their speed, pause time, transmission range which affects the number of radio link changes in the network and so the topology and dynamism of network. The first scheduling problem in HoneyMANET's designing is to formulate how the honeynodes should move and position themselves in the network in order to enhance global security. Experiments are done to find the optimal coordination of the honeynodes movements to optimize network wide performance metrics.

4.2.1.1 Determining the free-movement zones

Intrusion detection requires good surveillance of whole region - zone wise and also globally. In this step, we first determine the size and number of free-movements zones in which to divide the topology to achieve the communal mission of deception, surveillance (tracking/scanning), relaying and intrusion detection. In the simulations, only the honeynonde's movement is made restricted to specific regions or zones. The foreign nodes joining the network are allowed to move anywhere in the whole topology.

In our experiments, four different types of free-movement zones are simulated in 1000x1000 meter sq region as shown in Figure 4.1. In first scenario, honeynodes can move freely in whole 1000x1000 meter sq. region. In second scenario, the whole region is logically divided into 5-

500x500 meter sq. zones. In third case, whole region is divided into 9 -350x350 meter sq. zones. In another case, whole region is divided into 16 - 250x250 meter sq. regions and a mesh of fixed honeynodes is deployed at center of each region.

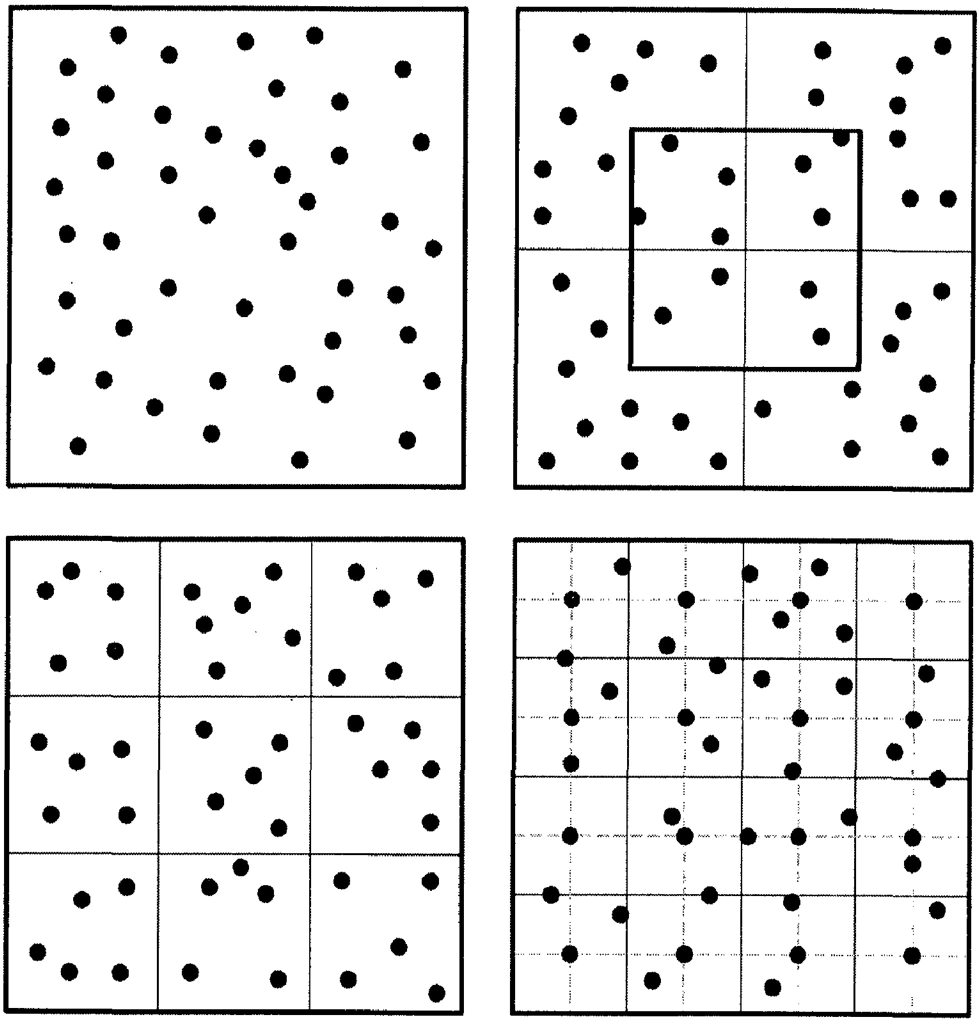

Figure 4.1 Free Movement Zones: All in One Scenario, Five Zones Scenario, Nine Zone Scenario and 16 Zones Mesh Network

Simulated Scenarios:

We conducted experiments to determine the effect of varying the number of honeynodes in different zones scenarios on the percentage sniffing of honeynodes. Table 4.2 gives the different simulated scenarios.

Table 4.2 Simulated Free Movement Zones

Division Type	Survellience Area Size	Total Number of Honeynodes
1 free-movement zone	1000x1000 m-sq	(9, 18, 27, 36, 45)
5 free-movement zones	500x500 m-sq	(10, 20, 25, 35, 45)
9 free-movement zones	350x350 m-sq	(9,18,27,36,45)
Mesh Network of fix honeynodes	250x250 m-sq	(Single Mesh of 16 honeynodes)

The speed, pause time, data send rate and transmission range of nodes are kept at low constant values to nullify their effects on sniffing capability as given in Table 4.3.

Table 4.3 Simulation Parameters used for Free Movement zones

Number of total nodes in the network	Speed	Pause time	data send rate	Transmission range
45	10m/sec	2.0 sec	8kbps	250 m

4.2.1.2 Determining the speed and number of honeynodes

The next experiment is conducted to decide the speed and number of honeynodes to use in each zone.

Simulation Scenarios:

We conducted experiments to see the effect of varying number of honeynodes and their speed on percentage traffic sniffed by them as given in Table 4.4.

Table 4.4 Varied Simulation Parameters for Speed determination

Speed of Honeynodes	(10m/sec, 30m/sec, 50m/sec)
Number of Honeynodes	(10, 20, 25, 35, 45)

The free-movement region is decided from experiment 1. The pause time, data send rate and transmission range of all nodes are kept at low constant values to nullify their effects as given in Table 4.5.

Table 4.5 Constant Simulation Parameters for Speed determination

Number of total nodes in the network	Pause time	data send rate	Transmission range
45	2.0 sec	8kbps	250 m

This experiment helps in decision the restriction zone, and the number and speed of honeynodes to use in those zones.

4.2.2 Initial Congestion Level of HoneyMANET

Network Congestion in MANETS leads to packet loss, increase in queueing delay and unnecessary energy consumption. Two types of congestion can occur in wireless environment: Node-level congestion – it is caused by buffer overflow in the node. Link-level congestion –it occurs when wireless channels are shared by several nodes arising in collisions. Therefore, congestion in a network depends on the send data rate of nodes, buffers availability, max available bandwidth of network and node's density.

4.2.2.1 Determining the data send rate of honeynodes

Experiments are conducted to see the effect of congestion on Honeynodes' sniffing capabilities. In this work, we have set the initial congestion level of network by varying the data send rate of honeynodes only, keeping the other factors constant as given in Table 4.6.

Simulation Parameters

Table 4.6 Simulation Parameters for Data Rate determination

Available Queue Length	Available bandwidth	Node's density	Data-send Rate
50 Packets	2 Mbps	45 nodes in 1000x1000 meter sq. region	Varied from 8-64 kbps

This experiment gives the data send rate to be used by honeynodes.

4.3 Logging Module

The simulation of monitoring and logging module is done in NS2 at DSRAgent.cc file. In NS2, each node has DSRAgent attacked to it. We have added the promiscuous mode logging functionality by adding tapping functionality to DSRAgent. The data from MAC Layer (listened in promiscuous mode) is sent directly to the tap function in DSRAgent and is logged as given in the Table 4.7.

Table 4.7 Data Logging Trace Format

HoneyMANET trace format

```
7 [packet path: 6 7] r 1.019893612  2 cbr  [17 6] ------- [ 6 :0 7 :0] OR
7 [packet path: 6 7] r 1.019893612  2 dsr  [17 6] ------- [ 6 :0 7 :0] [RREQ/RR/RERR]
```

Node_id

[path in source routing header]

event_type (receive- r, sent- s, drop- D, forward- f, other's-data- o)

Scheduler::instance().clock()

unique_id (every packet generated at source is assigned a unique id)

traffic_type

[destination_MAC_Address   source_MAC_Address]

[IP_Source : Source_port   IP_Destination :Destination_port]

[Type of DSR Packet: Route Request (RREQ), Route Reply (RR) or Route Error (RERR)]

4.4 Integration Module

Data integration process is implemented using awk [30].

1. Firstly, packets are arranged in the sequential order, to represent the order in which they moved (generated, forwarded or received) in the network. In simulation, the rearrangement is done by sorting the packets according to their unique ids. (Whenever a packet is generated in the network, it is assigned a unique id by the simulator and it remains same while it's being forwarded or received). In real time scenario, we can either use the time of logging of packets, or can use the set (source-add, destination-add, packet-sequence-number) to identify the sequence of packets generated.

2. Second step is to remove the duplicate. Duplicate packets having similar unique ids and same last hop entry at MAC layer are removed.

Calculating percentage of traffic sniffed - The integrated database is compared with the NS2 default trace file to calculate the percentage of traffic sniffed by Honeynodes.

4.5 Attack Simulation

In our present work, we have used the rule base for identifying DOS attacks at network Layer – Simple Packet dropping attack, Black Hole Attack, Route Request Flooding Attack, Sleep Deprivation Attack and Grey Hole Attack. The simulations of Simple Packet dropping attack, Black Hole Attack, and Route Request Flooding Attacks are done (in DSRAgent.cc file of attackers) to validate the efficiency of HoneyMANET's Attack Detection Model.

4.5.1 Simple Packet Dropping Attack:

In this attack, the function in DSRAgent.cc whenever a forwarding packet is received, forwarding packets are dropped maliciously by the attacker and the packets are freed.

4.5.2 Black Hole Attack:

In simulations, whenever a malicious node receives the RREQ packet, it will immediately send a false RREP packet, showing that it has a closest route to the destination. It fills the next two entries in the Source routing header by its IP address and destination IP address, creates a RREP packet with a modified higher sequence number and sends back to the source. The source node assumes that malicious node has the fresh route towards the destination, ignores the RREP packet received from other nodes and begins to send the data packets over malicious node. This way, the malicious node attracts all the routes towards itself and then drops the data packets in receive function of DSR Agent.

4.5.3 RREQ Flooding attack:

In our present simulation, an attacker node violates the above rules to exhaust the network resource. The attacker node resends excessive RREQ without waiting for the RREP or round-trip time and doesn't wait for back-off period, rejecting the max allowed RREQ_RATELIMIT within per second. Moreover, the TTL of RREQ is set up to a maximum without using expanding ring search method.

The detection efficiency is tested by varying parameters like the number of malicious nodes present, type of attack (Local to a zone or globalised), Type of DOS attack launched and the time at which an attack is launched as given in Table 4.8.

Table 4.8 Simulated Attack Scenarios

<table>
<tr><td>Simulation Time</td><td colspan="11">1200 Seconds</td></tr>
<tr><td>Dos Attacks</td><td colspan="11">Flooding Attack, Simple Packet Dropping Attack, Black Hole Attack</td></tr>
<tr><td>Attack Type</td><td colspan="11">Localized Attack in Zone 1</td></tr>
<tr><td>Attacker</td><td colspan="3">One</td><td colspan="3">Two</td><td colspan="2">Three</td><td colspan="2">Four</td><td>Five</td></tr>
<tr><td>Start Attacking at time (sec)</td><td colspan="3">200</td><td colspan="3">400</td><td colspan="2">600</td><td colspan="2">800</td><td>1000</td></tr>
<tr><td>Attack Type</td><td colspan="11">Global Attack</td></tr>
<tr><td>Attacker</td><td>One</td><td>Two</td><td colspan="2">Three</td><td>Four</td><td colspan="2">Five</td><td>----------------------</td><td colspan="3" rowspan="2">Twenty three
1150</td></tr>
<tr><td>Start Attacking at time (sec)</td><td>50</td><td>100</td><td colspan="2">150</td><td>200</td><td colspan="2">250</td><td>----------------------</td></tr>
</table>

4.6 Evaluation Criterion

Intrusion Detection Efficiency of Anomaly and Misuse based Modules is estimated using following statistics:

1. **Detection Rate:** Ratio of attacks detected to total attacks launched is calculated to find the attack detection rate of HoneyMANET.
2. **False Alarm Rate** is estimated by calculating the ratio of false alarms (alarm generated when no attack was happening) generated to total no attacks instants.

HoneyMANET not only detects attack, but also help in evaluating the local and global impact of different localized and globally distributed attacks on different regions of the network. The impact of different attacks on network is evaluated by honenyMANET in terms of following parameters:

1. **Packet Delivery Ratio of Network:** The ratio of the data packets delivered to the destinations to those generated by the CBR sources, i.e., Packet delivery ratio = Received Packets / Sent Packets.
2. **Packet Drop Ratio of network nodes:** The ratio of the forwarding data packets dropped by a node to those received by that node to forward = (Packet received by node to forward – packets forwarded by node) / Packet received by node to forward
3. **Send Ratio or send bandwidth or Bandwidth available to nodes:** Ratio of packets send by a node in the network to those sent by the CBR application running on that node = Actual packets sending rate to network / Packet Generation rate of Application Layer
4. **Global Throughput**: Rate at which traffic is received at the destination nodes.
5. **Global Loss rate** = Rate at which traffic is dropped in the network.

CHAPTER 5
RESULTS AND DISCUSSION

The honeynodes at any time are involved in several security related functions – measuring the data to be transmitted, relaying data from other sources, sniffing and aggregating data from multiple sources and intrusion detection analysis. A notion of priority among the different tasks (surveillance, relaying, deception and intrusion detection) to be served may have to be developed before making any decision.

5.1 Decision regarding Free-movement zones

Requirement 1: Finding the free-movement zone that helps in making robust IDS.
Local intrusion detection requires network to be divided into zones, so that an overview of network activities at specific regions can be obtained and mitigated there only. Average sized zones like 5 zones scenario or 9 zones scenario are better suited for intrusion detection.

Requirement 2: Finding the free-movement zone that shows more real-world scenario.
The All-in-one-scenario where all nodes move freely in all the 1000x1000 meter sq region depicts more of a real world scenario. As the movements get restricted to more and more limited area, the realism of situation loses.

Requirement 3: Finding the free-movement zone that helps in relaying.
With the increase in restriction of honeynode's movement to more limited area, keeping track of honeynodes location with time becomes easy. The randomness of network decreases and so do their link changes- keeping the network of honeynodes stable and making communication easy and cost effective. The honeynode mesh type of scenario is best for fast and efficient relaying.

Requirement 4: Finding the free-movement zone that gives high area coverage and better traffic sniffing with minimal number of honeynodes.

Table 5.1 gives the percentage of traffic sniffed by honeynodes in different free-movement zones scenarios as the number of honeynodes are varied in server-client type of application model. Server being a Honeynode.

Table 5.1 Effect of free movement zone on percentage sniffing

Number of Honeynodes in each region/zone	9	18	27	36	45
All-in-one-scenario	85.81%	95.88%	96.84%	97.15%	97.42%
Number of Honeynodes in each region/zone	2	4	5	7	9
5 zones-scenario	84.81%	96.35%	96.87%	97.02%	97.76%
Number of Honeynodes in each region/zone	1	2	3	4	5
9 zones-scenario	82.87%	96.63%	97.93%	97.63%	97.90%
16-regions/zones with Honeynode-mesh-scenario (each region having one fix honeynode at center) 96.04%					

Figure 5.1 shows that in all different zone scenarios, the effect of variation in number of honeynodes on percentage of traffic sniffed is almost same. The minimal number of nodes required for maximum area coverage and high enough traffic sniffing is from 16-20 depending on zone- 18 nodes in all-in-one-scenario, 20 in 5-regions scenario, 18 in 9-region scenario and 16 for honeynodes mesh to give as high as 95-97% traffic sniffing coverage.

Decision: Keeping in consideration different requirements of - high realism, good relaying and effective area coverage with minimal number of nodes and zone requirement for intrusion detection- we have decided upon 5-zones scenario for HoneyMANET setup.

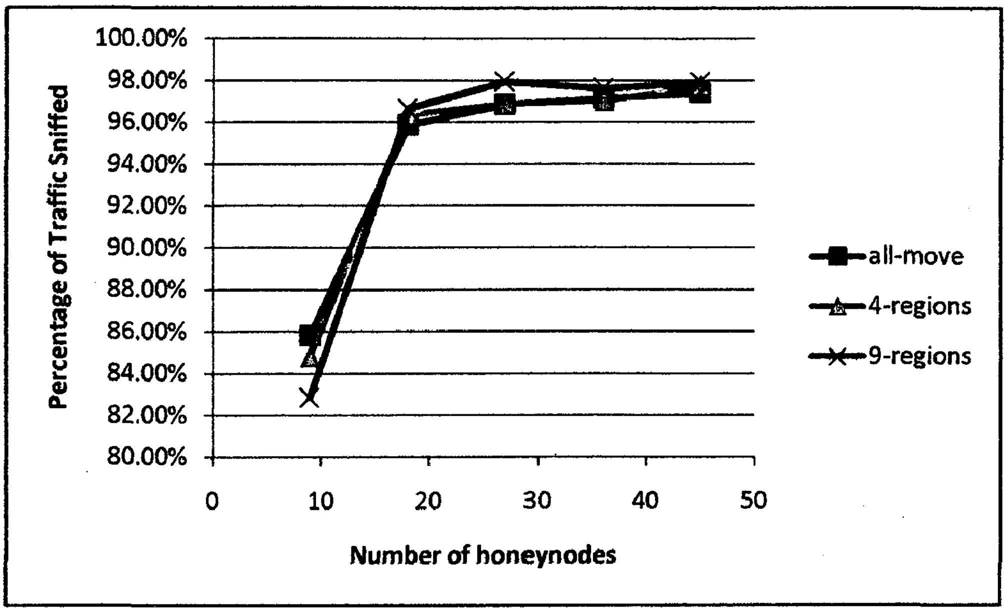

Figure 5.1 Effect of Free Movement Zones Percentage traffic sniffed

5.2 Decision regarding Speed and Number of Honeynodes

Requirement 1: Finding the speed that helps in making robust IDS.

Intrusion detection requires network packets to be sniffed properly. Speed that favors high percentage of packet sniffing will be apt for good intrusion detection.

Requirement 2: Finding the speed that shows more real-world scenario.

Randomly chosen speeds varying from 10 to 50 m/sec depicts real world situation.

Requirement 3: Finding the speed favoring relaying.

With the increase in nodes speed the dynamism of network increases and so do their link changes- making the network of honeynodes unstable. The mesh of fix Honeynodes is best for fast and efficient relaying.

Requirement 4: Finding the speed that gives better traffic sniffing with minimal number of honeynodes.

Table 5.2 gives the percentage of traffic sniffed by honeynodes in 5 zones topology as the number and speed of honeynodes are varied in server-client type of application model. Server being a Honeynode.

Table 5.2 Effect of Speed on Percentage traffic sniffed

5-zones Topology	10 m/sec	30m/sec	50m/sec
45	97.99%	97.61%	95.277%
35	96.58%	94.24%	91.58%
25	95.65%	91.98%	80.43%
20	94.02%	87.66%	78.71%
10	84.81%	69.77%	68.05%

Figure 5.2 shows that speed has more effect on sniffing percentage when the number of honyenodes in each zone is less. As the number increases, the speed doesn't have much effect on sniffing percentage.

Decision: Keeping in consideration different requirements of - high realism, good relaying and effective traffic sniffing with minimal number of nodes- we have decided upon 10m/sec speed for honeynodes.

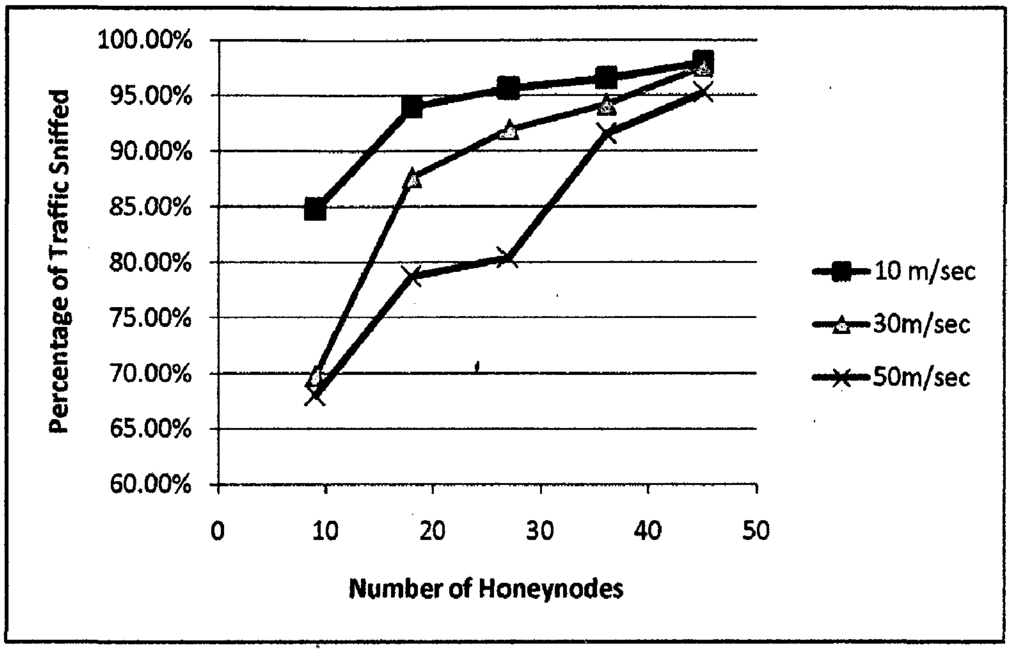

Figure 5.2 Effect of Speed on percentage traffic sniffed by honeyndoes

5.3 Decision regarding the Data Send Rate of Honeynodes

Requirement 1: Finding the data send rate that shows more real-world scenario.

Randomely chosen data send rate between 8 to 64 kbps with maximum available bandwidth of 2 mbps in mobile environment depicts real world situation.

Requirement 2: Finding the data send rate favoring relaying.

With the increase in congestion in network, packet loss in the network increases and affects the relaying process. The minimal rate of 8kbps is best suited for fast and efficient relaying.

Requirement 3: Finding the data send rate that gives better traffic sniffing.

Simulation Results

Table 5.3 shows effect of data send rate of Honeynode on percentage of traffic sniffed. Foreign nodes send data at varing rates from 8 to 64 kbps.

Table 5.3 Effect of data send on percentage traffic sniffed

Data Send Rate	8kbps	16kbps	32kbps	64kbps
Percentage Traffic sniffed	94.40%	87.18%	76.31%	63.68%

Figure 5.3 shows that the network congestion has more drastic effect on sensing capabilities then mobility of nodes. As the data send ate increases, congestion increases and percentage of sniffing drops from 97% to 63%.

Decision: Keeping in consideration different requirements of - high realism, good relaying and effective traffic sniffing with minimal number of nodes- we have decided upon minimum 8kbps data send rate for honeynodes.

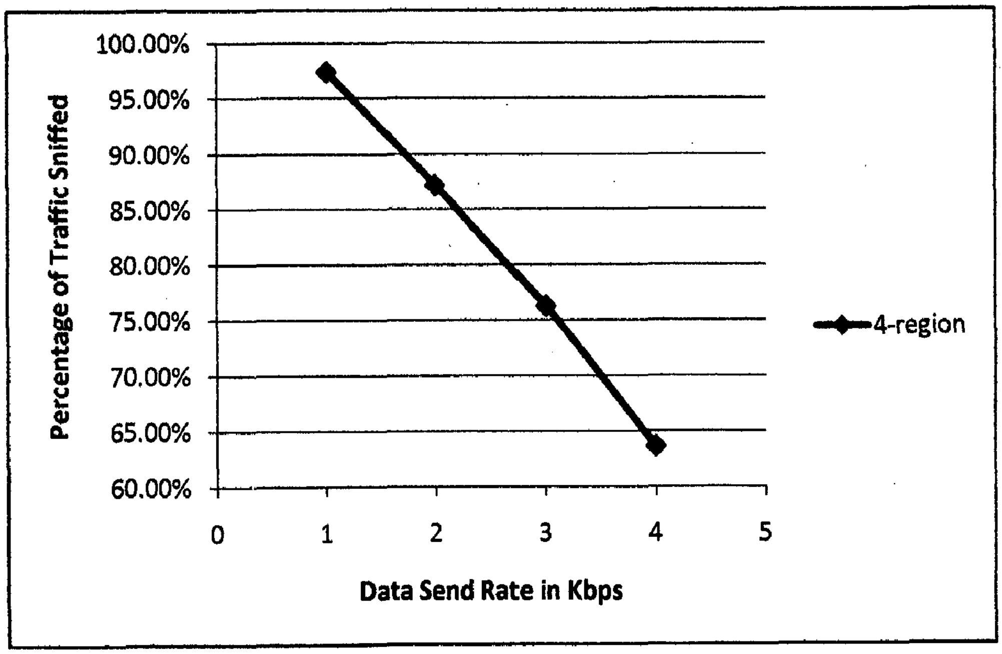

Figure 5.3 Effect of data send rate on percentage traffic sniffed by honeynodes

5.4 Summary of Design Parameters of HoneyMANET

Keeping in consideration different requirements of – good surveillance, high realism, good relaying and robust intrusion detection, the design parameters used for HoneyMANET's set up are summarised in Table 5.4.

Table 5.4 Design Parameters of HoneyMANET

Real World Scenario to emulate	Random Waypoint Mobility model
Type of Application to deploy	Client-Server over UDP connection
Area under survellience and the area of free-movement zones	1000x1000 m sq region with 5 free-movement zones of 500x500 m sq
Number of honeyndes for sniffing in each zone	4 honeynodes in each zone- total 20
Speed and pause-time of Honeynodes	10m/sec and 2.0 sec
Traffic generation by Honeynodes	CBR traffic at constant rate of 8kbps

5.5 Anomaly Detection Results

Three kinds of Denial of Service Attackers (Simple Packet Drop Attack, Black Hole Attack and RREQ Flooding Attack) are launched on HoneyMANET environment using NS2 simulations and the anomaly detection efficiency of HoneyMANET is tested.

Table 5.5 gives the time at which different attackers start launching the local attack in zone 1.

Table 5.5 Simulated Localised Attack Scenarios

Attack Type	Localized Attack in Zone 1					
Attacker	Zero	One	Two	Three	Four	Five
Start Attacking at time (sec)	0	200	400	600	800	1000

Table 5.6 gives the time at which different attackers start launching the attack globally.

Table 5.6 Simulated Global Attack Scenarios

Attack Type	Global Black Hole Attack							
Attacker	Zero	One	Two	Three	Four	Five	---------	Twenty Three
Start Attacking at time (sec)	0	50	100	150	200	250	---------	1150

Figure 5.4 and Figure 5.5, show the detection rate and false alarm rate of Anomaly Detection Module with time, respectively. P1, P2 and P3 represent the detection and false alarm rate as P value is changed.

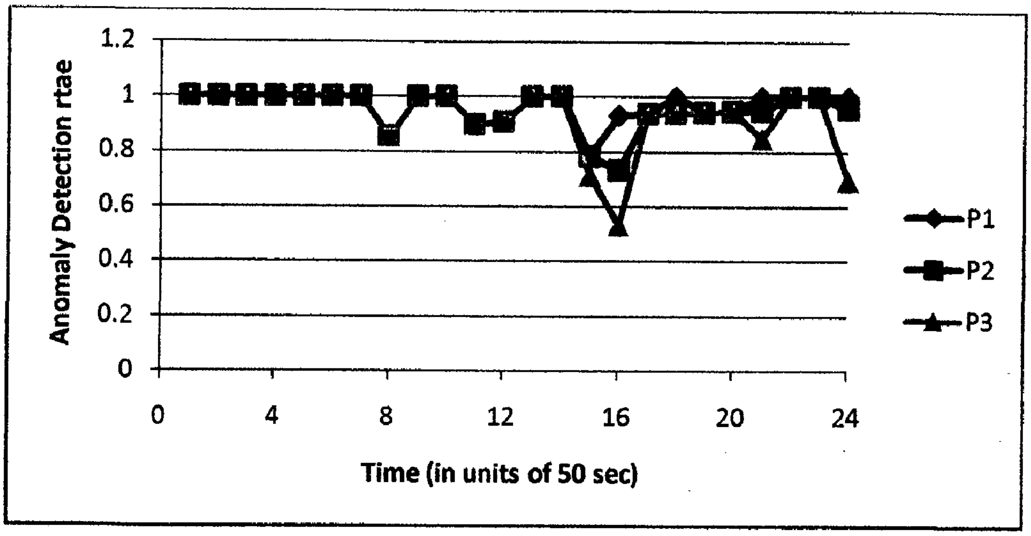

Figure 5.4 Anomaly Detection rate

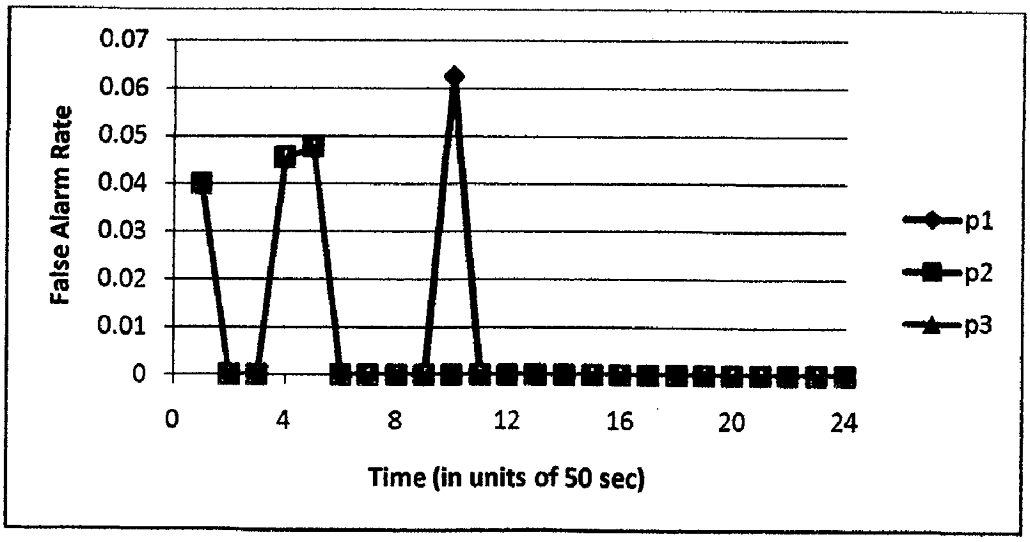

Figure 5.5 False Alarm rate

In general, low P values gives low false alarm rate and as the P value is increased detection rate increases. The result shows that the Anomaly Detection Module has high attack detection rate value around 1 (even at low P value) and also has appreciable low false alarm rate of 0 (even at high P value), independent of number of attackers.

5.6 Misuse Detection Results

5.6.1 Black Hole Attack

The rule base use to identify black hole attack is shown in Table 5.7. It is the deviation in Packet drop ratio and packets received by a node to forward that reflects the presence of black hole attack in network. Table 5.7 also shows the result of signature based attack identification Module as black hole attack is launched.

Table 5.7 Result of Black Hole Detection

Identification Rule for Black Hole	Node's Packet drop ratio >> Expected && Packets received by node to forward – PRF >> Expected
Attackers detected	100 %
Identified as Simple Packet Drop Attacker	40 %
Identified as Black Hole Attacker	60 %
False Alarm Rate	0

5.6.2 Simple Packet Dropping Attack

The rule base use to identify simple packet drop attack is shown in Table 5.8. It is the deviation in packet drop ratio of a node that reflects the presence of simple drop attack in network. Table 5.8 shows the result of signature based attack identification Module.

Table 5.8 Result of Simple Packet Drop Attack Detection

Identification Rule for simple packet drop attack	Node's Packet drop ratio >> Expected && Packets received by node to forward – PRF ~ Expected
Attackers detected	100 %

Identified as Simple Packet Drop Attacker	90 %
Identified as Black Hole Attacker	10 %
False Alarm Rate	0

5.6.3 RREQ Flooding Attack

The rule base use to identify RREQ Flooding Attack is shown in Table 5.9. When the number of Route Request sent by a node is very high than the expected value, it is seen as flooding attack. Table 5.9 also shows the result of signature based attack identification Module as RREQ Flooding Attack is launched.

Table 5.9 Result of RREQ Flooding Attack Detection

Identification Rule for Flooding Attack	Node's Route Request send ratio >> Expected
Attackers detected	100 %
Identified as Flooding Attack	100 %
False Alarm Rate	0

5.7 Discussion

Reason for Misidentification of some black hole attackers as simple packet drop attackers and vice versa by signature based module, can be determined by observing the packet drop ratio and packet received to forward graphs of different nodes in different scenarios (as observed by HoneyMANET).

Figure 5.6 shows the packet drop ratio of nodes of Zone 1 with time when local balck hole attack is launched. Figure 5.7 and Figure 5.8, show the change in route requests send and packets received by nodes to forward as the time and number of attackers increases in localized black hole attack.

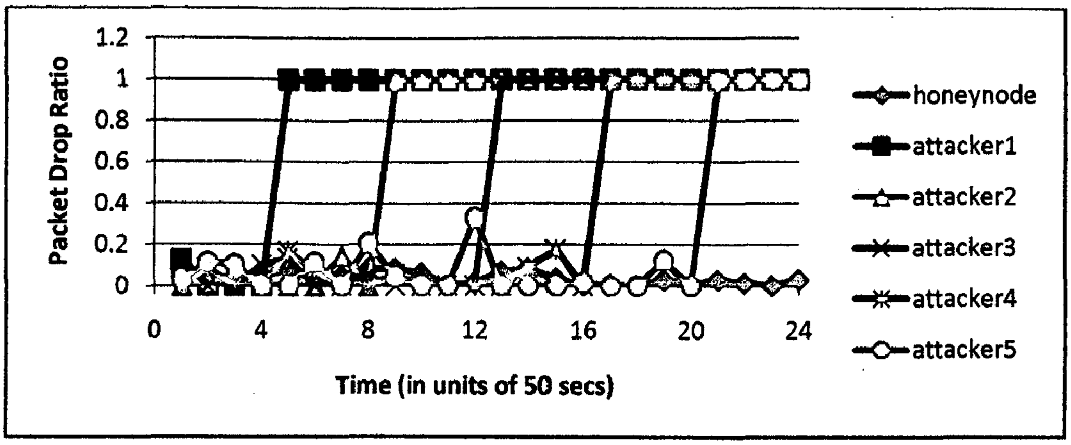

Figure 5.6 Packet Drop Ratio in Localised Black hole Attack (as observed by HoneyMANET)

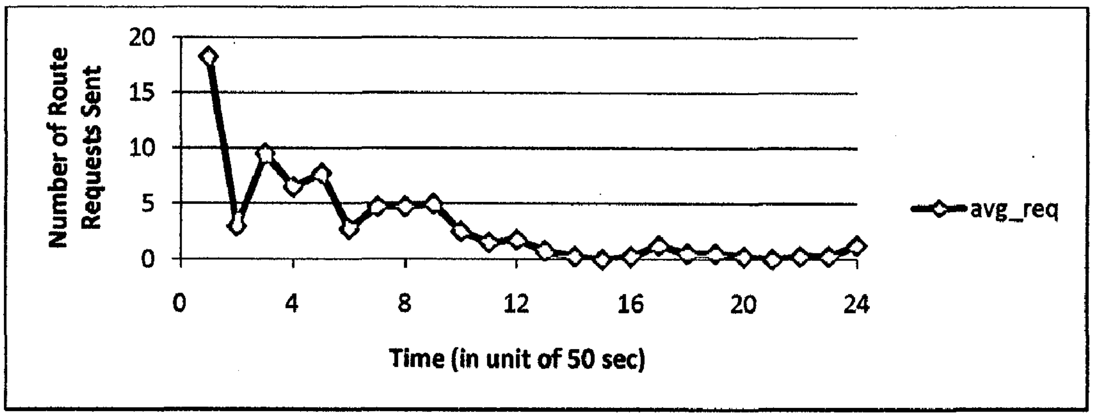

Figure 5.7 Route Requests sent in Zone 1 in Localised Black hole Attack (as observed by HoneyMANET)

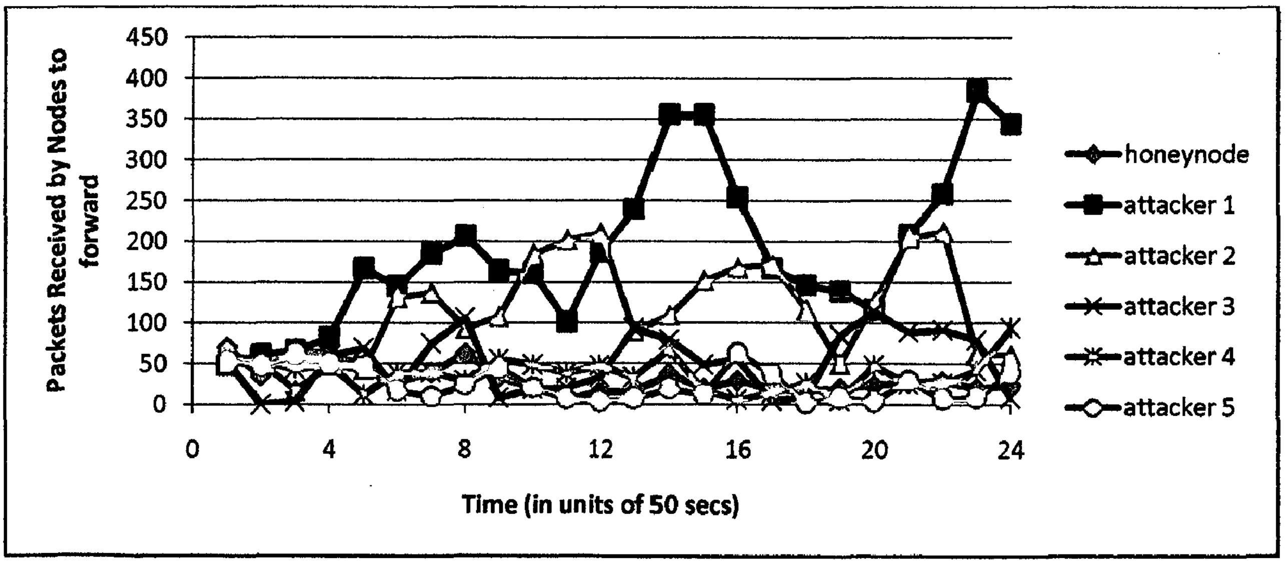

Figure 5.8 Packets Received by Zone 1 nodes to forward in Localised Black hole Attack (as observed by HoneyMANET)

Figure 5.9 and Figure 5.10 shows the packet drop ratio and packets received by nodes to forward by nodes of Zone 1 with time when simple packet drop attack is launched.

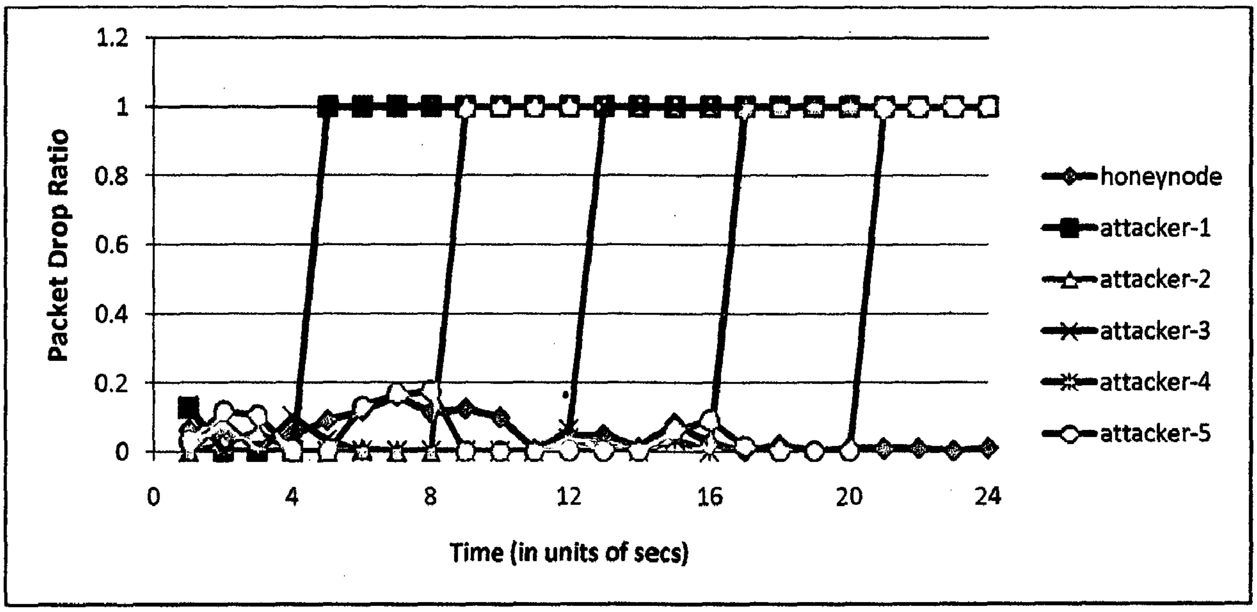

Figure 5.9 Packet Drop Ratio of zone 1 in Simple Packet Drop Attack (as observed by HoneyMANET)

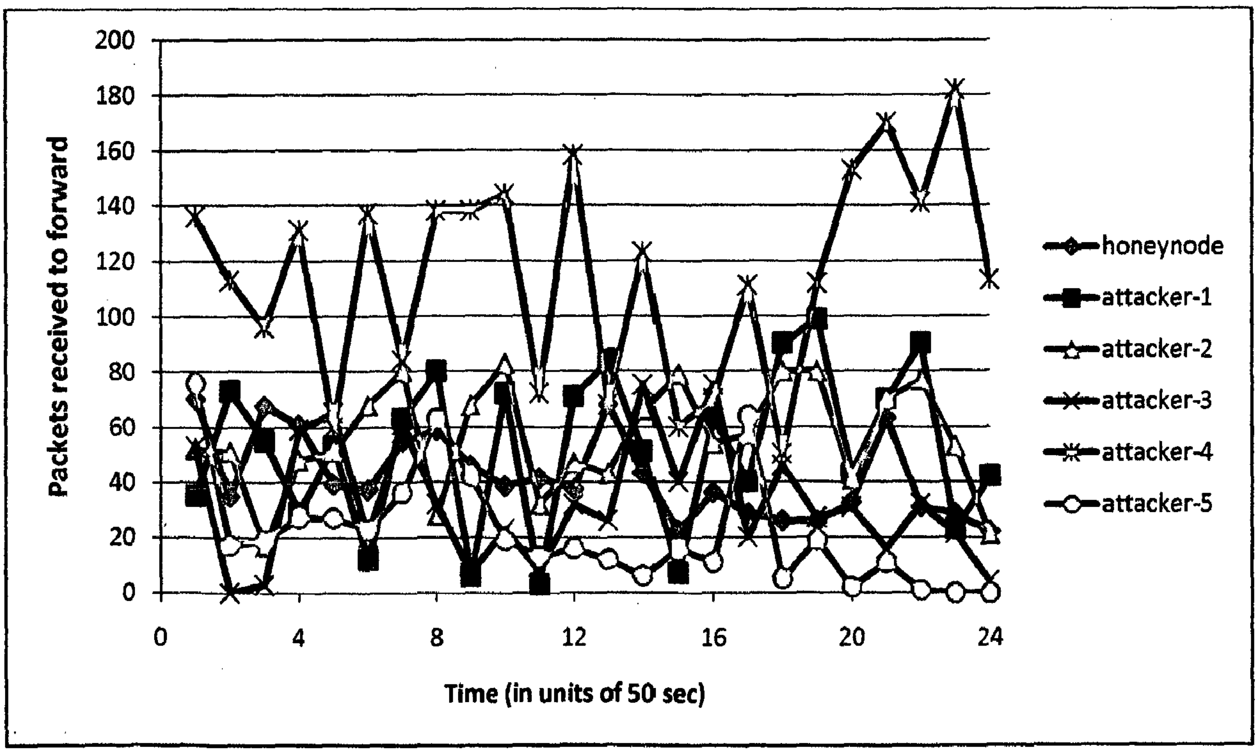

Figure 5.10 Packets received to forward by zone 1 nodes in Simple Packet Drop Attack (as observed by HoneyMANET)

Figure 5.11 and figure 5.12 show the packet drop ratio of attackers in case of global black hole attack and global simple packet drop attack as their number increases in the network globally with time. As number of attackers' increases, average packet drop rate of attackers increases in both the cases. In simple drop attack initially, till 5 attackers, drop is almost stable. In black hole attack, it is steeper at lower values of attackers also. Almost both attacks show the same drop rate.

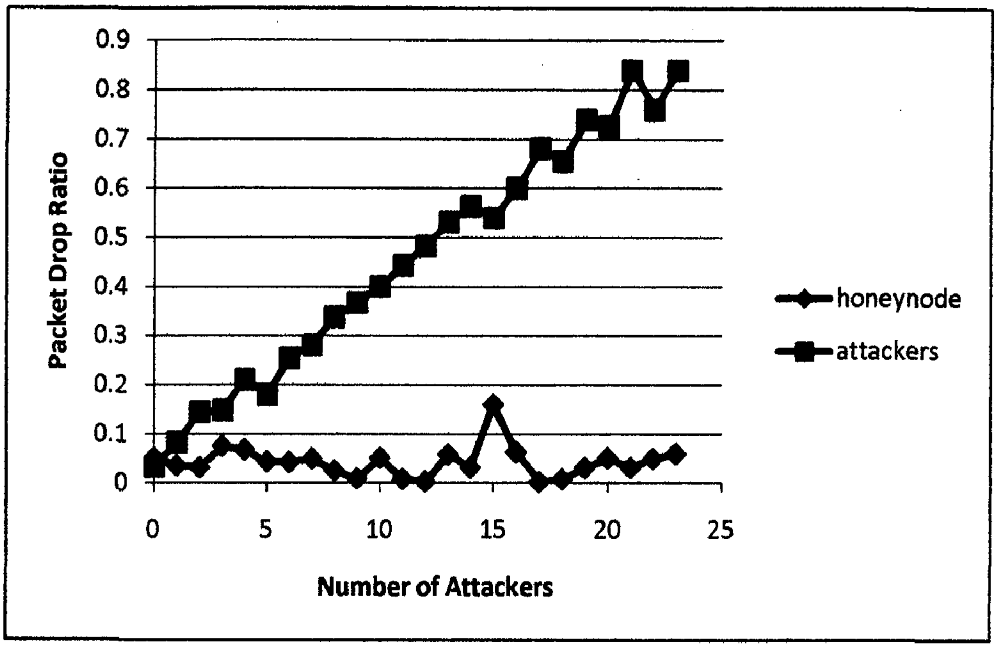

Figure 5.11 Global Packet Drop Ratio as Global Black hole is launched

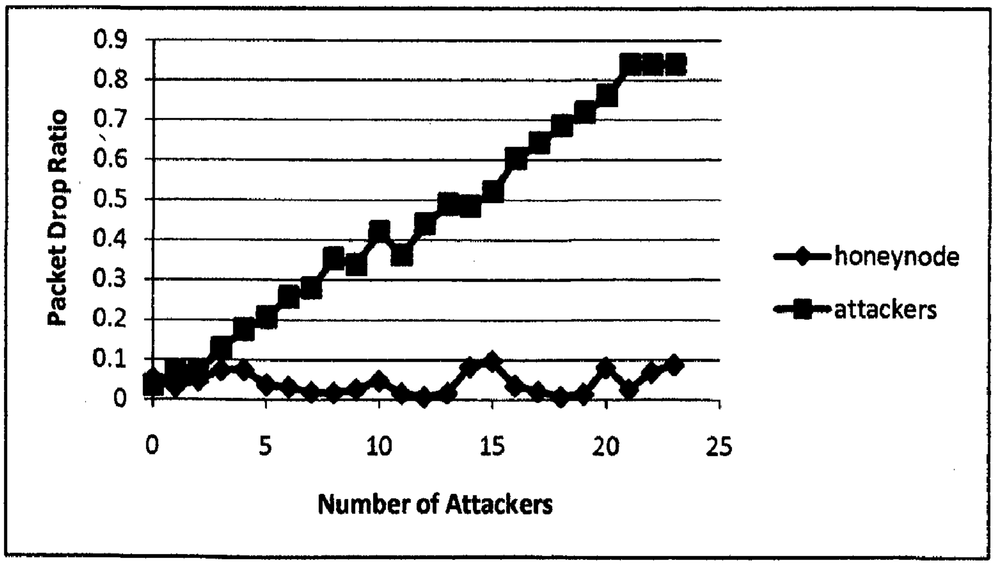

Figure 5.12 Global Packet Drop Ratio as Global Simple Packet drop attack is launched

Figure 5.13 and figure 5.14, show the number of packet received by nodes to forward in case of global black hole attack and global simple packet drop attack as their number increases in the network globally with time. In case of Black hole attack, as the number of attackers' increases, more and more traffic get concentrated towards attackers. In simple drop attack, as number of attackers increases, the network traffic decreases and so does the packets to be forwarded by them.

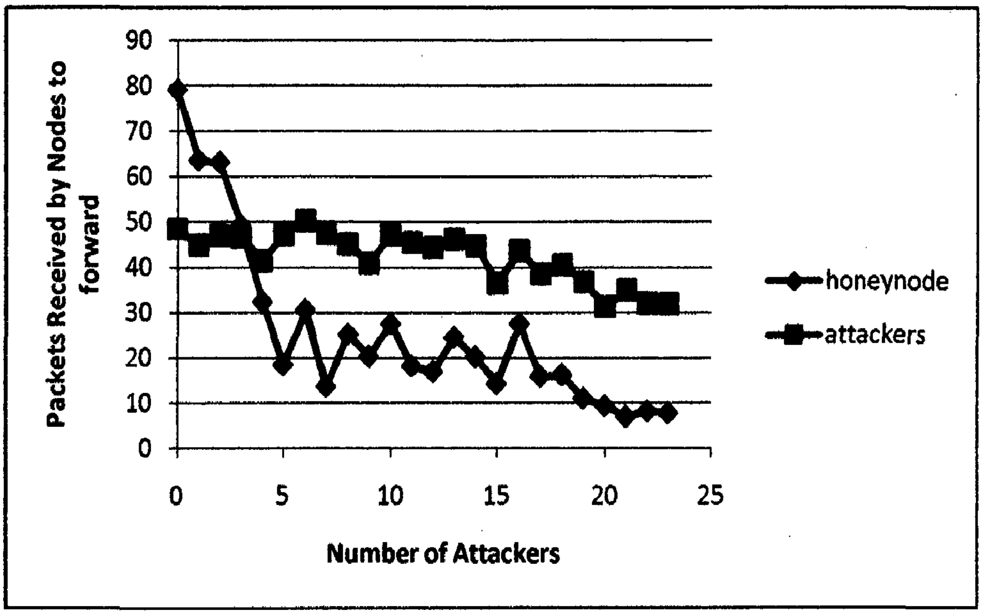

Figure 5.13 Packets received by nodes globally as global black hole attack is launched

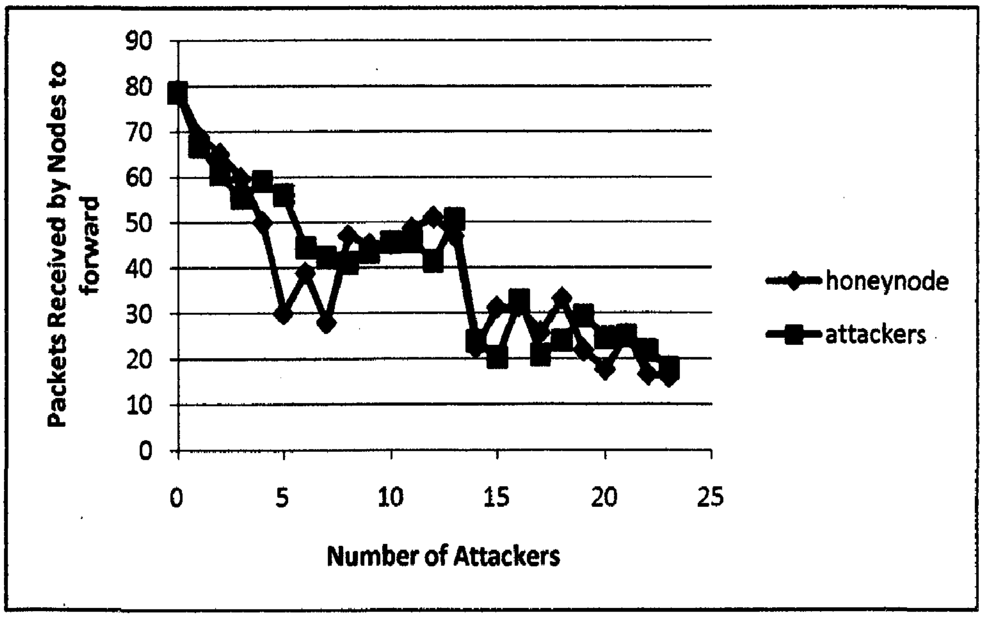

Figure 5.14 Packets received by nodes globally simple packet drop attack is launched

The graphs indicate that the attackers that launched the black hole attack earlier are able to attract a large amount of traffic towards them. However, as the time increases, the route requests decreases, NO new routes are initiated and so most of the traffic kept flowing through the earlier route created by attackers. Therefore, the attackers that launched the black hole attack later are identified as simple packet drop attacks and not as black hole attack by Attack Identification Module.

Reason for misidentification of simple packet drop attack is that in network some nodes exist that actually have very good route to destination server. Therefore more than average traffic flows through them. These nodes when launch the simple packet drop attack got misclassified as black hole attackers by HoneyMANET's misuse detection module.

5.8 Summary of Attack Detection Module

Simulations are done to find the different design parameters of HoneyMANET – the free movement zone to use, number, speed and data generation rate of honeyMANET, to make HoneyMANET robust to detect intrusions in different attack scenarios: different types of attacks, launched at different times, at different regions – local to a zone or globally distributed, with varying rate and number of attackers.

In general ad-hoc networks, production nodes of the network are involved in decision making process for intrusion detection. Therefore, as the ratio of malicious nodes (Bad production nodes/ Total production nodes) increases, the detection rate of whole model decreases [31][32]. On the other hand, in HoneyMANET system, we have honeynodes for localised and global monitoring which are always 100% trusted and are separate from good/bad production nodes. Therefore, the attack detection efficiency remains unaffected by the presence of bad nodes. This also has been proved by simulation results which show that the detection efficiency of joint anomaly and misused based intrusion detection model of HoneyMANET is high and mostly remains at value 1, independent of type of attack or number of attackers in the network. The false alarm rate is also low (mostly remaining at value 0). This is a great achievement as compared to IDS of general ad-hoc network where detection rate decreases as number of attacker increases in the network.

Figure 5.15 shows in general how the detection rate is affected when IDS based solutions are implemented in general ad-hoc environment and of HoneyMANET's intrusion detection system implemented with its deceptive environment, as the ratio of bad nodes is increased. Simulation results show that HoneyMANET is a reliable, robust and efficient intrusion detection system.

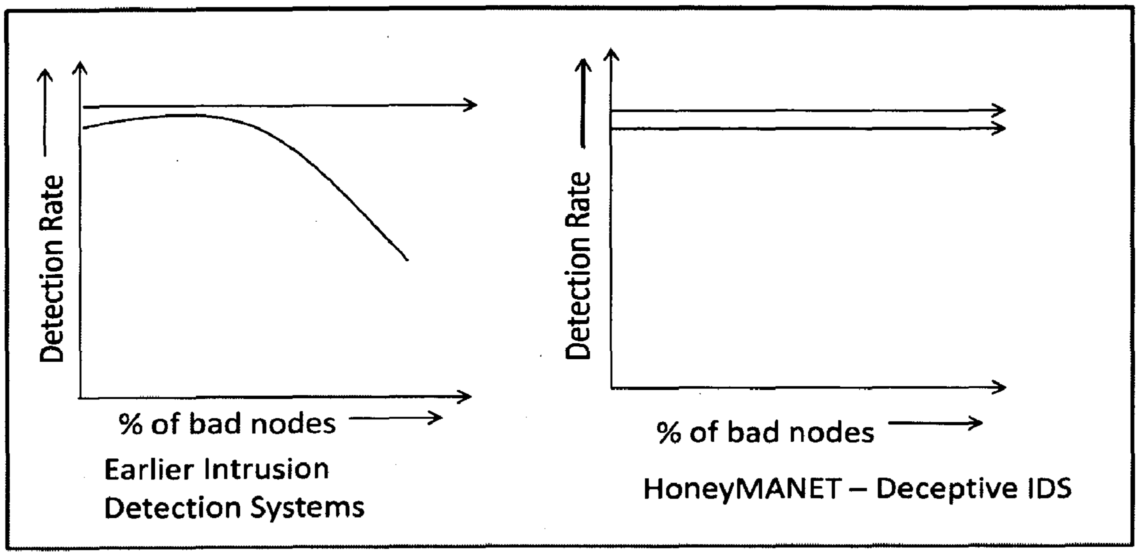

Figure 5.15 IDS in General ad-hoc network Vs. HoneyMANET's IDS in deceptive environment

CHAPTER 6 HONEYMANET FOR EVALUATION OF ATTACKS' IMPACT

The honeynodes at any time are involved in sniffing and aggregating data from multiple sources and intrusion detection analysis. The hierarchical HoneyMANET gives us a localised and global picture of network activities. It can help us to evaluate the impact of different attacks on network using the network traffic sniffed by honeynodes from different regions.

6.1 Impact of localized black hole and simple packet dropping attack

Figure 6.1 shows the effect of black hole and simple packet dropping attack on local region in terms of local delivery ratio as the numbers of attackers is increased in the region. The result shows that when the number of attackers is low, black hole brings down delivery ratio at much faster rate than simple packet drop attack. At high attackers' number value, both attacks have same effect on network.

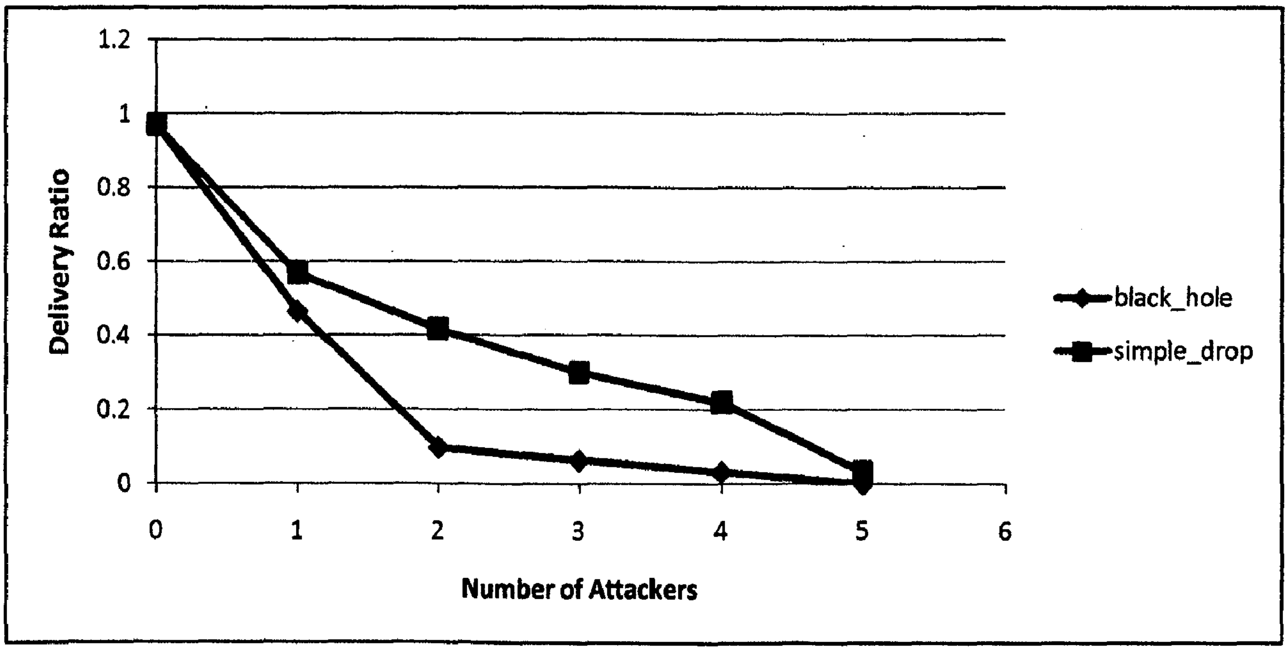

Figure 6.1 Delivery Ratio of Zone 1 when localized black hole and simple drop attacks are launched (as observed by HoneyMANET)

CHAPTER 6
HONEYMANET FOR EVALUATION OF ATTACKS' IMPACT

The honeynodes at any time are involved in sniffing and aggregating data from multiple sources and intrusion detection analysis. The hierarchical HoneyMANET gives us a localised and global picture of network activities. It can help us to evaluate the impact of different attacks on network using the network traffic sniffed by honeynodes from different regions.

6.1 Impact of localized black hole and simple packet dropping attack

Figure 6.1 shows the effect of black hole and simple packet dropping attack on local region in terms of local delivery ratio as the numbers of attackers is increased in the region. The result shows that when the number of attackers is low, black hole brings down delivery ratio at much faster rate than simple packet drop attack. At high attackers' number value, both attacks have same effect on network.

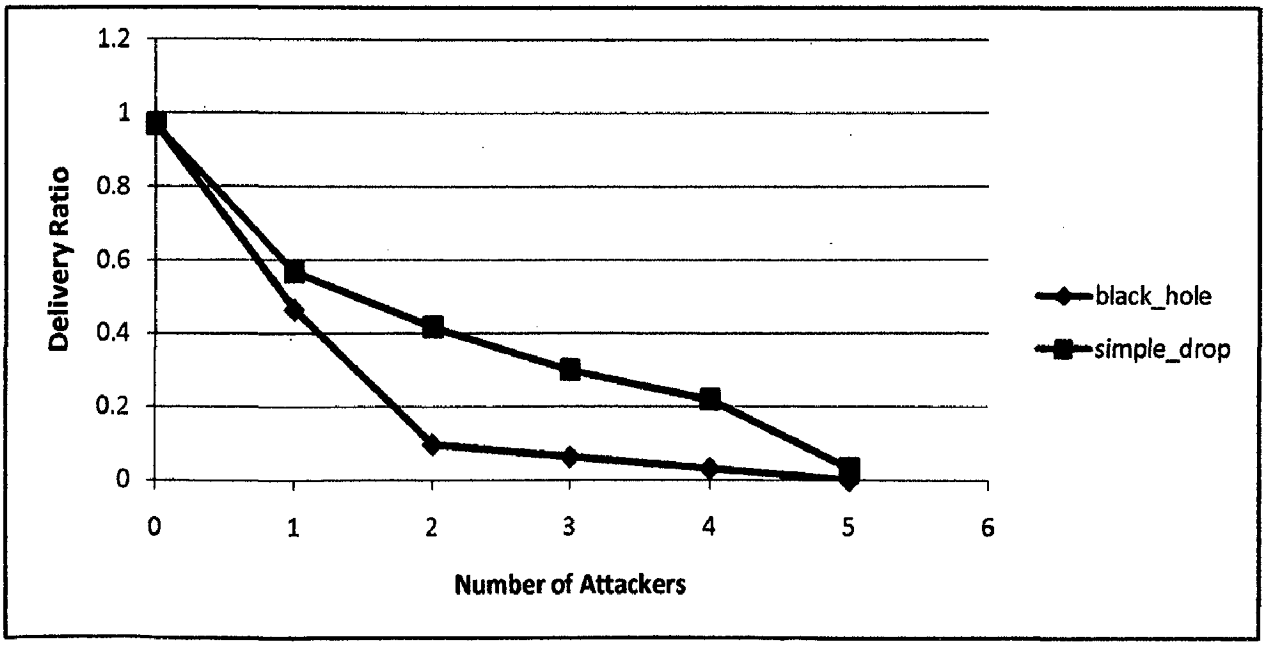

Figure 6.1 Delivery Ratio of Zone 1 when localized black hole and simple drop attacks are launched (as observed by HoneyMANET)

Figure 6.2 and 6.3, show the global impact of local black hole and simple packet dropping attack on data packet drop rate and throughput rate of whole network. The result shows that the impact of black hole attack is more than the simple packet drop attack.

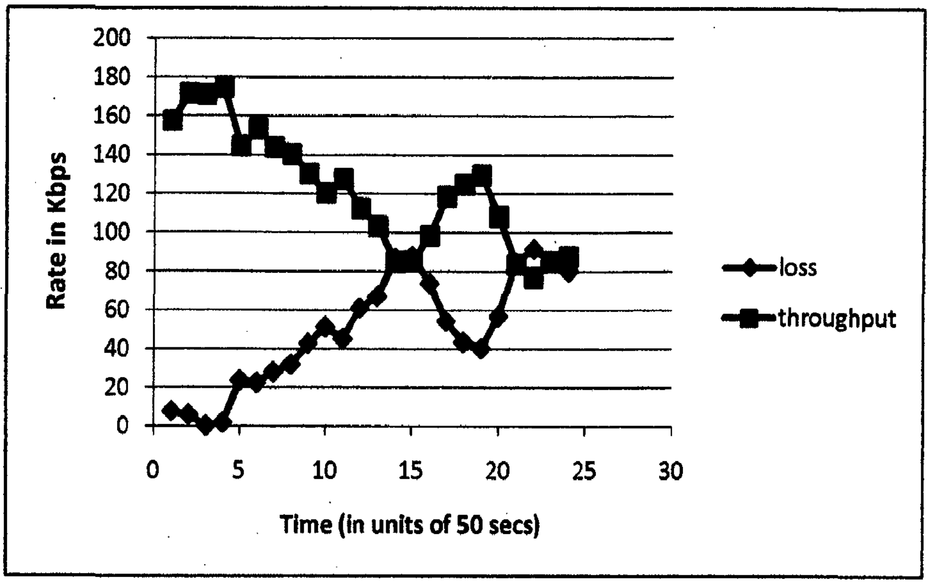

Figure 6.2 Global Throughput and Packet loss rate as Localised Black Hole is launched (as observed by HoneyMANET)

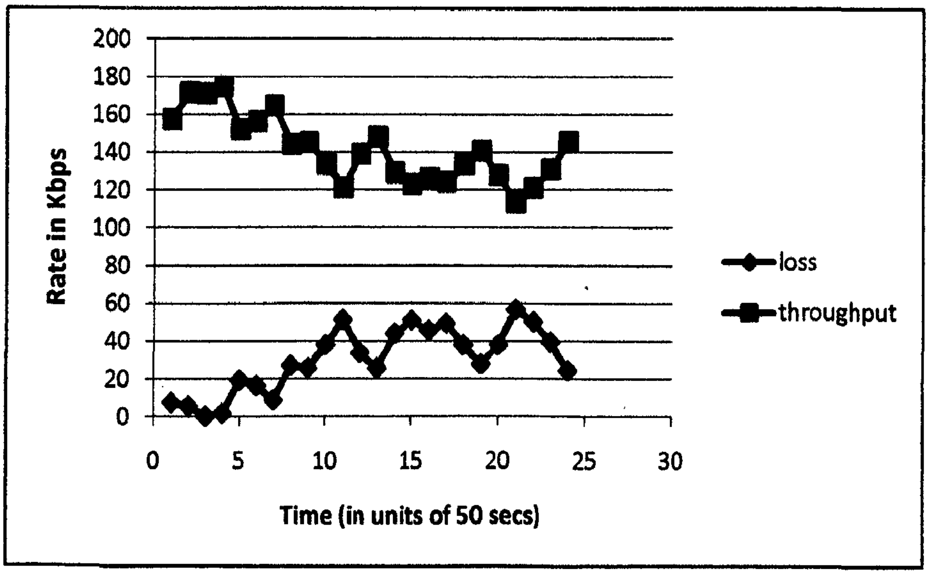

Figure 6.3 Global Throughput and Packet loss rate in Localised Simple Drop Attack (as observed by HoneyMANET)

Figure 6.4 and 6.5 show the global impact of local black hole and simple packet dropping attack on delivery ratio of whole network. The result shows that the impact of black hole attack is more than the simple packet drop attack.

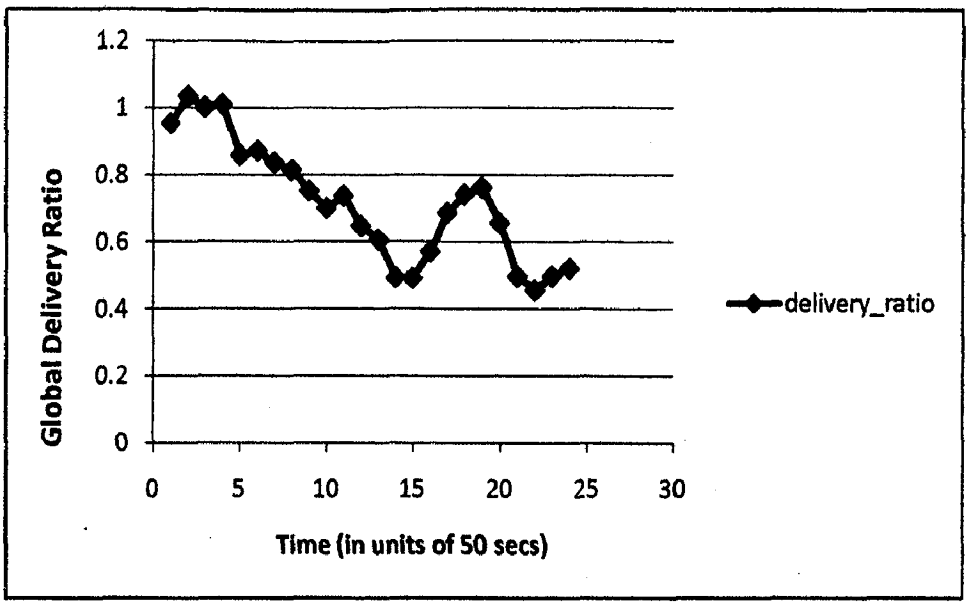

Figure 6.4 Impact of Localized balck hole attack on global delivery ratio (as observed by HoneyMANET)

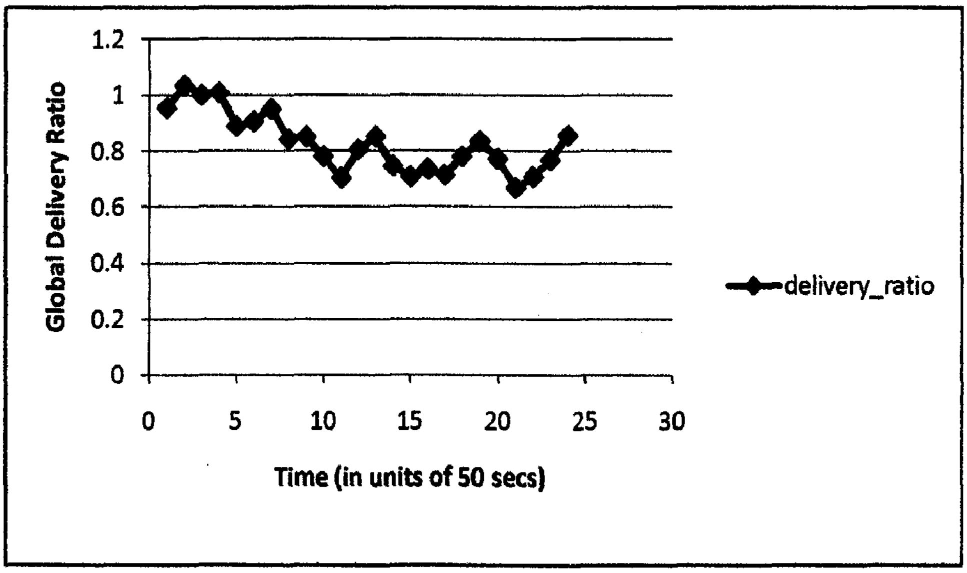

Figure 6.5 Impact of Localized Simple Packet Drop Attack on global delivery ratio (as observed by HoneyMANET)

6.2 Impact of Global black hole and simple packet drop attack

The global impact of global black hole and simple packet dropping attack on data packet drop rate and throughput rate of whole network is shown in figure 6.6 and 6.7, and on packet delivery ratio of whole network is shown in figure 6.8 and 6.9.

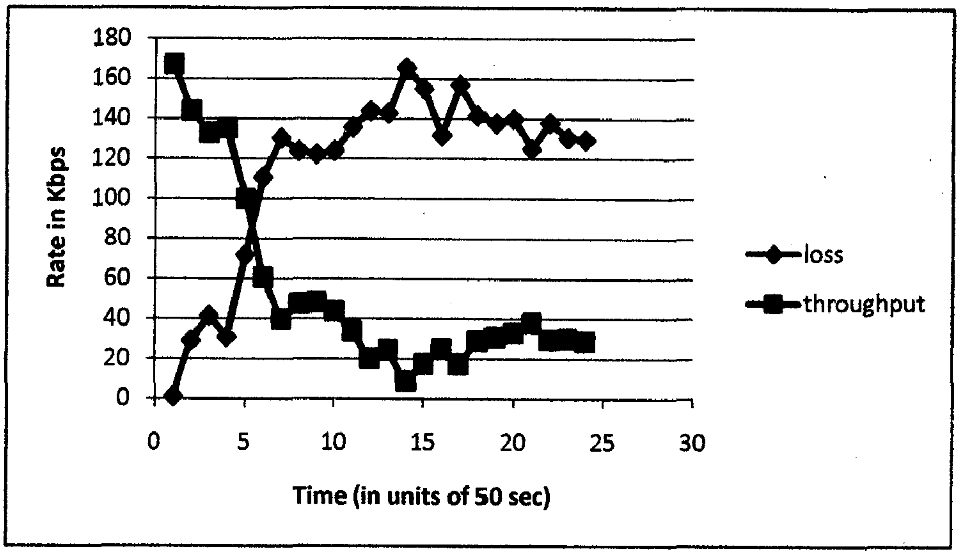

Figure 6.6 Global throughput and packet loss rate in global black hole attack (as observed by HoneyMANET)

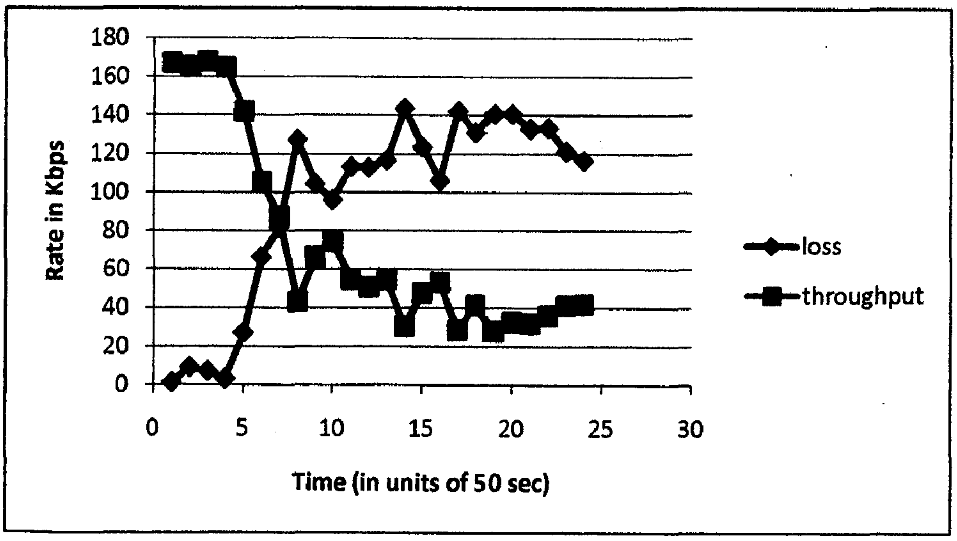

Figure 6.7 Global throughput and packet loss rate in global simple packet drop attack (as observed by HoneyMANET)

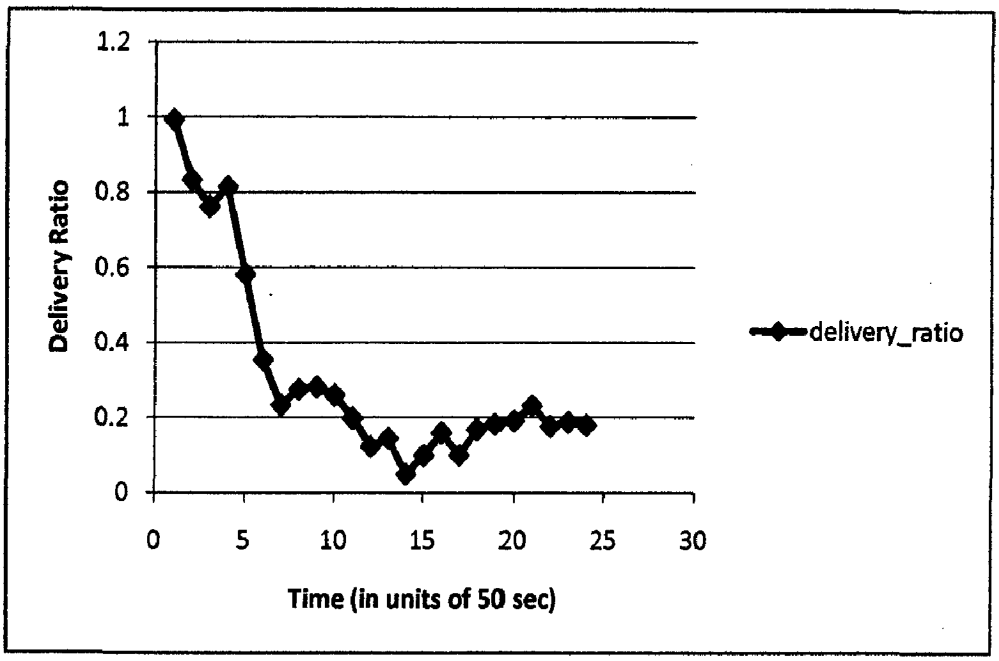

Figure 6.8 Global Delivery Ratio in Global Black Hole attack (as observed by HoneyMANET)

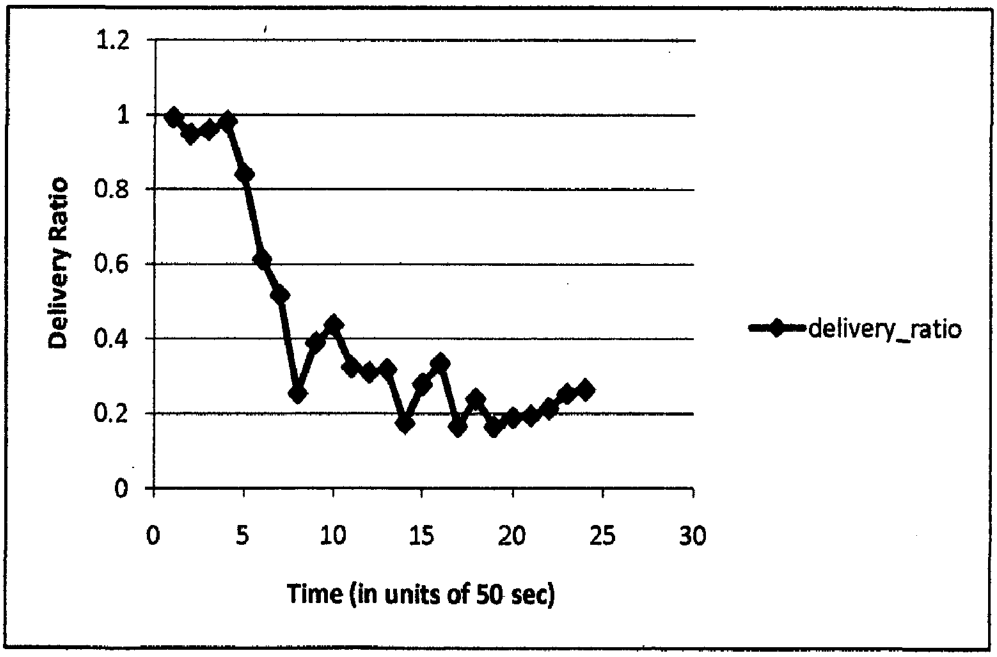

Figure 6.9 Global Delivery Ratio in Global Simple Packet drop attack (as observed by HoneyMANET)

The graphs show that the impact of black hole attack is slightly more than the simple packet drop attack when number of attackers is low. As the number of attackers increases with time, both attacks bring down the network almost by same percentage.

6.3 Impact of Flooding attack

Figure 6.10 shows the impact of local flooding attack on send ratio, drop ratio, and delivery ratio of local region, when number of attacker is one and the attack rate increases from 8 kbps to 2.4 mbps. Figure 6.11 shows the impact of local flooding attack on send ratio, drop ratio, and delivery ratio of local region 1 when number of attackers increases from 0 to 4 at attack rate of 16 kbps (drop-ratio(16), send-ratio (16) and delivery-ratio (16)) and 64 kbps (drop-ratio, send-ratio and delivery-ratio series in graph). The graphs show that attack rate has more drastic effect on network performance than the number of attackers.

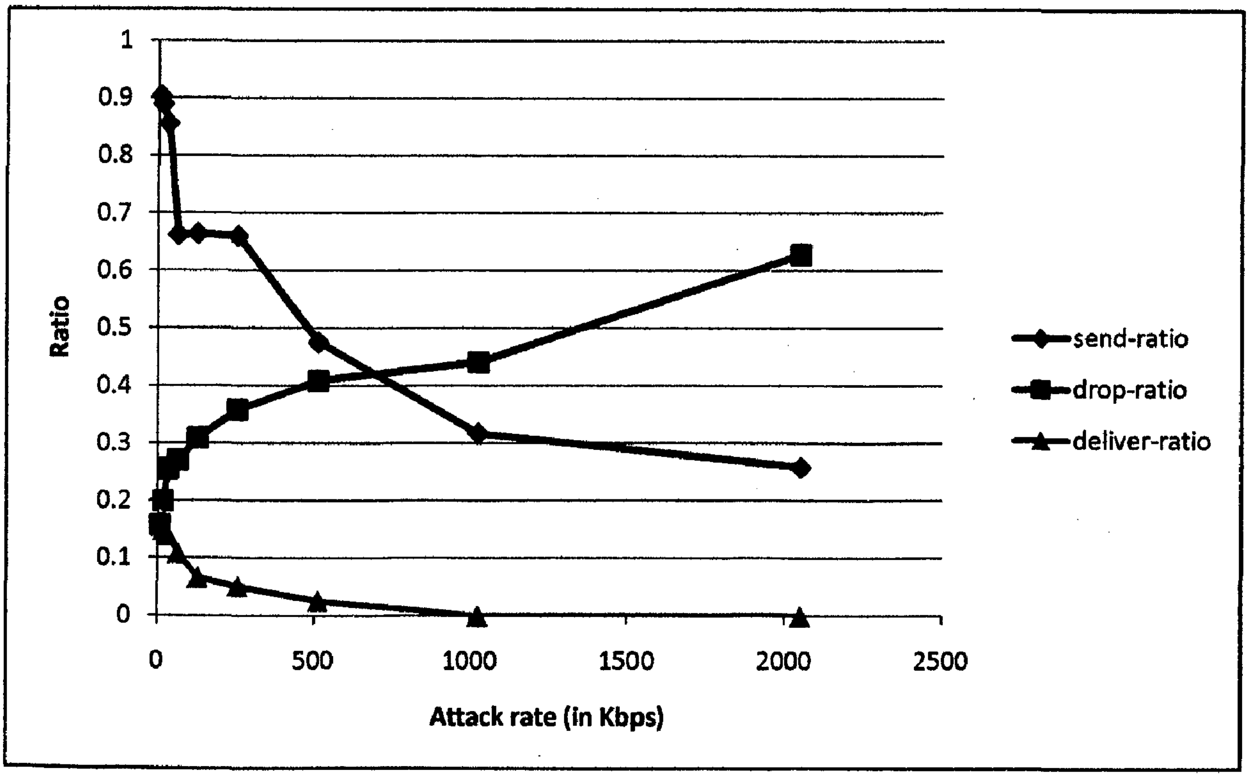

Figure 6.10 Impact of local flooding attack on zone 1 as attack rate is increased (as observed by HoneyMANET)

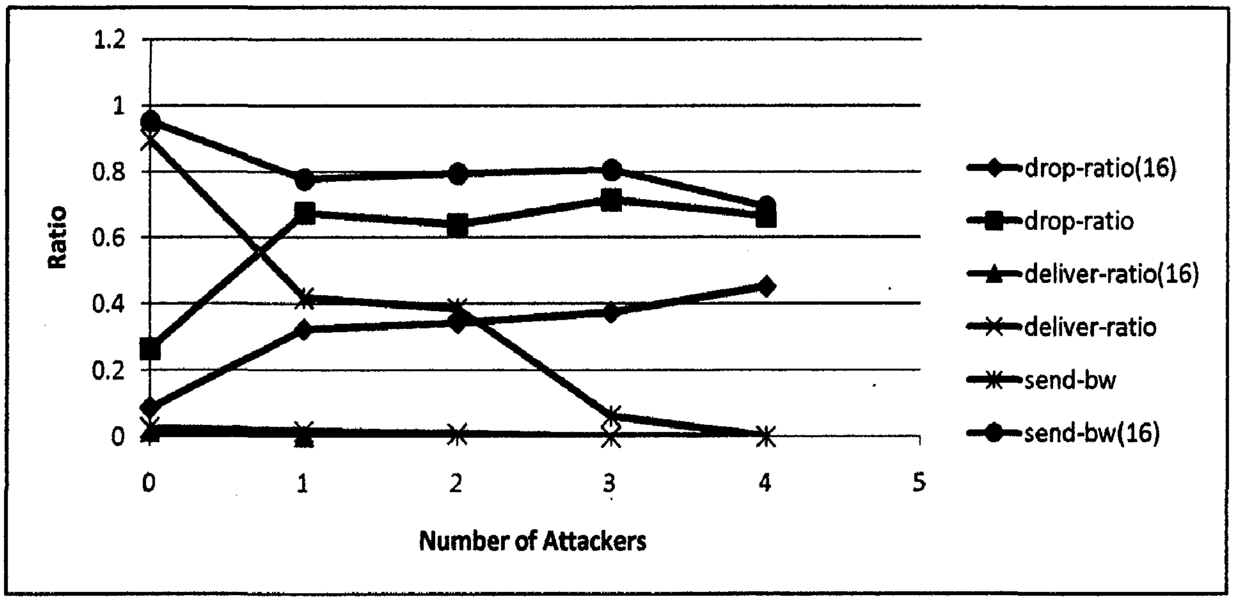

Figure 6.11 Impact of local flooding attack as number of attackers in increased at attack rate 16 kbps and 64 kbps (as observed by HoneyMANET)

Figure 6.12 shows the increasing drop ratio and decreasing send ratio of local region 1 as number of global flooding attackers increases at attack rate 16 kbps.

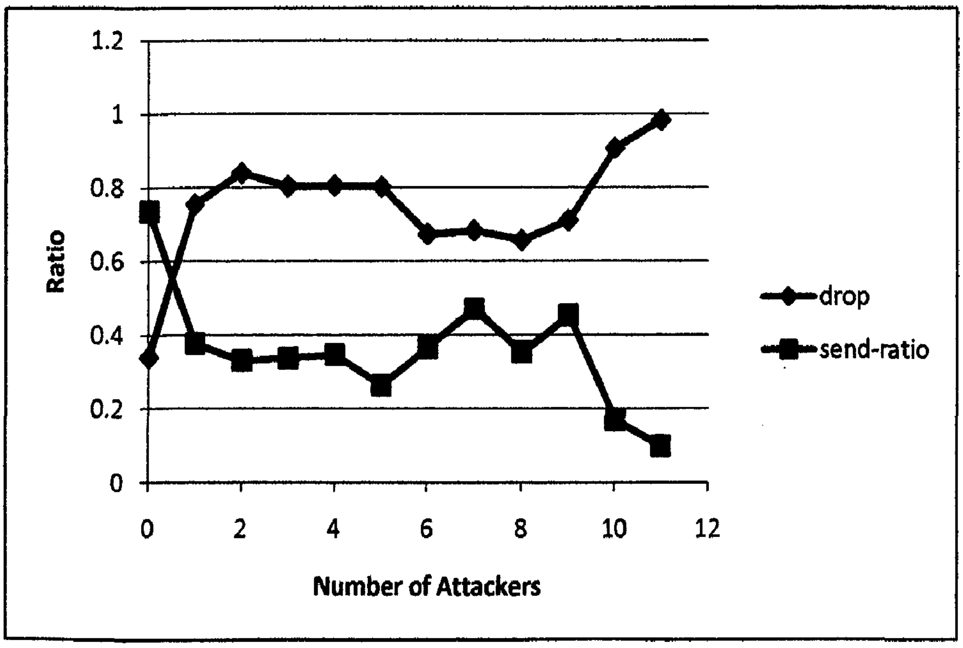

Figure 6.12 Impact of global flooding attack on zone 1 as number of attackers is increased (as observed by HoneyMANET)

Figure 6.13 shows the decreasing global delivery ratio as the number of flooding attackers increases globally. Figure 6.14 shows the increasing global drop ratio as the number of flooding attackers' increases globally at attack rate 16kbps.

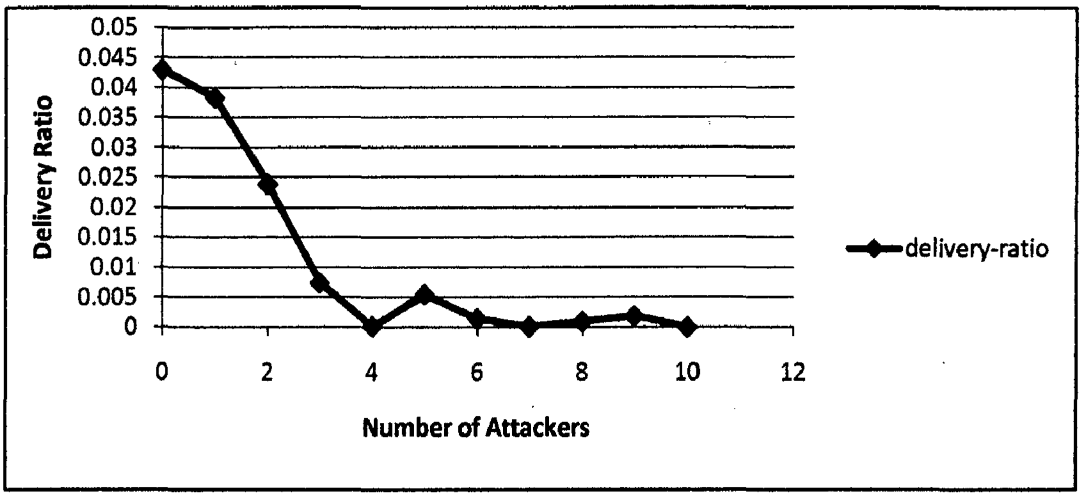

Figure 6.13 Impact of global flooding attack on global delivery ratio as number of attackers is increased (as observed by HoneyMANET)

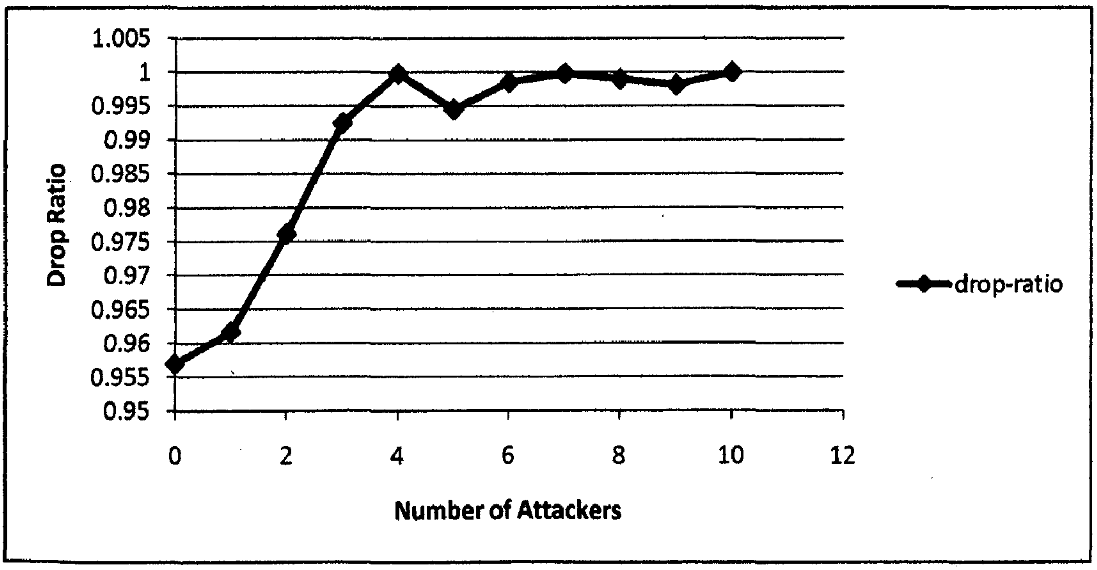

Figure 6.14 Global Impact of global flooding attack on drop ratio as number of attackers is increased (as observed by HoneyMANET)

6.4 Summary

1. RREQ flooding attack is more severe denial of service attack than black hole or simple packet drop attack.

2. In RREQ flooding attack network bandwidth is exhausted and it leads to congestion in the network. Legitimate nodes are denied service by non-availability of bandwidth. Drop ratio increases due to congestion. These two factors lead to decrease in delivery ratio.

3. In flooding attack, attack rate has more drastic effect on network performance than the number of attackers.

4. In black hole attack, traffic is being attracted towards attackers' nodes. The drop ratio increases and delivery ratio got decreased.

5. Simple packet drop attack is less severe than black hole attack. But as the number of attackers' increases, its effect is almost same as that of black hole attack.

6. Local simple packet drop attack affects only the local region. Local black hole attack affects the throughput of the whole network.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusions

In this dissertation, a novel deception based hierarchical intrusion detection system, coined as HoneyMANET because of its deceptive and luring qualities, was designed and simulated using NS2. The whole decision making process regarding the selection of different design parameters for HoneyMANET's setup is given. The attack detection efficiency of HoneyMANET's Anomaly and Misuse detection module is validated and the impact of different attacks on network is evaluated using HoneyMANET.

The following conclusions about HoneyMANET can be drawn from the various simulation results done:

1. Deception as active network defense: The fact that the front-end network where foreign nodes come and join, is a deception and is being monitored, is made hidden from the foreign nodes. Honeynodes deviate the attention of attacker from real production nodes, decreasing the probability of attacks on them. The type of security solution- the intrusion detection model implemented at backend network is unknown to attackers. Therefore, the IDS itself is less prone to be attacked.

2. Controlled network: The randomness and congestion level of the network are controlled. The speed, number and data generation rate of honeynodes is decided by hierarchical management so as to keep these factors- deception, monitoring, relaying and intrusion detection efficient. Simulation results show that the percentage traffic sniffed depends on only the number of honeynodes and not on free movement zone. The minimal number of nodes required for maximum area coverage and high enough traffic sniffing is around 16-20 to give as high as 95-97% traffic sniffing coverage. The speed has more effect on sniffing percentage when the number of honyenodes in each zone is less. As the number increases, the speed doesn't have much effect on sniffing percentage. Considering the

effect of data send rate of honeynodes, it has been shown that increase in data send rate, decreases the sensing capabilities more drastically than the mobility of nodes.

3. Reliable: Honeynodes are always 100% trusted nodes for decision making. They never behave selfishly or maliciously. Therefore, a hundered percent trust factor is always associated with these nodes and so with the network data that is collected and the analysis results that are obtained.

4. Robust: The experimental results show that HoneyMANET is robust in detecting attacks of different kinds (both anomalies and signature based intrusions), targetted at different regions (both localized and globally distributed attacks) of varying rates (at different attack rate and different attackers' number) in dynamically changing ad-hoc environment. The joint intrusion detection module provides a global and localized overview of the all the activities taking place within the network.

5. All types of anomaly detection: Three types of normal profiles are generated- local profile of zone (based on honeynodes behaviour), personal profile (based on its own past behavior) and global profile (based on past behavior of whole network). These different kind of profiles help to detect anomalies of all types and secures the MANET from a wide variety of known and new zero day attacks.

6. Unsupervised intrusion detection: The dynamics of ad-hoc environment - the congestion, bandwidth and randomness effects of network, are captured from the behavior of honeynodes. At any time, the behavior of honeynodes is observed and foreign nodes behavior is tested against it. No separate training phase is needed for anomaly or misuse detection.

7. Efficient: HoneyMANET has high detection rate (mostly remaining at value 1) irrespective of number of attackers because of always presence of trusted nodes. This is a great achievement as compared to IDS of general ad-hoc network where detection rate decreases as number of attacker increases in the network. The false alarms of anomaly detection engine are detected at signature based intrusion detection module, decreasing the false alarm rate of entire model to almost value 0.

8. Evaluation of impact of different attacks: Impact of different attacks on network can also be evaluated using HoneyMANET. As observed by HoneyMANET that RREQ flooding attack is more severe denial of service attack than black hole or simple packet drop attack. In flooding attack, attack rate has more drastic effect on network performance than the number of attackers. Simple packet drop attack is less severe than black hole attack. But as the number of attackers' increases, its effect is almost same as that of black hole attack. Moreover, local simple packet drop attack affects only the local region where as Local black hole attack affects the throughput of the whole network.

9. Less overhead: The two level hierarchy - zone wise data collection and analysis model and centralized global intrusion detection, used in our backend network very much resembles the hierarchical cluster-based intrusion detection architecture. However, in HoneyMANET, zone-heads are pre-decided and their mobility is controlled. This way the overhead of making clusters, choosing the cluster heads, and association of trust is eliminated.

10. More resources: Honeynodes has no other production function except providing security. Therefore, all resources of honeynodes – processing capabilities and energy resources can be dedicated to monitoring, relaying and intrusion detection. They can incorporate more intelligence than the IDS agents installed at mobile production nodes of general ad-hoc networks. They can monitor the network for longer hours and will not even behave selfishly.

7.2 Suggestions for future work

Since this is an open area for research, the following issues may be addressed in future.

1. In the proposed scheme we have used hierarchical architecture at backbone for intrusion detection. Other IDS architectures like distributed and Cooperative, Mobile Agent based IDS running on 100% trusted honeynodes in backbone can be tried out.
2. The proposed scheme deals with only network layer intrusions. Cross Layer attack analysis can be done and host based intrusion detection capabilities can also be added to honeynodes.
3. The attack response module can be added for attack mitigation.
4. The proposed scheme may fall short if the deception is revealed. The security of hidden backend network is itself an issue. Since, the backend network is hidden, therefore primary line of defense i.e. preventive measures can be sufficient to maintain the security of backend network. More resources (CPU processing and Power) can be given to honeynodes. The communication between them at backend network, then can be encrypted to maintain its confidentiality. Certificates can also be pre-distributed to honeynodes for authentication.

REFERENCES

[1]. A. Mishra and K. M. Nadkarni, Security in Wireless Ad Hoc Networks, in Book The Handbook of Ad Hoc Wireless Networks (Chapter 30), CRC Press LLC, 2003.

[2]. P. Papadimitraos and Z. J. Hass, Securing Mobile Ad Hoc Networks, in Book The Handbook of Ad Hoc Wireless Networks (Chapter 31), CRC Press LLC, 2003.

[3]. D. Djenouri, L. Khelladi, and A. N. Badache, "A survey of security issues in mobile ad hoc and sensor networks," Communications Surveys & Tutorials, IEEE, vol. 7, pp. 2-28, 2006.

[4]. H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: challenges and solutions," Wireless Communications, IEEE, vol. 11, pp. 38-47, 2005.

[5]. R. Goel, A. Sardana, R.C. Joshi "Wireless Honeypot: Framework, Architectures and Tools," International Journal of Network Security, (Accepted, Manuscript Id: IJNS-2010-08-23-1)

[6]. I. Stojmenovic, Mobile Ad Hoc Networks and Routing Protocol, in Book The Handbook of Wireless Network and Mobile Computing (Chapter 17), Wiley-Interscience Publication, 2006.

[7]. D. E. Denning, "An Intrusion-detection Model," IEEE Transaction on Software Engineering, 13, No. 7, Pp 222-232, Feb 1987.

[8]. L. Spitzner, "Honeypots: definitions and value of honeypot," 2002, [Online: www.tracking-hackers.com/papers/honeypots.html].

[9]. R. Goel, A. Sardana, R.C. Joshi, "Parallel Misuse and Anomaly Detection Model using C4.5 and CBA algorithms", International Journal of Network Security, (Accepted, Manuscript Id: IJNS-2010-08-16-1).

[10]. C. Krugel and T. Toth, "Applying mobile agent technology to intrusion detection," ICSE Workshop on Software Engineering and Mobility, 2001.

[11]. S. Marti, T. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," In Proceedings of the Sixth Annual International Conference on Mobile Communication and Networking, 2000.

[12]. Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad hoc Networks", In Proc. ACM/IEEE Int'l. Conf. on Mobile Computing and Networking, pp 275-283, 2000.

[13]. Y. Zhang, W. Lee and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," ACM Wireless Networks Journal (ACM WINET), Vol. 9, No. 5, September 2003.

[14]. S. Buchegger and J. Le Boudec, "Performance analysis of the CONFIDANT protocol: (Cooperation of nodes - fairness in dynamic ad-hoc networks)," In Proc. IEEE / ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc'02), Lausanne, Switzerland, pp.226-336, June 2002.

[15]. P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," Communication and Multimedia Security Conference (CMS'02), September 2002.

[16]. P. Albers, O. Camp, et al., "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches," In Proc. 1st International Workshop on Wireless Information Systems (WIS-2002), pp. 1-12, April 2002.

[17]. O. Kachirski and R. Guha, "Effective Intrusion Detection using Multiple Sensors in Wireless Ad hoc Networks," In Proc. 36th Annual Hawaii Int'l. Conf. on System Sciences (HICSS'03), Pp.57.1, 2003.

[18]. Y. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," In Proc. ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN'03), pp. 135-147, October 2003.

[19]. B. Sun, K. Wu and U. Pooch, "Zone-based Intrusion Detection for Mobile Ad hoc Networks," Int. Journal of Ad Hoc and Sensor Wireless Networks, 2003.

[20]. D. Sterne, P. Balasubramanyam, et al., "A General Cooperative Intrusion Detection Architecture for MANETs," In Proc. of the 3rd IEEE International Workshop on Information Assurance (IWIA'05), pp. 57-70, 2005.

[21]. I. Stamouli, P. G. Argyroudis, and H. Tewari, "Real-time Intrusion Detection for Ad hoc Networks," In Proc. of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05), IEEE 2005.

[22]. A. Karygiannis, E. Antonakakis, and A. Apostolopoulos, "Detecting Critical Nodes for MANET Intrusion Detection Systems," 2006.

[23]. S. Bose, P. Yogesh and A. Kannan, "Neural Network Approach for Anomaly Intrusion Detection in Adhoc Networks using Agents," International Journal of Soft computing, 2006.

[24]. X. Wang, "Intrusion Detection Techniques in Wireless Ad Hoc Networks," In Proc. 30th Annual International Computer Software and Applications Conference COMPSAC06, 2006

[25]. R. Ranjana and M. Rajaram, "Detecting Intrusion Attacks in Ad-hoc Networks," Asian Journal in Information Technology, 6(7), 758-761, 2007, ISSN: 1682:3915, Macdwell Journal 2007.

[26]. N. Marching and R. Datta, "Collaborative Technique for Intrusion Detection in Mobile Ad hoc Network," Ad hoc Networks, 6, Issue 4, June 2008 Page 508-523.

[27]. S. Madhavi and T. H. Kim, "An Intrusion Detection System in Mobile Ad hoc Networks," International Journal of Security and its Application, 2, No 3, July 2008.

[28]. S. Sen and J. A. Clark, "A Grammatical Evolution Approach to Intrusion Detection on Mobile Ad hoc Networks," In Proc. Second ACM Conference on Wireless Network Security WiSec'09, March 2009.

[29]. Network Simulator- NS2, [Online: http://www.isi.edu/nsnam/ns/].

[30]. Awk, [Online: http://www.grymoire.com/Unix/Awk.html].

[31]. M. Just, E. Kranakis, and T. Wan, "Resisting malicious packet dropping in wireless ad hoc networks," In Proc. ADHOC-NOW, Montreal, Canada, Oct. 2003

[32]. Nakkeeran, B. Partibane, S. S. Murugan, and N. Prabagarane, "Detecting the Malicious Faults in Manet," In Proc. National Conference on Communications (NCC), India, 2005.

List of Publications

[1]. Radhika Goel, Anjali Sardana, R.C. Joshi, "C4.5 based Sequential Attack Detection and Identification Model," In Proc. of International Conference on Advances in Communication, Network, and Computing (CNC 2010).

[2]. Radhika Goel, Anjali Sardana, R.C. Joshi, "Parallel Misuse and Anomaly Detection Model using C4.5 and CBA algorithms," International Journal of Network Security, (Accepted, Manuscript Id: IJNS-2010-08-16-1).

[3]. Radhika Goel, Anjali Sardana, R.C. Joshi, "Wireless Honeypot: Framework, Architectures and Tools," International Journal of Network Security, (Accepted, Manuscript Id: IJNS-2010-08-23-1).

PAPER COMMUNICATED

[1]. Radhika Goel, Anjali Sardana, R.C. Joshi, "Honeypot based Mobile Ad-Hoc Networks – HoneyMANET," Information Security Journal: A Global Perspective, (Under Review, Manuscript Id: UISS-2010-0080).