

DTQ BASED DETECTION AND IDENTIFICATION OF PACKET FORWARDING MISBEHAVIOR IN MANETS

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

INTEGRATED DUAL DEGREE

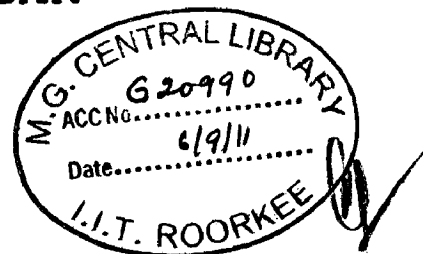
in

COMPUTER SCIENCE AND ENGINEERING

(With Specialization in Information Technology)

By

MAYANK MAHAJAN



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 “**DTQ based Detection and Identification of Packet Forwarding Misbehavior in MANETs**” 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 **Dr. Anjali Sardana, Assistant Professor, 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: June, 2011

Place: IIT Roorkee



(Mayank Mahajan)

CERTIFICATE

This to certify that the work contained in the dissertation entitled “**DTQ based Detection and Identification of Packet Forwarding Misbehavior in MANETs**” by Mayank Mahajan 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: June, 2011

Place: IIT Roorkee


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

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. Anjali Sardana**, Assistant Professor, 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 her inspirational support and I consider myself fortunate to have done my dissertation under her guidance.

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 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.

Mayank Mahajan

Table of Contents

CANDIDATE'S DECLARATION	i
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	v
LIST OF TABLES	vi
ABSTRACT	viii
CHAPTER 1: INTRODUCTION	1
1.1 MOTIVATION.....	2
1.2 PROBLEM DESCRIPTION.....	3
1.3 STRUCTURE OF THE REPORT.....	3
CHAPTER 2:BACKGROUND AND LITERATURE SURVEY	5
2.1 ATTACKS IN MANETS.....	5
2.2 DESCRIPTION OF ATTACKS BASED ON PACKET FORWARDING MISBEHAVIOR.....	6
2.2.1 Black hole attack.....	6
2.2.2 Gray hole attack.....	6
2.2.3 Neighbor attack.....	6
2.2.4 Message Tampering.....	7
2.3 INTRUSION DETECTION IN MANETS.....	7
2.3.1 Behavior based techniques.....	7
2.3.2 IDS Architectures in MANETs.....	8
2.4 MULTICAST ROUTING IN MANETS.....	9
2.4.1 Mesh Establishment in ODMRP	10
2.4.2 Route redundancy in ODMRP.....	11
2.4 EXISTING SECURITY TECHNIQUES FOR MANETS.....	12
2.5 RESEARCH GAPS.....	18
CHAPTER 3: DTQ BASED TECHNIQUE FOR DETECTION OF MISBEHAVING NODES	19
3.1 DEFINITIONS.....	19
3.2 ASSUMPTIONS.....	19

3.3 DTQ BASED DETECTION OF PACKET FORWARDING MISBEHAVIOR.....	20
3.3.1 Detection of Malicious node.....	20
3.3.2 Elimination of Malicious node.....	21
3.3.3 Algorithmic details.....	22
3.4 PROPOSED IDS ARCHITECTURE AND DESIGN.....	23
CHAPTER 4: SIMULATION AND IMPLEMENTATION DETAILS.....	26
4.1 QUALNET.....	26
4.2 SIMULATION PARAMETERS.....	26
4.3 SIMULATION OF ATTACKS.....	27
4.3.1 Black hole.....	27
4.3.2 Gray hole.....	30
4.4 SIMULATION OF THE PROPOSED TECHNIQUE.....	31
CHAPTER 5: SIMULATION RESULTS AND ANALYSIS.....	33
5.1 PERFORMANCE METRICS.....	33
5.2 CHOICE OF DTQ THRESHOLD.....	33
5.3 RESULTS AND DISCUSSION FOR PACKET DROP ATTACKS.....	33
5.3.1 Varying small bucket size and large bucket size.....	33
5.3.2 Varying DTQ threshold.....	36
5.3.3 Varying transmission error probability.....	38
5.3.4 Varying number of attackers.....	39
5.4 DISCUSSION FOR DATA ALTERATION ATTACKS.....	41
5.5 COMPARISON WITH EXISTING TECHNIQUES.....	42
5.6 ANALYSIS OF THE APPROACH.....	43
CHAPTER 6: CONCLUSION & FUTURE WORK.....	45
REFERENCES.....	46

LIST OF FIGURES

Figure 1.1	Interference of malicious node.....	2
Figure 2.1	Classification of Security Attacks for different layers	5
Figure 2.2	Multicast versus Unicast.....	9
Figure 2.3	On-Demand Multicast Routing Protocol.....	11
Figure 2.4	Classification of misbehaving node detection techniques	12
Figure 3.1	Promiscuous listening.....	19
Figure 3.2	Identifying malicious nodes.....	21
Figure 3.3	Distribution of IDS agents.....	23
Figure 3.4	IDS design.....	24
Figure 3.5	IDS and TCP/IP protocol stack.....	25
Figure 4.4	Modified files with their functions.....	31
Figure 5.1	varying small bucket size.....	34
Figure 5.2	effect of large bucket on detection time.....	35
Figure 5.3	varying DTQ threshold.....	36
Figure 5.4	Malicious node detected Vs. Forwarding rate.....	37
Figure 5.5	Varying transmission error probability.....	38
Figure 5.6	Throughput vs. No. of Attackers.....	39
Figure 5.7	Comparison in terms of percentage improvement in throughput.....	39
Figure 5.8	end-to-end delays vs. No. of Attackers.....	40
Figure 5.9	Malicious node detected vs. No. of Attackers.....	41
Figure 5.11	DTQ Vs D/E.....	43

LIST OF TABLES

Table 2.1	Comparison of Existing solutions.....	17
Table 4.1	Simulation parameters.....	26
Table 4.2	Attack Design parameters	27

ABSTRACT

In recent years the widespread availability of wireless communications, mobile computing and handheld devices has led to the growth and significance of wireless mobile ad hoc networks. The ability to establish communication without an infrastructure at very low cost and the capacity to communicate beyond the node's wireless transmission range embarks Mobile Ad hoc Networks (MANET) as the deployment ground for various fields such as wireless sensor networks, ubiquitous networks and peer-to-peer networks.

Security issues are paramount in such networks. Nodes in MANETs may launch various attacks or may become selfish to save their resources. These nodes can be termed as malicious. Detection of such malicious nodes is critical to success of MANETs. Detecting malicious nodes in an open ad hoc network is more complicated than in traditional wired networks. Also, MANETs have several operational constraints like bandwidth, battery power, CPU, memory. Considering all these factors, developing a technique for detection of malicious nodes in MANETs is very challenging. A variety of techniques have been proposed but all of them have their limitations.

In this work, a new technique for detection of malicious node is suggested. This technique is based on a metric known as Data Transmission Quality (DTQ) which decides the communication quality of a node by considering both its near-term and long-term behaviour. A node is blacklisted if its DTQ value falls below the threshold. The DTQ function is defined in a way that it will keep close to constant or change smoothly for legitimate nodes and will keep decreasing for suspicious nodes. We have also proposed a design of an intrusion detection system based on the DTQ based technique.

The proposed solution is tested by simulating it in Qualnet. The results show that the solution works well and is able to find all the malicious nodes. False positives are found to be very low and can be reduced to zero by adjusting parameters like bucket sizes, DTQ threshold etc depending on the network behaviour.

Chapter 1 Introduction

A MANET is referred to as a network that is autonomous, self-configuring, and network without infrastructure where mobile nodes communicate via wireless links. Nodes within each other's wireless transmission ranges can communicate directly; however, nodes outside each other's range use the concept of multi-hop communication where several intermediate hosts relay the packets sent by the source host before they reach the destination host [1]. In MANETs, every node functions both as a host and as a router. The nodes in MANETs move freely, in any direction or speed, and are allowed to organize themselves arbitrarily.

In MANETs, the network topology changes dynamically and unpredictably. A Node can forward data to any other node, often in a peer-to-peer, multi-hop mode. Therefore, MANETS possess a need to dynamically determine routing based on availability or visibility of nodes. MANETs also have nodes whose energy storage is very limited. Often, they are battery equipped, with very limited to no recharging or replacement possible. Another limited resource in MANETs is bandwidth.

In MANETs, security is a major concern. Due to lack of a fixed infrastructure, dynamic topology and limited resources, securing MANET becomes very challenging. There is a wide variety of attacks in MANETs (A detailed discussion over possible attacks in MANETs is presented in section 2.1). An adversary can launch a misbehaving node in the network or a legitimate node may become selfish in order to save its resources. Such misbehaving nodes, also termed as malicious nodes, have to be detected and are to be avoided in forwarding of data. Therefore, guaranteeing data safety and reliability is a major concern.

MANETs have a lot of applications: In a battlefield, a network need to be formed in real time for transmitting information; business associates may need to share information during a meeting; during an interactive conference; and during emergency, disaster relief personnel coordinating efforts after a fire, hurricane, or earthquake. The other possible applications include personal area and home networking, location-based services, and sensor networks.

1.1 Motivation

The connectivity of mobile nodes in MANETs strongly relies on the fact that ensures cooperation among the nodes in the network. Recently variety of network layer attacks have been identified and heavily studied in research papers. As a consequence of attacking network layer, adversaries can easily disturb and absorb network traffic, inject themselves into the selected data transmission path between the source and destination, and thus control the network traffic flow, as shown in Figure 1.1, where a malicious node M can interfere between any of the intermediate nodes participating in the communication in the chosen path (in the figure 1 to N represents the number of intermediate nodes) between source S and destination D [2].

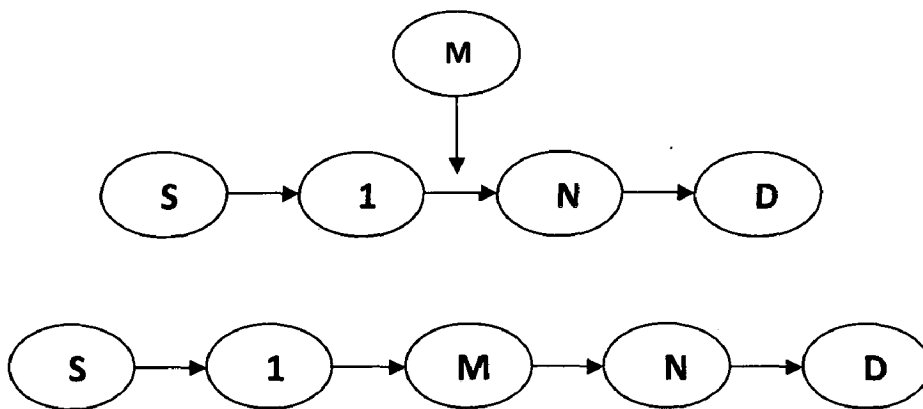


Figure 1.1 Interference of malicious node

The packets in the network traffic could be dropped completely or forwarded selectively which introduces significant packet losses in the network. Packets may be sent to a non-existing path or to a sub-optimal path. This kind of packet forwarding misbehavior results in degradation of network performance. Since MANETs have a variety of applications, as discussed in previous section, detection of such misbehaving nodes is critical for the success of MANETs. A lot of efforts have been made in this direction. But, all of them have one or more limitations. Therefore, there is still a need of a solution which overcomes all limitations and is able to detect such misbehaving nodes effectively.

While securing MANET there are certain challenges to be faced because of some of its inherent characteristics. Like, nodes in MANETs are highly mobile and topology changes in sometimes unpredictable manner. MANETs lack fixed traffic points, i.e. there are no firewalls or

routers as in classical computer networks, and each node acts as a router. Also, Host-resident network intrusion detection systems have their limitations in case of MANETs. Sometimes, Detectors may also become the target of an attack. Wireless communication (RF medium) is susceptible to eavesdropping, jamming, interference and many other MAC threats that may result in loss of packets and connectivity. The resources in MANET environment are limited, e.g. energy (battery operated nodes), varying throughput because of dynamic topology configuration. All these factors have to be considered while designing a technique for malicious node detection.

1.2 Problem Description

1.2.1 Problem Statement

To design an efficient and accurate mechanism for detection and identification of packet forwarding misbehavior in MANET environment.

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

1. To propose an algorithm to detect and identify misbehaving nodes in MANETs.
2. To enhance the algorithm for improved performance.
3. To simulate and validate the proposed mechanism.

1.3 Organization of the report

This report contains six chapters including the current chapter. The rest of the report is structured as follows:

Chapter 2 provides the background and literature review. It discusses Intrusion detection in MANETs, attacks in MANETs and the exiting solution to secure MANETs. It concludes with the research gaps in the existing solutions.

Chapter 3 presents the DTQ based malicious node detection in MANETs and discusses it in detail.

Chapter 4 provides a brief overview of the simulator used in this work i.e. Qualnet. Then it discusses the simulation and implementation details of the proposed approach.

Chapter 5 shows the simulation results and analysis.

Chapter 6 concludes the report and provides suggestion for future work.

Chapter 2 Background and Literature Review

This chapter is divided into six sections. Section 2.1 discusses attacks in MANETs; Section 2.2 provides description for major network layer attacks; Section 2.3 details out Intrusion Detection in MANETs and their architectures; Section 2.4 elaborates the multicast routing protocol used in this work i.e. ODMRP; Section 2.5 discusses the existing solutions and points out their limitations; Section 2.6 presents the research gaps.

2.1 Attacks in MANETs

MANETs are prone to many attacks. Broadly, attacks in MANETs can be classified into two categories [3]: Active and Passive. During a passive attack the exchanged data is obtained without disrupting the communication while an active attack involve information interruption, modification or fabrication [4]. Figure 2.1 shows the classification of attacks in MANETs.

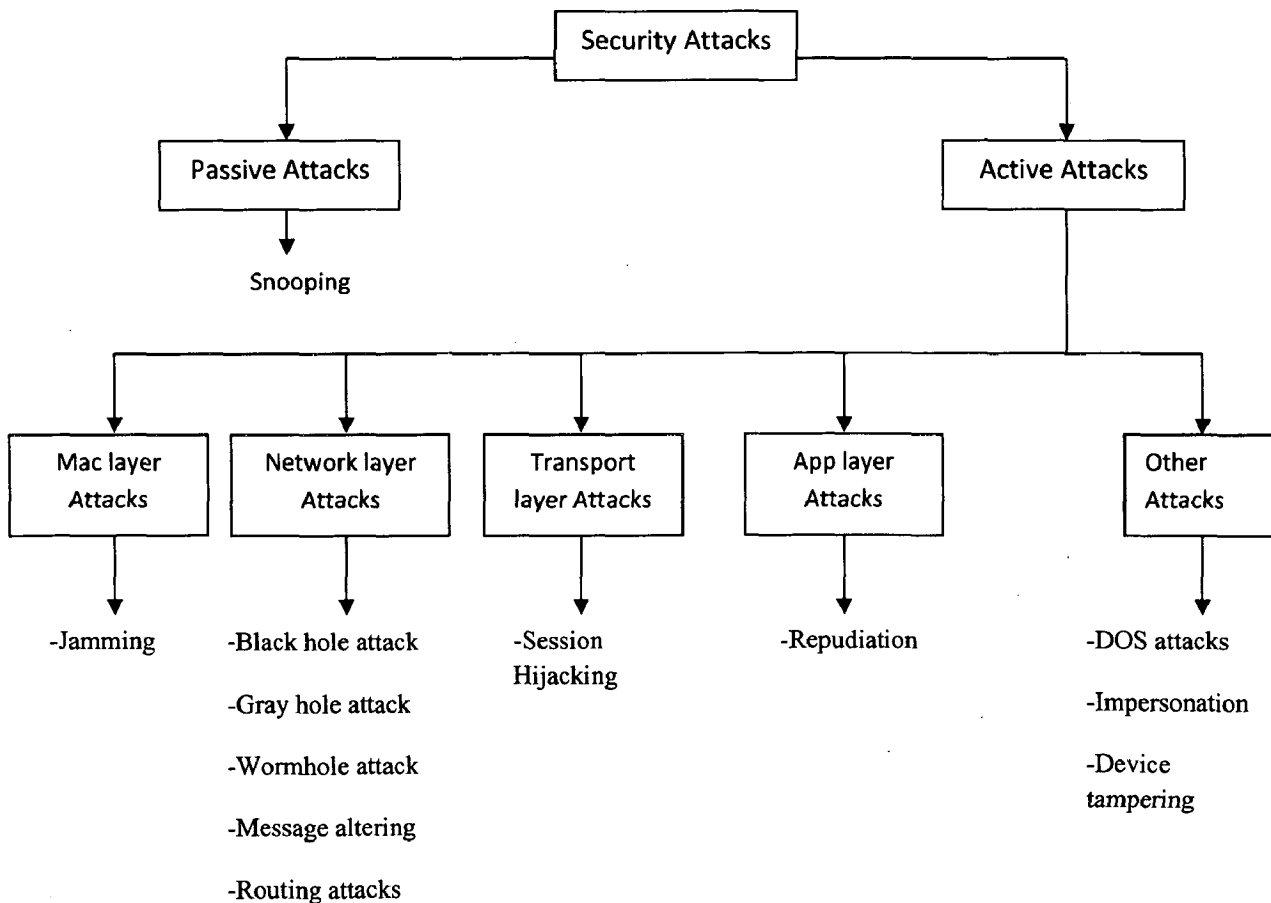


Figure 2.1 Classification of Security Attacks for different layers.

2.2 Description of Network Layer Attacks

Since our focus is on the attacks based on packet forwarding misbehavior, which are a kind of active attacks, we will only be discussing some of the major attacks in that category. Packet forwarding misbehavior is further classified as: (1) Packet drop attacks e.g. Blackhole and Grayhole; (2) Data alteration attacks e.g. Message tampering and Neighbor attack.

2.2.1 Black hole Attack

The variety of attacks in the network layer differs such as not forwarding the packets or adding and modifying some parameters of routing messages; such as sequence number and hop count. The most basic attack executed by the nodes in the network layer is that an adversary can stop forwarding the data packets. The consequence caused by this is that, whenever the adversary is selected as an intermediate node in the selected route, it denies the communication to take place. For example: In AODV routing protocol, consider a malicious node which keeps waiting for its neighbors to initiate a RREQ packet. As the node receives the RREQ packet, it will immediately send a false RREP packet with a modified higher sequence number. So, that the source node assumes that node is having the fresh route towards the destination. The source node ignores the RREP packet received from other nodes and begins to send the data packets over malicious node. A malicious node takes all the routes towards itself. It does not allow forwarding any packet anywhere. This attack is called a black hole as it swallows all the data packets [5].

2.2.2 Gray hole Attack

A variation of black hole attack is the gray hole attack, in which nodes either drop packets selectively (e.g. dropping all UDP packets while forwarding TCP packets) or drop packets in a statistical manner (e.g. dropping 50% of the packets or dropping them with a probabilistic distribution). Both types of gray hole attacks seek to disrupt the network without being detected by the security measures in place [6].

2.2.3 Neighbor attack

Upon receiving a packet, an intermediate node records its ID in the packet before forwarding the packet to the next node. However, if an attacker simply forwards the packet without redirecting its ID in the packet, it make two nodes that are not within the communication range of each other

believe that they are neighbors (i.e. one hop away from each other), resulting in a disrupted route. The neighbor attack and black hole attack prevent the data from being delivered to the destination. But the neighbor attacker does not catch and capture the data packets from the source node. It leaves the settings as soon as sending the false messages.

2.2.3 Message Tampering

This type of attack is launched by the adversaries acting as compromised nodes during communication. They tend to take all the data packets and modify the data which may be regarding the network topology, optimal routes etc; either by adding additional bytes or by deleting existing bytes. A small change in the data may obviously cause abnormalities or havoc in the network.

2.3 Intrusion Detection in MANETs

An IDS is a device (or application) that monitors network and/or system activities for malicious activities or policy violations and produces reports to a Management Station. Traditional IDS were developed keeping in mind the wired infrastructure of the network. These solutions place the IDS on certain strategic points like Switches, Gateways, etc [7]. MANETs do not have any fixed infrastructure; old systems need a lot of modification and changes to work in wireless environment. There have been many approaches to intrusion detection in MANETs. The approaches can be broadly classified into two categories: Authentication based and Behavior based. Authentication based approach rely on the identification of nodes by a unique identifier. Use of encryption keys fall into this category, and they have been deeply studied. The second approach is behavioral based techniques where intrusion is defined based on nodal activities, rather than its identifier.

2.3.1 Behavior based Intrusion detection techniques

The behavioral based techniques of intrusion detection can be divided into two main categories: Anomaly detection and misuse detection. Anomaly detection techniques assume that all intrusive activities are necessarily anomalous. This means that if we could establish a "normal activity profile" for a system, then in theory, all system states varying from the established profile by statistically significant amounts are intrusion attempts. The concept behind misuse

detection scheme is that there are ways to represent attacks in the form of a pattern or a signature so that even variations of the same attack can be detected.

2.3.2 IDS Architectures in MANETs

MANETs could have one of the two possible network infrastructures: flat or multi-layer. In flat infrastructure, all nodes are considered equal. In multi-layer infrastructure, clusters are formed in which nodes within a cluster can communicate directly while inter-cluster communication is done through cluster-heads. Therefore, IDS architecture may depend on network infrastructure [8]. The various possible IDS architectures are:

Stand-alone Intrusion Detection Systems: In this architecture, each node runs an intrusion detection system to determine intrusions. There is no cooperation among nodes in the network. Every decision made is based only on information collected at its own node. Therefore, no data is exchanged. This architecture has not been chosen in most of the IDS for MANETs because of its low efficiency and high power consumption. It is more suitable for flat networks.

Distributed and Cooperative Intrusion Detection Systems: In this architecture, every node participates in intrusion detection and response by having an IDS agent running on them. An IDS agent is responsible for detecting and collecting local events and data to identify possible intrusions, as well as initiating a response independently. However, neighboring IDS agents cooperatively participate in global intrusion detection actions when the evidence is inconclusive. Similarly to stand-alone IDS architecture, this architecture is more suitable for flat network infrastructure.

Hierarchical Intrusion Detection Systems: Hierarchical IDS architectures have been proposed for multi-layered network infrastructures where the network is divided into clusters. Each node runs an IDS agent and is responsible for detecting local intrusions. A cluster-head is responsible locally for its node as well as globally for its cluster, e.g. monitoring network packets and initiating a global response when network intrusion is detected.

Mobile Agent for Intrusion Detection Systems: A concept of mobile agents has been used in several techniques for intrusion detection systems in MANETs. Due to its ability to move through the large network, each mobile agent is assigned to perform only one specific task, and

then one or more mobile agents are distributed into each node in the network. This allows the distribution of the intrusion detection tasks [9].

2.4 Multicast Routing in Mobile Ad hoc Networks

In MANETs, communication and collaboration among a given group of nodes are usually necessary. Multicast routing is a preferred communication mechanism over multiple unicast transmissions in order to deliver the same data to multiple recipients. Multicast is a form of communication that delivers information from a source to a set of destinations simultaneously in an efficient manner; the messages are delivered over each link of the network only once and only duplicated at branch points, where the links to the destinations split (Figure 2.2). Multicast communications are especially useful in applications such as distribution of newsletters and software; audio/video conferencing; online education; online Internet games; and communications among military troops or rescue teams.

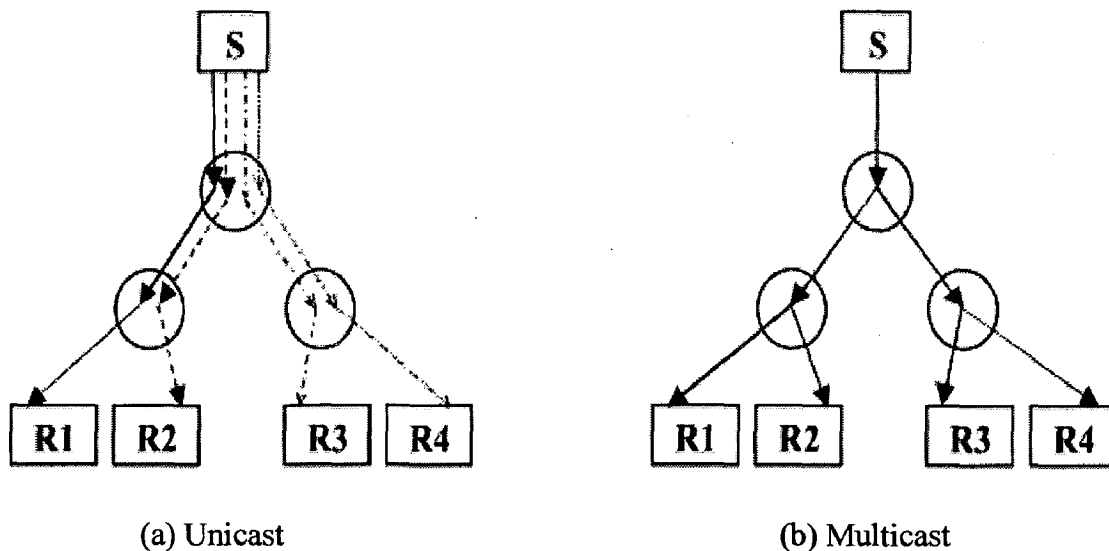


Figure 2.2: Multicast versus Unicast

Existing multicast routing protocols in MANETs can be classified broadly into two categories: tree-based and mesh-based. In tree-based multicast protocols, there is usually only one single path between a sender and a receiver, while in mesh-based multicast protocols, there

may be multiple paths between each sender-receiver pair. Example of tree-based multicast protocols is MAODV [10]. Typical mesh-based multicast protocols are ODMRP [11] and DCMP. MCEDAR is a hybrid multicast protocol that provides both mesh-based and tree-based infrastructure.

Compared to tree-based protocols, mesh-based protocols are more robust and suitable for systems with frequently changing topology such as MANETs. Maintaining a single multicast tree is not appropriate for MANETs because the tree could easily break due to highly dynamic topology and node mobility. The instability of multicast trees results in higher packet losses, and an increase in the number of retransmissions. In contrast, routing meshes allow multicast data to be delivered to a destination on alternative paths even when the main route breaks, due to the availability of multiple paths between a source and a destination. In our study, we used the On-Demand Multicast Routing Protocol (ODMRP), a mesh-based routing protocol, due to its simple implementation and high packet delivery ratio. The following sections describe the operation of ODMRP in detail.

2.4.1 Mesh Establishment in ODMRP

ODMRP uses the concept of forwarding group, which is a set of nodes responsible for forwarding multicast data on shortest delay paths between a sender and a receiver. An ODMRP source periodically updates routing tables and membership information by flooding the network with route refreshment packets, Join Query. The period of time between each Join Query transmission is called refreshment interval. Upon receiving a Join Query, an intermediate node stores the ID of the upstream node from which it receives the packet in to the routing table, and then rebroadcasts the packet (duplicate Join Query packets will be discarded). When the Join Query packet reaches a multicast receiver, the receiver replies with a Join Reply packet, which contains the multicast source ID, and the corresponding next node ID from which it received the Join Query packet. The Join Reply packet is then relayed back towards the multicast source via the reverse path traversed by the Join Query packet.

When a node receives a Join Reply, it checks if the next node ID in the Join Reply packet matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets a forwarding group flag and sends its own Join Reply further to a next node based on the routing table. The Join Reply is thus propagated by each

forwarding group member until it reaches the multicast source. The route between a source and receiver is established after the source receives the Join Reply packet. This process constructs (or updates) routes from the sources to the receivers, and builds a mesh of forwarding nodes, the “forwarding group”.

An example of the mesh establishment phase is illustrated in Figure 2.2 (a). For initializing the multicast mesh, sources S1 and S2 flood the network with Join Query packets. When receivers R1, R2, R3 and R4 receive the Join Query packet, each node sends a Join Reply packet along the reverse path to the sources. For example in Figure 2.3 (a), receiver R1 receives Join Query packets from sources S1 and S2 through paths S1-I1-I3-R1 and S2-I3-R1, respectively. When node I3 receives the Join Reply packet from receiver R1, it sets the forwarding group flag and becomes the forwarding node for that particular multicast group. After sources S1 and S2 have received the Join Reply packets, a multicast mesh for sources S1 and S2 is established as shown in Figure 2.3 (b).

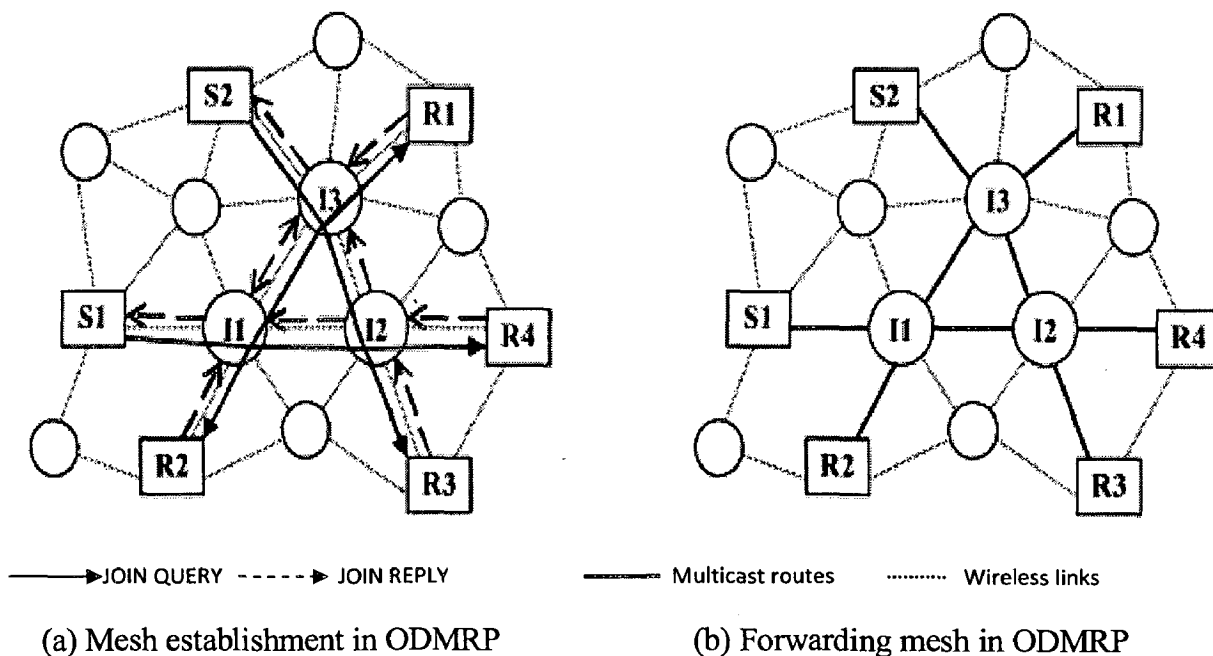


Figure 2.3 On-Demand Multicast Routing Protocol

2.4.2 Route Redundancy in ODMRP

Route redundancy in the multicast mesh helps ODMRP overcome frequent link breaks due to node mobility and channel fading in wireless communications. In Figure 2.3 (b), suppose

the route from source S1 to receiver R4 is S1-I1-I2-R4. If the link between nodes I1 and I2 breaks or fails, R4 can still receive data packets from S1 through an alternative route S1-I1-I3-I2-R4. This redundancy helps to achieve high connectivity among multicast members, and hence high percentage of packet delivery ratio.

2.5 Existing security techniques in MANETs

There are various techniques which have been used for detection of misbehaving nodes. The method of detection can generally be classified into authentication-based method and behavior-based. The idea of authentication-based method is to confirm the authenticity of a node by verifying its key information, or some functions that depends on its key. Thus how the keys are established plays a critical role. The idea behind behavior based techniques is that legitimate nodes will behave differently in comparison to malicious nodes. Behavior based techniques are further classified into two categories: Misuse based and Anomaly based. Next section will discuss the existing security techniques to detect packet forwarding misbehavior. Figure 2.4 shows the classification based on detection technique. We first discuss some major authentication based techniques and their limitations. Then we will discuss some major behavior based techniques.

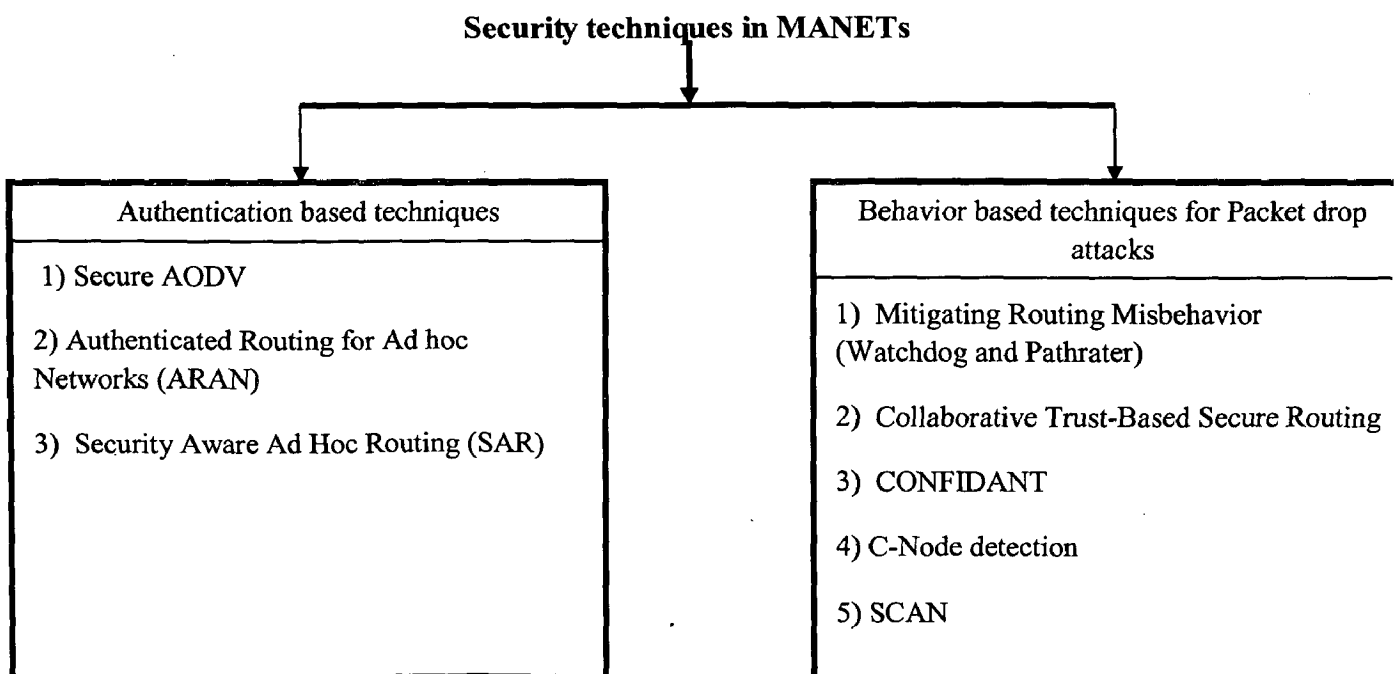


Figure 2.4 Classification of misbehaving node detection techniques

Secure AODV (SAODV) [12] [13] employs asymmetric mechanisms to achieve authentication, integrity, confidentiality and non-repudiation. It is assumed that the public key should come with the IP address of the node and the network leader's IP address as the mask address, to avoid impersonation attacks. The source with the help of signature key pair signs the mutable fields of the RREQ and in the case of RREP it is signed by destination. Hence both can verify and authenticate each other using their public keys. The signature contains the seed of the hash chain embedded within it, which secures the hop count. For each hop the intermediate nodes increase the hop by hashing the previous hash count value. The one-way nature of the hash chain prevents the reduction of the hop count.

Drawbacks: Certificates bounded with IP addresses are unrealistic, as nodes may be assigned with dynamic IP addresses. Deployment of asymmetric key techniques not only raise issues like incremental deployment and key revocation but also consumes huge resource in an energy-constrained environment due to the high processing overhead at each node for every request. SAODV is still prone to the same distance fraud [14], where the forwarding node fails to increment the route metric, as there is no enforcement to do so. Moreover, SAODV never considers the misbehaving detection methods and also does not take any attempt to prevent DOS attacks because it assumes that DOS attacks are more predominant and restricted to physical layer; this is not true, for example colluding malicious nodes can drop packets during route discovery phase.

In **SAR** (Secure-Aware Ad Hoc Routing protocol), nodes are grouped based on the trust level [15] and the source node initiating the route request suggests that only nodes satisfying the minimum security level can take part in the route discovery and other nodes that do not have the necessary trust level have to drop the request packets. To secure and differentiate each level, a level is assumed to share a key, which can also participate with lower levels but not with higher levels.

Drawbacks: A malicious node at a particular level can launch any attack at its level or at lower levels. Moreover, it fails to address the global secure routing problem and concentrates on secure routing in a context, where nodes of a certain group are assumed to be trustworthy. The fixed assignment of trust levels further worsens the design.

In ARAN (Authenticated Routing for Ad hoc Networks) [16], the certificates signed by the certificate authority, associate each node's IP address with its public key. In a route request, the source includes its certificate, target's IP address, nonce, and timestamp for freshness and authenticity. An intermediate node removes the previous forwarding node's signature and certificate (except the source node's signature and certificate), signs the route request and includes its own certificate. Similarly, when any node receives the route reply, it removes the signature and certificate of the previous hop from whom the route reply was received (except the signature and certificate of target node, which is actually the destination node for the route request), signs the original reply from the target and includes its own certificate. The intermediate node establishes an entry in the routing table for the source or the target, when it receives the request or reply respectively. Route request and route reply are similar except that the request is a broadcast and the reply is a unicast.

Drawbacks: Due to the heavy computation involved with the certificates, the ARAN system is vulnerable to many DOS attacks. Even when there is no malicious node, the load levied on the legitimate intermediate nodes force them to drop the packets in order to conserve their resources.

SCAN (self-organized network layer security in mobile ad hoc networks) focuses on securing packet delivery. It uses AODV, but argues that the same ideas are applicable to other routing protocols. SCAN assumes a network with sufficient node density that nodes can overhear packets being received by a neighbor, in addition to packets being sent by the neighbor. SCAN nodes monitor their neighbors by listening to packets that are forwarded to them. The SCAN node maintains a copy of the neighbor's routing table and determines the next-hop node to which the neighbor should forward the packet; if the packet is not overheard as being forwarded, it is considered to have been dropped [17].

In [18], Sergio Martí et al proposes a system that can mitigate the effects of packet dropping has been proposed. This is composed of two mechanisms that are kept in all network nodes: a watchdog and a pathrater. The watchdog mechanism identifies any misbehaving nodes by promiscuously listening to the next node in the packet's path. If such a node drops more than a predefined threshold of packets the source of the communication is notified. The path rater mechanism keeps a rate for every other node in the network it knows about. A node's rate is

decreased each time a notification of its misbehavior is received. Then, nodes' rates are used to determine the most reliable path towards a destination, thus reducing the chance of finding a misbehaving node along the selected path.

Drawbacks: Exchanging ratings opens door for blackmail attack, where a malicious node can report a legitimate node to be a misbehaving node due to lack of authenticity. Moreover, the watchdog might not detect a misbehaving node in the presence of ambiguous collisions, receiver collisions or nodes capable of controlling their transmission power. Such weaknesses are the result of using promiscuous listening to determine whether a node has forwarded a packet or not.

In [19] Pissinou et al proposes a collaborative method for secure routing. Each ROUTE REQUEST contains a trust-level field, which is modified by the receiving intermediate nodes to include the trust level of the node that sends the ROUTE REQUEST. The intermediate node after retransmitting the ROUTE REQUEST monitors the one-hop neighbors for verification. If any change is found in the one-hop neighbor's re-broadcasted ROUTE REQUEST, the monitoring node generates a warning. On successful completion of ROUTE REQUEST phase, the destination chooses the route based on the trust-metric and replies with the ROUTE REPLY, which contains the next to avoid the black-hole attack. The approach declines to address how the trust is represented, captured and evaluated.

In [20] a method is proposed known as **CONFIDANT**. It has four components:

The Monitor: Monitors the environment and invokes reputation system when it detects a deviating behavior.

The Reputation System: The rating in the Reputation System gets altered once the action exceeds the threshold limit. Further, if the rating of misbehaving node surpasses intolerable level, then the Path Manager is called to take action.

The Path Manager: The Path Manager apart from deleting the misbehaving node in its routes generates an ALARM message to the Trust Manager

The Trust Manager: Can also receive ALARM message externally from the friends or other nodes through the Monitor component for trust examination and evaluation. The generated ALARM messages are sent to friends or to the route initiator.

Drawbacks: There is no proper method to integrate the Monitor component with fault tolerance techniques. Also the system becomes entangled if two friends report each other to be malicious through ALARM messages.

In [21], Tao Li et al proposed a very efficient technique for detection of compromised nodes in Wireless Sensor Network. The technique is based on a metric known as Data Transmission Quality. In this method, the source node calculates historical throughput and recent throughput for all the nodes that belong to the same group and are along the same communication path. Whenever the sender receives acknowledgement from the receiver, DTQ values for all the intermediate nodes is increased. Otherwise, value is decreased. Finally, compromised node is found by voting process. The method involves very less computation and is very effective since it considers both long-term and near-term behavior of nodes.

Drawbacks: The technique was designed for Wireless Sensor Networks where the number of nodes is very high. The technique assumes grouping in the network which is against the theme of MANETs. Also, sensor nodes have very little resources as compared to nodes in MANETs. This difference can be exploited to make the detection process more effective.

Table 2.1 Comparison of Existing solutions

S.No	Name	Type	Power Cons.	Bandwidth	Attacks handled	Drawbacks
1.	SAR	Authentication	High	No overhead	-	it needs different keys for different levels of security
2.	SAODV	Authentication	High	No overhead	DoS, Routing Attacks	Vulnerable, attacker can disguise as legitimate node
3.	ARAN	Authentication	High	No overhead	All Network layer attacks	Unrealistic assumption prior security coordination
5.	SCAN	Behavior	Low	High	Packet drop attacks	Assumes high node density Limited to simple packet drop attacks
6.	Watchdog and Pathrater	Behavior	Low	High	Packet drop attacks and data alteration attacks	Cannot detect false misbehavior
7.	CONFIDANT	Behavior	Low	High	-	No method to integrate monitoring component to false tolerance module
8.	C-Nodes Detection	Behavior	Low	Low	Packet drop attacks	Designed for WSN. Limited Packet drop attacks

2.6 Research gaps

Authentication based system cannot provide complete security. Keys can be stolen or guessed. Regardless of which keying scheme is used, it is possible that an adversary can crack the keys by brute-force search or reverse engineering of chips or programs on nodes. If instruction level source codes are available, then it will take far less time to locate the storage position of keys or find the keying schema. Even the key information is not available, nodes might be compromised too, i.e., the chip is hacked. Thus, compromised nodes will become an inevitable problem. Also, MANETs have certain challenges in key management due to lack of infrastructure, absence of dedicated routers and mobility of nodes, limited processing power and limitation of battery power, bandwidth and memory. Authentication based systems involve a lot computation which results in more power consumption. Also, these systems cannot identify the malicious node but can only detect. Therefore, one cannot guarantee complete security using only authentication techniques.

Behavior based system used along with authentication is preferred. In existing behavior based systems like Watchdog and pathrater, the node behavior values are interchanged between the nodes. Also it is not able to detect a selective forwarder. This exchange of node ratings consumes a large amount of bandwidth which is a limited resource in case of MANETs. Also these systems do not consider communication failure in the network which results in false positives. So a system which involves less communication between the nodes and have low rate of false positives is preferred. In [22], a very efficient technique is proposed but the technique is developed for Wireless Sensor Networks. But it can be used in MANETs with little modifications. Although the technique is limited to handle only packet drop attacks but it does so very efficiently.

The main requirement to ensure security in MANETS is to have an approach which should have properties to accurately detect malicious nodes with very less resource consumption and it should be self-stable against attacks.

Chapter 3

DTQ based Detection and Identification of Packet Forwarding Misbehavior in MANETs

This chapter describes the proposed DTQ based technique for misbehaving node detection and the IDS design based on this technique. Misbehaving nodes are detected based on their communication quality. First, misbehaving nodes are identified and then these nodes are blacklisted so that routing protocols avoid these nodes. The detailed procedure is discussed in section 3.3. Section 3.4 will discuss the proposed IDS design based on our technique.

3.1 Definitions

Neighbor: We use the term neighbor to refer to a node that is within wireless transmission range of another node. Likewise, neighborhood refers to all the nodes that are within wireless transmission range of a node.

Malicious node: A node is considered malicious either if it is dropping data packets completely or selectively or if it is modifying the data packets.

3.2 Assumptions

We assume wireless interfaces that support promiscuous mode operation. Promiscuous mode means that if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A. While promiscuous mode is not appropriate for all ad hoc network scenarios (particularly some military scenarios) it is useful in other scenarios for improving routing protocol performance [22].

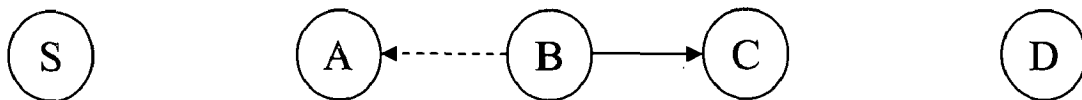


Figure 3.1 Promiscuous listening

When B forwards a packet from S toward D through C, A can overhear B's transmission and can verify that B has attempted to pass the packet to C. The solid line represents the intended direction of the packet sent by B to C, while the dashed line indicates that A is within transmission range of B and can overhear the packet transfer.

3.3 DTQ based detection of packet forwarding misbehavior

The process is divided into two steps: (1) Detection & Identification (2) Elimination.

3.3.1 Detection & Identification of malicious node

Malicious nodes are detected based on their communication quality. To measure a node's communication quality a new function is introduced known as Data Transmission Quality (DTQ). Each node calculates DTQ for all the neighbor nodes to which it has tried to communicate with. DTQ values change according to node behavior and is updated after each data burst ends. A data burst is defined as a stream of data packets transmitted between fixed time intervals.

The detection process works as follows: In MANETs a node can listen to its one hop neighbor. Therefore, after a node transmits a packet to its neighbor, it waits and listens to check whether the neighbor forwards it or not. It can be done by running the interfaces in promiscuous mode. To detect any alterations in data packet, the sending node stores the last packet sent and compares it with the packet forwarded by neighbor. In this way, any alterations in the packet can also be detected. Now, every node measures the number of packets forwarded by the neighbor nodes (D) and the total number of packets sent to that node (E). Thus D/E represents the fraction of packets successfully forwarded by neighbor node. If a node selectively forwards packets or start dropping packets after a certain period of time, its D/E value will not be able to reflect this change in the node's quality. Hence a stability factor is multiplied to make the detection process faster. Stability represents whether a node's behavior is improving or declining. To calculate stability, each node calculates recent throughput and historical throughput for its neighbor nodes. The ratio of these two quantities is stability of the node denoted as STB() (Section 3.3.3). "Data transmission quality" (DTQ) is defined as a function of STB(), D, E, and probability of error in the channel P(). Transmission error probability (p) reduces the effect of transmission failure due to network problems thereby preventing detection of false misbehavior.

$$DTQ = k \times \frac{D \times STB()}{p \times E} \quad (3.1)$$

A node keeps updating the DTQ values for its neighbor. Whenever the DTQ value falls below a certain threshold, the node is blacklisted. Figure 3.2 shows the detection process. The same process will be running at all the nodes in the network.

3.3.2 Elimination of malicious node

Whenever a node's DTQ value falls below the threshold, the node is isolated or eliminated from the network. For this, the monitoring node broadcasts the node id of the malicious node to all other nodes in the network so that the node is not further included in any routing path.

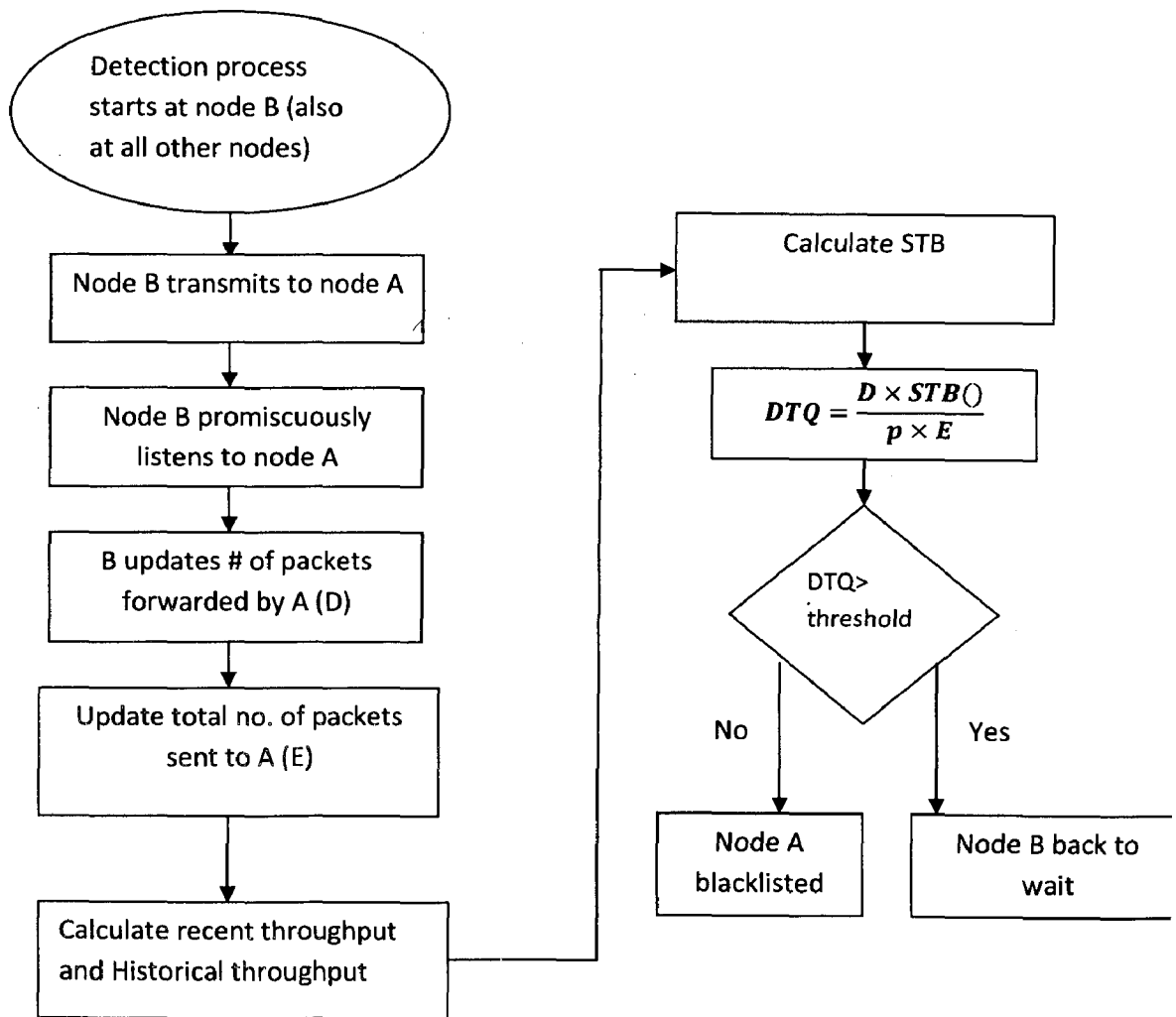


Figure 3.2 Identifying malicious nodes

3.3.3 Algorithmic details

STB(): the stability of a node, is a measure of how fast the transmission quality changes in a period of time. *STB()* function is defined as follows,

$$STB() = \begin{cases} \left[\frac{S(d,u)}{L(d,u)} \right]^\alpha = \left[\frac{\sum_{i=1}^{n/md_i} u_i}{\sum_{j=1}^{n/u_j} d_j} \right]^\alpha, & \text{if } \left[\frac{\sum_{i=1}^{n/md_i} u_i}{\sum_{j=1}^{n/u_j} d_j} \right]^\alpha < 1 \\ \left[\frac{S(d,u)}{L(d,u)} \right]^{1/\alpha} = \left[\frac{\sum_{i=1}^{n/md_i} u_i}{\sum_{j=1}^{n/u_j} d_j} \right]^{1/\alpha}, & \text{if } \left[\frac{\sum_{i=1}^{n/md_i} u_i}{\sum_{j=1}^{n/u_j} d_j} \right]^\alpha > 1 \end{cases} \quad (3.2)$$

where d_i and u_i represent the bytes successfully transmitted and the bytes attempted to be transmitted, respectively, when sending the past i^{th} data burst; $\alpha > 1$; n is a positive integer, which gives how many historical data sending statistics are kept by the node; m is a positive integer, and $n \% m = 0$. Here, m is referred to as `small_bucket_size` and n is referred to as `large_bucket_size`.

The quotient of $S(d,u)/L(d,u)$ reflects the quality trends in recent data transmission. If $S(d,u)/L(d,u) > 1$, it indicates that the data transmission quality is increasing; otherwise, the recent data transmission quality is decreasing. A step power function is used to amplify the impact of decreasing and reduce the impact of increasing of quality. For example, given $\alpha=4$ and $S(d,u)/L(d,u)=0.5$, then *STB()* will decrease the DTQ value over 90%.

In DTQ equation, D/E reflects the fraction of data successfully forwarded by a node. If a malicious node selectively forwards data, drops packets, or sends data to incorrect routing path, its D/E value decreases. The *STB()* function specifies the performance stability by comparing the success rate between short-term and long-term statistics. The instantaneous increase of statistical quality does not help to increase a node's DTQ value. However, the decrease of node's stability quickly decreases the DTQ value. When environment factors influence the data transmission, the value of function $1/p()$ reduces the impact to the accuracy of our measurement. An ideal assumption is that the $p()$ could totally erase the environment impact on DTQ.

Threshold: Threshold must be chosen according to the network behavior. Therefore, the best way to choose threshold is to take average of DTQ values of all nodes. Let N_{DTQ} be the set of nodes listed in DTQ table, $|N_{DTQ}|$ be the number of nodes, q_i be the DTQ value for node i . Threshold may be defined as,

$$Th = r \frac{1}{|N_{DTQ}|} \sum_{i \in N_{DTQ}} q_i \quad (3.3)$$

3.4 Proposed IDS Architecture and Design

The proposed approach can be implemented as an independent Intrusion detection system with Stand-alone architecture. In this architecture, every node participates in intrusion detection and response by having an IDS agent running on them. An IDS agent is responsible for detecting and collecting local events and data to identify possible intrusions, as well as initiating a response independently. The architecture is shown in Figure 3.3. Star sign represents IDS.

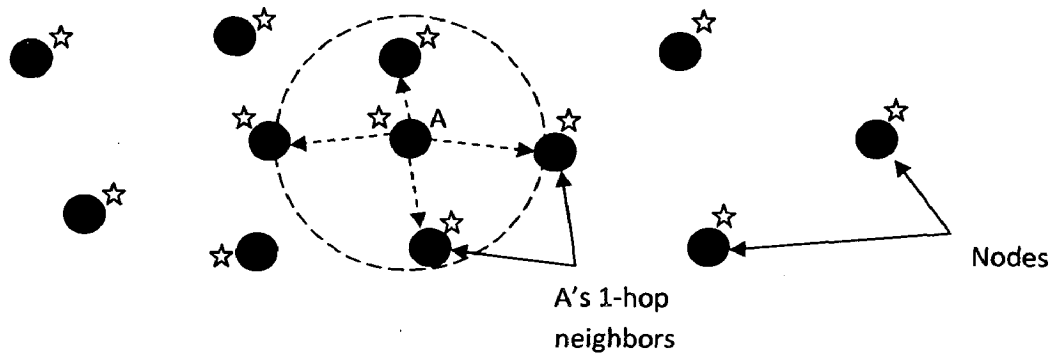


Figure 3.3 Distribution of IDS agents

3.4.1 System Design

Figure 3.4 shows the various modules of our IDS. Every node in the network will have similar design of the IDS. Therefore, our architecture is completely decentralized.

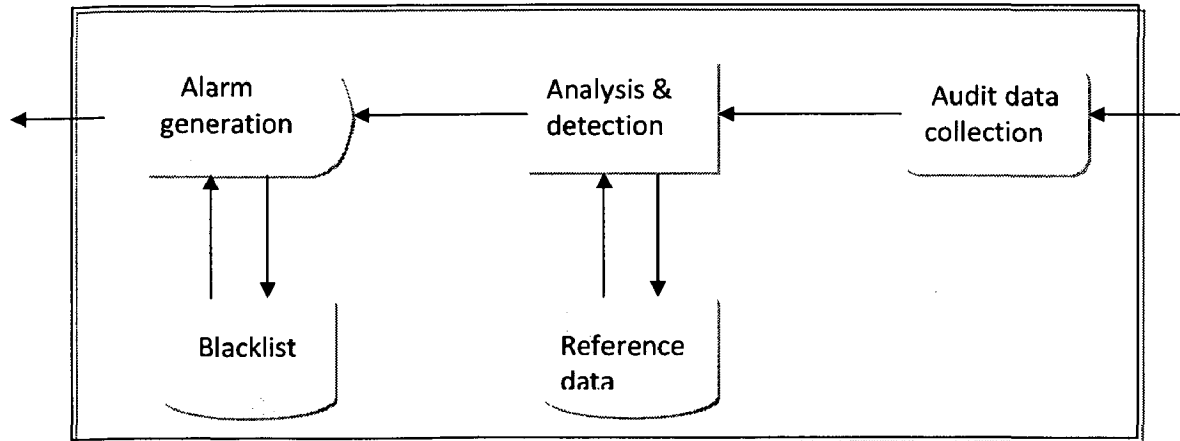


Figure 3.4 IDS design

Audit data collection: This module is used in the data collection phase. The data collected in this phase are analyzed by the intrusion detection algorithm to find anomalies. The collected data will contain the transmission statistics of the monitored node.

Analysis and detection: It is the main part of our Intrusion detection system. It is here that the algorithms to detect anomalous behavior are implemented. The data collected by the “Audit data collection” module will be analyzed here.

Reference data: The reference data storage stores information about profiles of normal behavior like DTQ thresholds etc. The profiles are updated when new knowledge about system behavior is available.

Alarm generation: This part of the system handles all output from the intrusion detection system. The output will be an automated response to an intrusion. All the other nodes in the network will be informed about the presence of malicious node in order to isolate it through this module. The isolated node will be added in the Blacklist.

Blacklist: The Blacklist storage will contain the list of malicious nodes. The nodes in this list will not be included in any routing path.

Figure 3.5 shows a diagram of our IDS and how it interacts with the TCP/IP protocol stack.

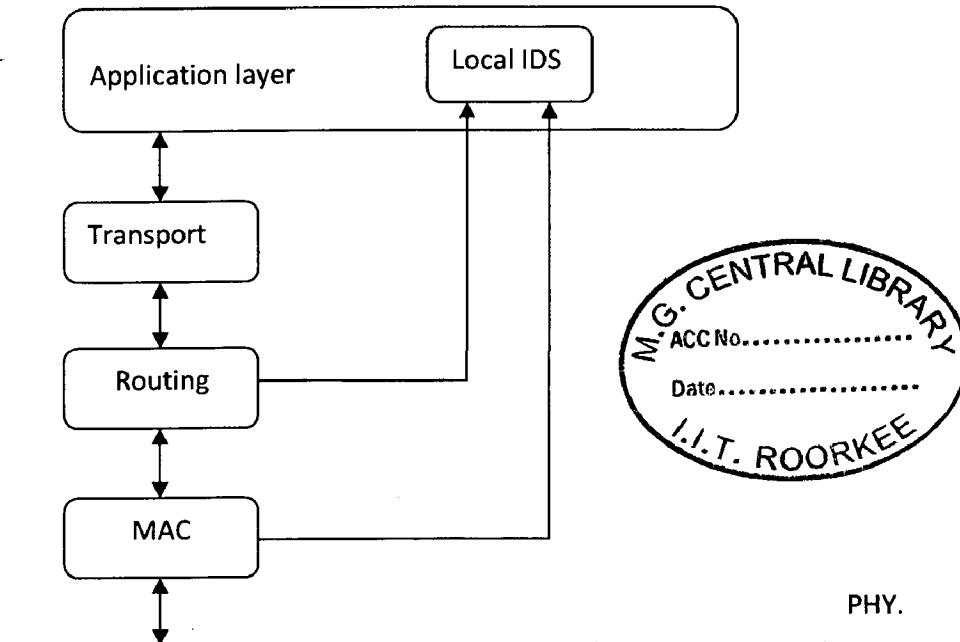


Figure 3.5 IDS and TCP/IP protocol stack

Our IDS is in the application layer. The “Data collection module” collects data from the MAC layer in the form of complete packets. This data is processed by the “Analysis and detection module” using the approach described in Section 3.3 to calculate DTQ values for its neighbors. These values are compared with the reference data to detect anomalies if any. If anomaly is detected, “Alarm generation module” will generate an alarm informing the network about the presence of malicious node. The node will be added in the Blacklist storage of all the nodes.

Chapter 4 Simulation and Implementation details

This chapter will provide the simulation and implementation details regarding our proposed approach. Section 4.1 will provide a brief overview of the simulator used i.e. Qualnet. Section 4.2 will discuss the simulation of attacks in Qualnet. Section 4.3 will discuss the simulation of the proposed approach and how the results are obtained.

4.1 Simulation environment

We have used Qualnet as the simulation environment. Qualnet is a network simulation tool that simulates wireless and wired packet mode communication networks. Qualnet Developer is a discrete event simulator used in the simulation of MANET, WiMAX networks, satellite networks, and sensor networks, among others. Qualnet has models for common network protocols that are provided in source form and are organized around the OSI Stack.

4.2 Simulation Parameters

Table 4.1 shows the simulation parameters used during the simulation.

Table 4.1 Simulation parameters

Parameters	Values Assigned
Routing Protocol	ODMRP
ODMRP refreshment interval	20 seconds
Radio type	802.11b
Packet size	512 bytes
Number of packets	1000
Traffic model	Multicast constant bit rate
Number of nodes	20
Mobility model of nodes	Random waypoint
Speed of nodes	5 m/s
Area	1000 m * 1000 m
Simulation time	2000 seconds

The simulation settings are as follows. The network consists of 20 nodes placed randomly within an area of 1000m x 1000 m. The random way point model is used as the mobility model. In this model, a node selects a random destination and moves towards that destination at a speed between the pre-defined maximum and minimum speed. The minimum speed for the simulations is 0 m/s while the maximum speed is 5 m/s. The packet size is 512 bytes. The traffic used is MCBR. The simulation time is 2000 seconds. The routing protocol used is ODMRP although the solution can be incorporated with any protocol.

4.3 Simulation of Attacks

This section provides the simulation details of various attack models. Section 4.3.1 provides the details for simulation of Blackhole attack. Section 4.3.2 provides the simulation details for Gray hole attack.

4.3.1 Simulation of Black hole attack

The black hole attack is simulated in two phases. First, the malicious node exploits the routing protocol to advertise itself as having a valid route to a destination node, even though the route is spurious, with the intention of intercepting the packets. In case of ODMRP, whenever a RREQ is received by a node, it waits for certain amount of time before forwarding that request in order to avoid collision. The malicious node does not wait before forwarding and makes itself a preferred choice in routing. This is also known as rushing attack. In second phase, the attacker consumes the packets and never forwards. Table 4.2 shows the design parameters for the simulation of the black hole attack.

Table 4.2 Attack Design Parameters

Number of Attackers	0-5
Attacks Starts at	0 sec
Attack ends at	2000 sec
Simulation time	2000 sec

Following changes are made in the files in order to simulate black hole attack:

In \include\node.h file, define a new flag which indicates whether a node is an attacker or not.

```
struct struct_node_str {  
  
    // a flag indicates whether the node is an attacker  
  
    BOOL attack_flag;  
  
    ..... }  
}
```

In \main\app_util.cpp file, in the function APP_InitMulticastGroupMembershipIfAny, set the delay to zero for attacking node to implement the rushing attack.

```
// if the node is a multicast member (denoted by letter 'G' in the modified  
configuration file then assign it a delay  
  
    if ((node->nodeId == srcAddr) && (node_type == 'G'))  
    {  
        // assign a simulated processing delay to the node  
        node->process_delay = delay_tmp;  
    }  
  
    // else if the node is a rushing attacker (denoted by letter 'A' in the modified  
configuration file  
  
    else if ((node->nodeId == srcAddr) && (node_type == 'A')) {  
  
        // set the attacker flag to TRUE  
  
        node->attack_flag = TRUE;  
  
        return;  
  
    }  
}
```

In `\libraries\wireless\src\multicast_odmr.cpp` file, in the function, `OdmrpHandleJoinQuery` set the delay to zero for attacking node to implement the rushing attack.

```
// This function is invoked once the node receives a Join Query packet
static void OdmrpHandleJoinQuery (OdmrpData *odmrp, Node *node, Message *msg) {
    // get the corresponding multicast source ID of this Join Query
    int sourceId = srcAddr & 0xff;

    // if the node is a rushing attacker
    if (node->attack_flag)
    {
        // forward Join Query with zero simulated processing delay
        NetworkIpSendPacketToMacLayerWithDelay (node, msg, DEFAULT_INTERFACE,
        ANY_DEST, 0);
    }

    // else if the node is an honest node
    else {
        // forward Join Query with non-zero simulated processing delay
        NetworkIpSendPacketToMacLayerWithDelay (node, msg, DEFAULT_INTERFACE,
        ANY_DEST, node->process_delay);
    }
}
```

multicast_odmrp.cpp (this file simulates the On-Demand Multicast Routing Protocol (ODMRP))

// This function is invoked once the node receives a data packet

```
static void OdmrpHandleData (Node *node, Message *msg) {  
    // if the node is a forwarding group member  
    if (OdmrpLookupFgFlag (mcastAddr, &odmrp->fgFlag) ) {  
        // if the node is a blackhole attacker  
        if (node->attack_flag) {  
            // then drop the data packet  
            MESSAGE_Free(node, msg);  
            return;  
        }  
    }  
}
```

4.3.2 Simulation of Gray hole attack

The simulation of Gray hole attack is very much similar to the simulation of Black hole attack. The only difference is, in Gray hole attack all packets are not dropped rather packets are dropped with a certain probability. The only additional change required is in multicast_odmrp.cpp file inside OdmrpHandleData. Add following lines:

```
if (OdmrpLookupFgFlag (mcastAddr, &odmrp->fgFlag) ) {  
    // if the node is a blackhole attacker  
    if (node->attack_flag) {  
        if ((rand()%10) > forwarding_rate)  
            MESSAGE_Free(node, msg); return;}  
        else forward_packet;
```

4.4 Simulation of the proposed technique

For simulation, the standard ODMRP protocol has been modified to incorporate the solution. Following files have to be modified:

1. \include\node.h: Define new variables inside structure node like DTQ table etc.
2. \main\app_util.cpp: Initialize variables defined in struct node inside function APP_InitMulticastGroupMembershipIfAny.
3. \libraries\wireles\src\multicast_odmrp.cpp: Inside function OdmrpHandleJoinQuery, join the network only if the node's DTQ value is above threshold. After receiving data from neighbor modify D and E value. Update DTQ also if end of data burst. (See Figure)

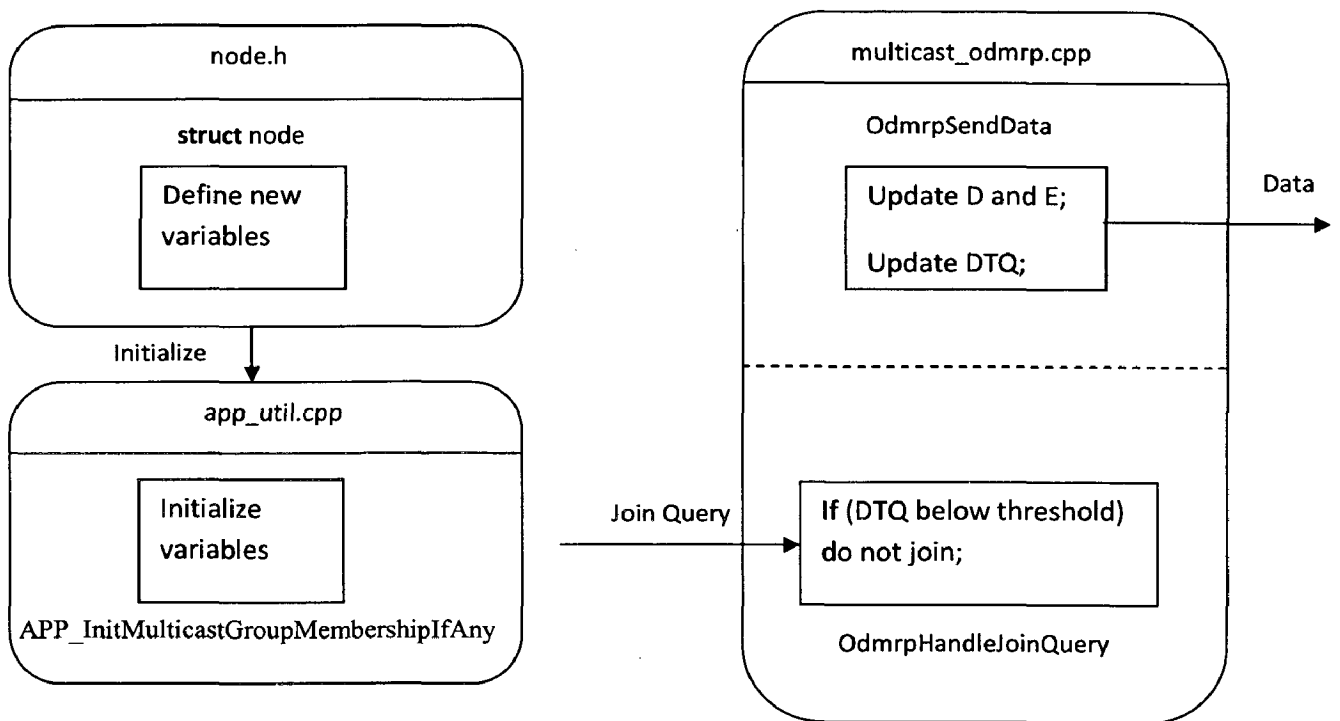


Figure 4.4 Modified files with their functions

The flow of events is as follows:

1. Each node maintains a DTQ table which contains DTQ values of all the neighboring nodes.

2. Whenever a node forwards a packet to its neighbor, it updates the value of total number of sent packets (E).
3. In order to find out whether that packet is forwarded successfully or not by the neighbor, we check the attack flag of that node. Note that in reality the node has to listen to its neighbor promiscuously to find this. But in case of simulation we can find this information by reading the attack flag of that node. Results will not be affected by this. If packet is successfully forwarded increment D otherwise keep it same.
4. Calculated DTQ if end of data burst.
5. If DTQ value falls below the threshold, node is blacklisted.
6. Inform other nodes in the network about the presence of malicious node.

Chapter 5 Simulation Results and Analysis

This chapter shows the simulation results and discusses them. The proposed approach is tested for Packet drop attacks like Black hole and Gray hole. The simulation is carried out several times varying some parameters while keeping others constant. The results are depicted in graphs for varying simulator settings and then the results thus obtained are discussed.

5.1 Performance metrics

The performance is evaluated based on three metrics: False positives, Average throughput and end-to-end delay.

5.2 Choice of DTQ threshold

DTQ threshold is chosen to be 0.5 throughout the simulation. It means a node whose forwarding rate is below 50% is considered as suspicious. This value seems reasonable as a node dropping packets more than 50%, although legitimate, has to be removed from the routing path. In case of Black hole, the mechanism works well with other thresholds also.

5.3 Results and Discussion for Packet drop attacks

5.3.1 Varying small_bucket_size and large_bucket_size

Figure 5.1 shows the effect of small_bucket_size on malicious node detection rate. Figure 5.2 shows the effect of large bucket size on the detection time.

Table 5.1 Simulation settings

DTQ threshold	0.5 (constant)
No. of Attackers	5 (constant)
Small bucket size	5-30 (varying)
Large bucket size	100 (constant)

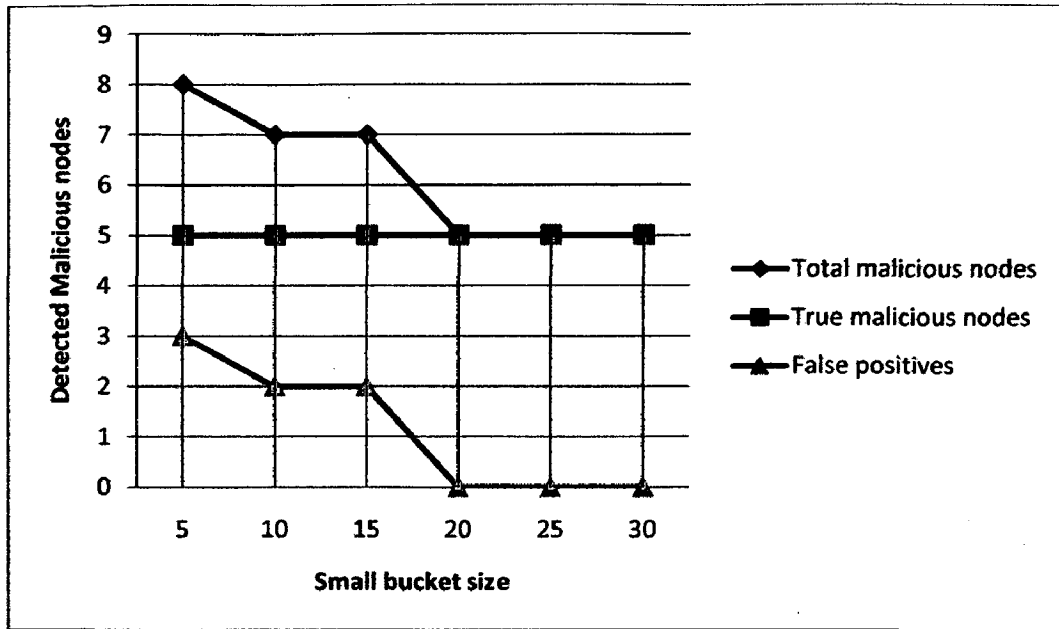


Figure 5.1 varying small bucket size

Discussion: All malicious nodes are detected correctly. It is noticed that when the `small_bucket_size` is low, false positives are high. This is expected, because a low `small_bucket_size` means that the behavior of the nodes is measured based on very few transmissions. As the bucket size becomes more in tune to the network's current settings of behavior, false positives become almost nil.

Table 5.2 Simulation settings

DTQ threshold	0.5 (constant)
No. of Attackers	5 (constant)
Small bucket size	30 (constant)
Large bucket size	100,200,300
Attack starts at	0 sec

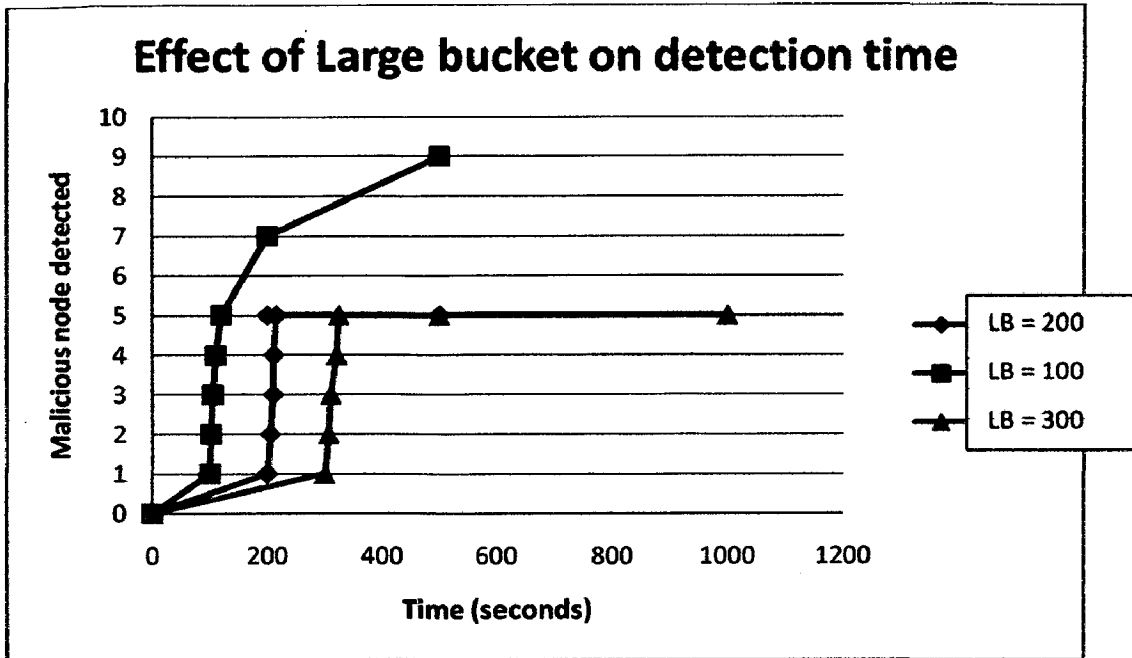


Figure 5.2 varying large bucket size

The second effect is that of having a low large bucket size. This also displays the same behavior as that above. This is because long-term-bucket measurements aim to capture the long term behavior of nodes. Say, historically, a node has an 80 percent acknowledgment rate. Then, using near-term buckets, we measure if the node is consistent with its “character” of 80 percent. If not, the activity is of interest and may be marked for a vote-request trigger. But, if the period over which history is measured is lowered (by reducing the number large bucket size), it does not present a true measure of regular node behavior.

A manifestation of using a larger bucket size is a larger “training time” before nodes can detect malicious nodes. Figure 5.2 shows this. In general, its value is kept 4-5 times the small bucket.

5.3.2 Varying DTQ threshold

Figure 5.3 shows the results of varying the DTQ threshold in case of Black hole attack. Figure 5.4 shows the effect in presence of Gray holes in the network.

Table 5.3 Simulation settings

DTQ threshold	0.1 - 0.5 (varying)
No. of Attackers	5 (constant)
Small bucket size	30 (constant)
Large bucket size	100 (constant)

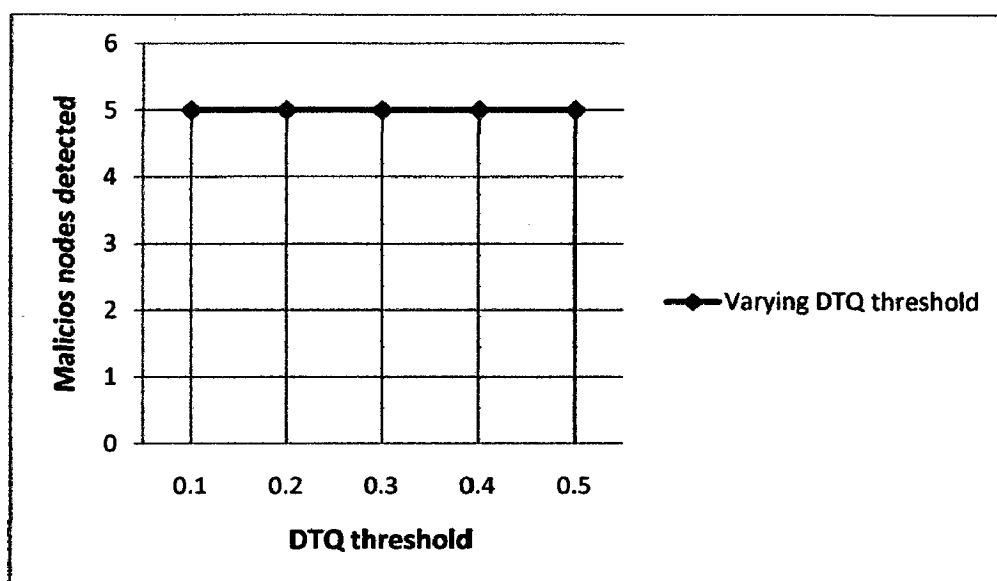


Figure 5.3 varying DTQ threshold

Discussion: It is seen from graph that there is no significant effect of the change of thresholds on malicious node detected. This is because our testing logic is right now a “selective forwarder” that does not forward any packet. That is, its forwarding rate is consistently 0. While threshold matters for reduction of false positive detection of non-malicious nodes due to transmission failures, threshold is not a sole factor that reduces false positive detection percentage. It is to be used in combination with other settings like bucket size etc. For the current scenario, any value of threshold works well, though conducting more tests may reveal an optimal value.

Table 5.4 Simulation settings

DTQ threshold	0.5,0.4
No. of Attackers	5 (constant)
Small bucket size	30 (constant)
Large bucket size	100 (constant)
Forwarding rate of attackers	0.1-0.6 (Varying)
Transmission error probability	0.2 (constant)

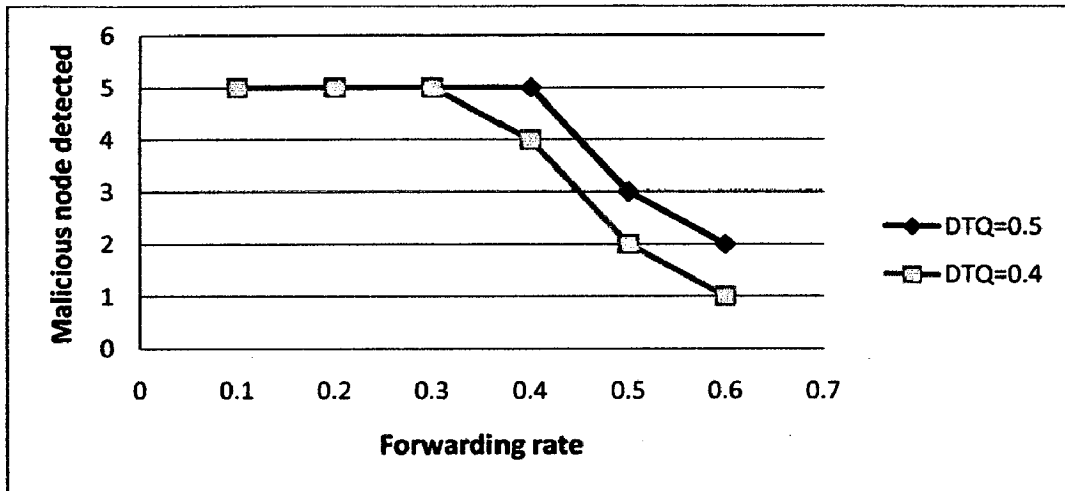


Figure 5.4 Malicious node detected Vs. Forwarding rate

Discussion: It is seen from graph that there is significant effect of the change of thresholds on malicious node detected as expected. It can be seen in the graph that some selective forwarders with forwarding rate near DTQ threshold is not detected as malicious. But our mechanism still serves the purpose of forcing it to forward at a minimum rate.

5.3.3 Varying transmission_error_probability

Figure 5.5 shows the results of varying the transmission error probability in the channel.

Table 5.5 Simulation settings

DTQ threshold	0.1 - 0.5 (varying)
No. of Attackers	5 (constant)
Small bucket size	10,20
Large bucket size	100 (constant)
Transmission error probability	0-0.4 (varying)

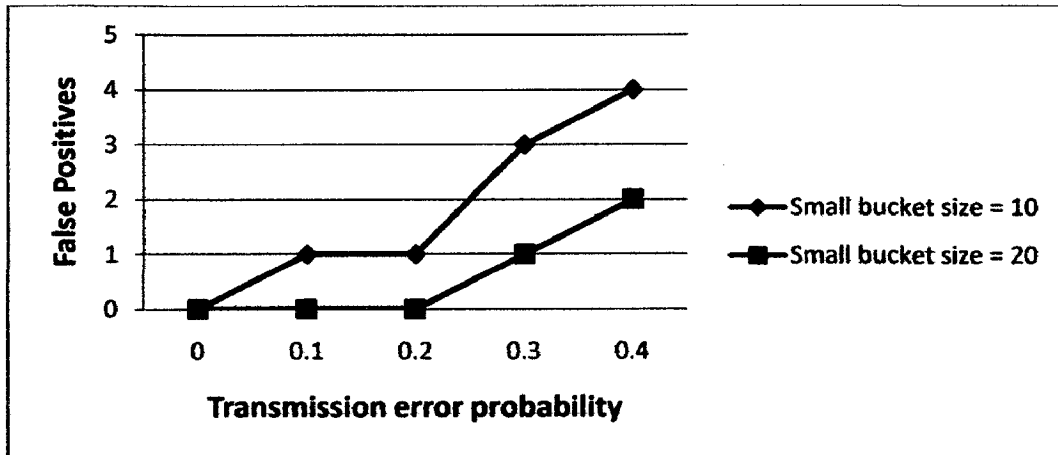


Figure 5.5 Varying transmission error probability

Discussion: False positives are zero for low value of transmission error probability which is generally the case in MANETs. Number of false positives increases with increase in transmission error probability but its effect can be countered by increasing the size of small bucket as shown in the graph. The reason for that is that larger small bucket size means the behavior of a node is measured for more number of transmissions. This provides a legitimate node more time to prove its innocence.

5.3.4 Varying number_of_attackers

Figure 5.6 shows the results of varying number of attackers in the network on the throughput.

Table 5.6 Simulation settings

DTQ threshold	0.5 (constant)
No. of Attackers	0-5 (varying)
Small bucket size	30 (constant)
Large bucket size	100 (constant)

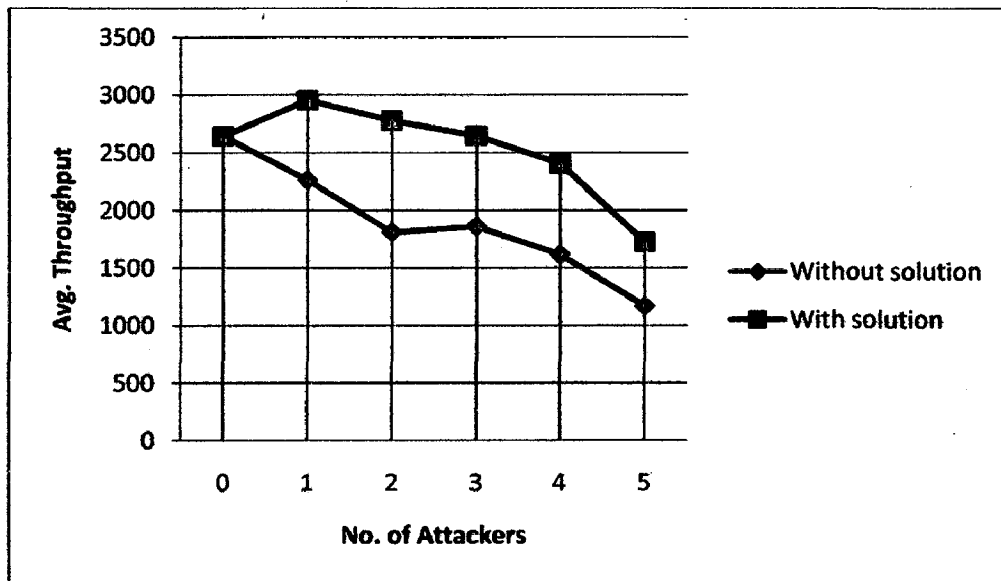


Figure 5.6 Throughput vs. No. of Attackers

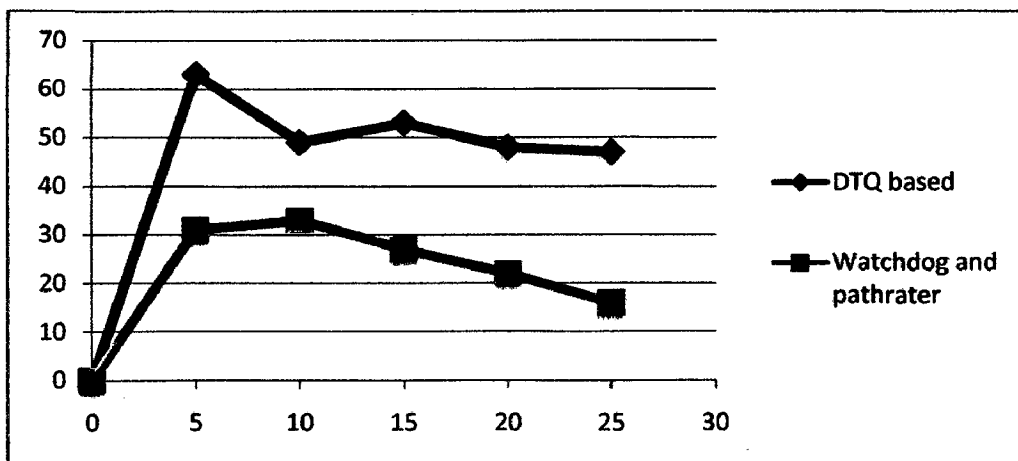


Figure 5.7 Comparison in terms of percentage improvement in throughput

Figure 5.8 shows the results of varying number of attackers in the network on the end-to-end delay in the network.

Table 5.7 Simulation settings

DTQ threshold	0.5 (constant)
No. of Attackers	0-5 (varying)
Small bucket size	30 (constant)
Large bucket size	100 (constant)

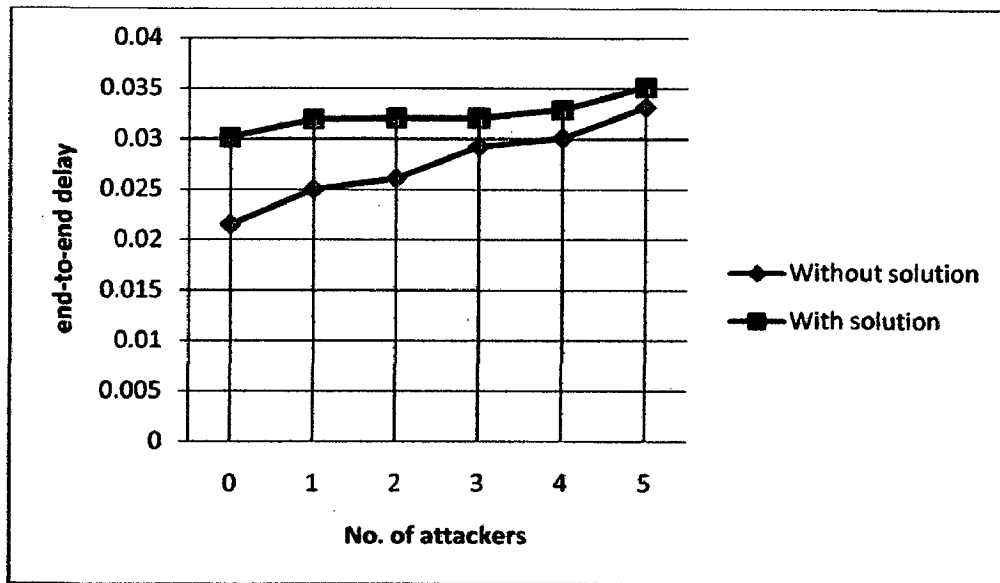


Figure 5.8 end-to-end delay vs. No. of Attackers

Figure 5.8 shows the results of varying number of attackers in the network on malicious node detection rate.

Table 5.8 Simulation settings

DTQ threshold	0.5 (constant)
No. of Attackers	0-5 (varying)
Small bucket size	30 (constant)
Large bucket size	100 (constant)

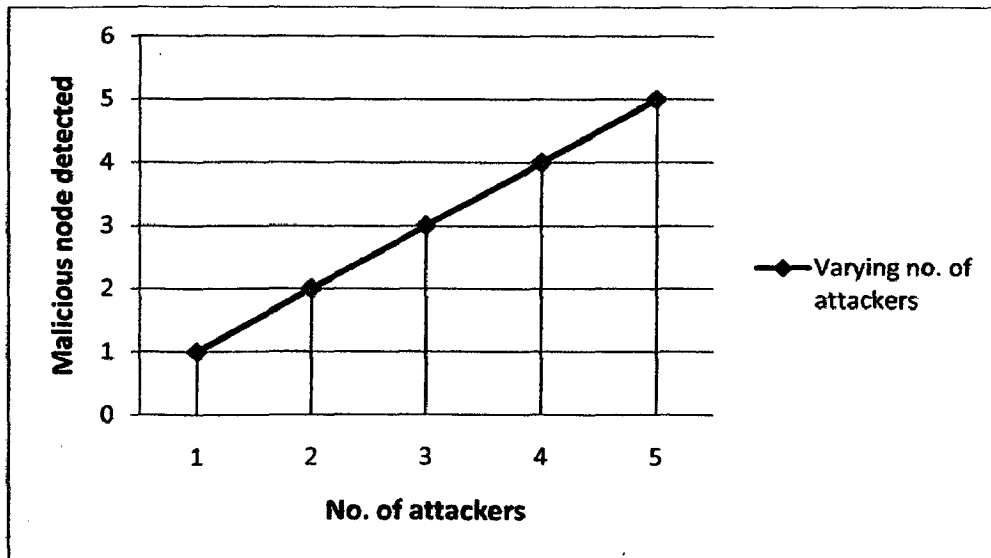


Figure 5.9 Malicious node detected vs. No. of Attackers

Discussion: As the number of attackers in the network is increasing network throughput is decreasing as expected in both the cases i.e. without IDS and with IDS. But graph in Figure 5.7 shows significant improvement in the Average throughput of the network after the IDS is running. Average throughput has increased by almost 50%. End-to-end delay has increased marginally as shown in the graph 5.8 after the IDS is running. But the overhead is very insignificant. Another factor is false positives. Number of attackers has no effect on number of false positives. This was expected since the method detects malicious nodes solely on the basis of node's communication quality which will not be affected by having more number of attackers in the network.

5.4 Discussion for Data alteration attacks

The proposed DTQ based mechanism is developed for detecting Packet drop attacks. But since it stores the last forwarded packet in buffer for comparison, any alteration in the packet can also be detected. In this way, all those attacks which are based on tampering the packet data like Neighbor attack, Message tampering etc can be accurately detected by the proposed mechanism. But, it increases the memory consumption by a very insignificant amount.

5.5 Comparison with Existing techniques

Table 5.9 shows the comparison between existing monitoring techniques and the proposed DTQ based technique. The comparison is done based on four factors: False positives, Bandwidth consumption, Power consumption and Attacks handled.

Table 5.9 Relative comparison of existing malicious node detection techniques

Name	False positives	Bandwidth consumption	Power consumption	Attacks handled
Watchdog and pathrater	High	High	High	Packet drop and Data alteration
SCAN	High	High	High	Packet drop
C-Node Detection	Low	Moderate	Moderate	Packet drop
DTQ based Detection	Low	No overhead	High	Packet drop and Data alteration

False positives: Watchdog and pathrater do not consider any transmission error in the channel which leads to false positives. The same problem is there with SCAN. Both, C-node detection and DTQ based detection consider transmission error in the channel.

Bandwidth consumption: Watchdog and pathrater involves exchange of node ratings among the nodes which lead to increased bandwidth consumption. In SCAN, new control packets such as MREQ, MREP etc needs to be send which increases bandwidth consumption. In C-node detection the only bandwidth overhead is during the voting process which is not required in case of DTQ based detection.

Power consumption: Watchdog, SCAN and DTQ based detection requires comparison between the last sent packet and overheard packet. This increases power consumption by a little amount. The calculation of DTQ is based on a very light theorem, so it does not cost any significant battery power and CPU.

Attacks handled: Watchdog and pathrater detects a misbehaving node on the basis of fraction of packets successfully transmitted (D/E value), so it is not able to detect those malicious nodes which started dropping packets after a certain period of time. But since our method considers both the fraction of packets successfully transmitted and a node's near term behavior, such malicious nodes will be detected. This fact can be seen in figure 5.11.

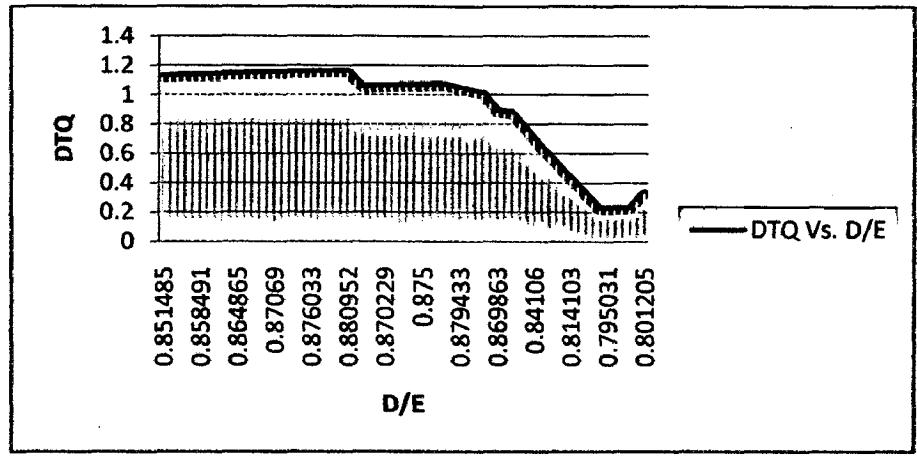


Figure 5.11 DTQ Vs D/E

Although the D/E value of a node is not below threshold but its DTQ value has dropped suddenly because of its near term behavior. C-node detection do not suffer from this problem but since it doesn't store any forwarded packets, so it is not able to detect any alterations in the packet.

5.6 Analysis of the approach

The set of points against which to measure the effectiveness of our solution is discussed in this section.

1. The mechanism for malicious node detection has already been discussed in the Chapter 3.
2. The proposed mechanism is time-continuous i.e. a node that started as a legal node, but was compromised by another malicious node at a later time can also be recognized. The solution decides a node's quality only on the basis of transmission behavior. The mechanism is also dynamic; every node regularly checks its neighbor statistics to determine abnormal behavior.

3. The proposed mechanism is truly distributed. Each node in the network or any number of nodes in the network can be configured to assume the responsibility of detecting abnormal behavior.
4. The solution is scalable and easy to employ. Nodes can join in the security solution by running the solution program code. No other additional infrastructure is necessary.
5. The solution can be implemented as an Intrusion Detection System or an existing protocol can be modified to incorporate the mechanism.
6. An additional advantage of this method is that the computation theorem itself is lightweight, and the periodicity with which it is calculated is configurable.
7. The solution is able to detect all kinds of packet drop attacks like black hole, gray hole etc. In addition to that, Data alteration attacks can also be detected.

Chapter 6 Conclusions and Future Work

We aimed to determine a method to identify malicious or compromised nodes in a MANET environment. We proposed a system in which anomalies in behavior is defined quantitatively by observing data exchange activity. A nodes communication quality is defined in terms of its long term and short term data exchange pattern. It is able to detect all kind of packet drop attacks along with data alteration attacks. The solution involves simple computation and is very effective in terms of detecting malicious nodes since it considers both the near-term and long-term behavior thereby reducing false positives. Also it does not involve huge exchange of data among the nodes such as node ratings etc for the process of detection. Therefore, the detection process is suitable for MANETs in terms of battery power and bandwidth constraints.

The proposed solution has been simulated in Qualnet. The data collected has shown that our proposed system works well. Our solution can detect malicious nodes with has zero false negatives i.e all the malicious nodes are detected accurately. The number of false positives is zero in case of Blackhole attack and is very reasonable in case of Gray hole.

6.1 Suggestions for Future Work

This section discusses a few areas where the current work can be taken further.

1. Test the approach with more mobility models.
2. Our system uses a “set” threshold for the network, and this threshold is never changed. Updating DTQ threshold based on the network weather or behavior may make the system more robust and reduce false positives. One possible method is discussed in third chapter.
3. The technique can be used to detect a variety of attacks with little modifications. It will require building more attack models (worm-hole, sybil etc) and test the mechanism on them.
4. Promiscuous listening may lead to problems [18]. Therefore, DTQ can be calculated using ACK received from the neighbor. Although this will limit the approach to only packet drop attacks. But it will prevent our system from Blackmail attack.

REFERENCES

- [1] Sunita Sahu and Sushir K. Shadily, "A Comprehensive survey on Intrusion Detection in MANETs," in *International Journal of Information Technology and Knowledge Management*, Volume 2, No. 2, July-December 2010, pp. 305-310.
- [2] Y. Xiao, X. Shen and D.-Z. Du (Eds.), "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks," in *Springer*, 2006, pp. 411-418.
- [3] S. Yi and R. Kravets, "Composite Key Management for Ad Hoc Networks," in *Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, June 2004, pp. 52-61.
- [4] Vikram Gupta, Srikanth Krishnamurthy and Michalis Faloutsos. "Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks," in *Proceedings of the IEEE Military Communications Conference*, October 2002, pp. 1118-1123.
- [5] Payal N. Raj and Prashant B. Swadas, "DPRAODV: A Dynamic Learning System Against Blackhole Attack in AODV based MANET," in *IJCSI International Journal of Computer Science Issues*, 2009, pp.54-59.
- [6] Oscar F. Gonzalez, Godwin Ansa, Michael Howarth and George Pavlou. "Detection and Accusation of Packet Forwarding Misbehavior in Mobile Ad-Hoc networks", in *Journal of Internet Engineering*, 2008, pp. 202-212.
- [7] Satria Mandala, Md. Asri Ngadi, A.Hanan Abdullah. "A Survey on MANET Intrusion Detection". *International Journal of Computer Science and Security*, Vol. 2(1), 2008, pp. 1-11.
- [8] P. Brutch and C. Ko, "Challenges in Intrusion Detection for Wireless Ad-hoc Networks," *Proceedings of 2003 Symposium on Applications and the Internet Workshop*, January 2003, pp. 368-373.
- [9] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion Detection in Wireless Ad Hoc Networks," *IEEE Wireless Communications*, Vol. 11, Issue 1, February 2004, pp. 48-60.
- [10] Elizabeth M. Belding-Royer and Charles E. Perkins, "Multicast Operation of the Ad Hoc On-Demand Distance Vector Routing Protocol," in *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, August 1999, pp. 207-218.
- [11] Sung J. Lee, William Su, and Mario Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *Mobile Networks and Applications*, Kluwer Academic Publishers, December 2002, pp. 441-453.
- [12] M. G. Zapata, "Secure Ad hoc On-Demand Distance Vector (SAODV) Routing," *IETF Internet Draft*, 2001.

-
- [13] M. G. Zapata, "Secure Ad hoc On-Demand Distance Vector Routing," *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3), 2002, pp. 106-107.
- [14] Y.-C. Hu, A. Perrig and D. B. Johnson, "Efficient Security Mechanisms for Routing Protocols," *Network and Distributed System Security Symposium, NDSS '03*, San Diego, USA, 2003, pp. 57-73.
- [15] S. Yi, P. Naldurg and R. Kravets, "A Security-Aware Routing Protocol for Wireless Ad Hoc Networks," in *6th World Multi-Conference on Systemics, Cybernetics and Informatics*, Florida, USA, 2002.
- [16] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A Secure Routing Protocol for AdHoc Networks," in *10th IEEE International Conference on Network Protocols (ICNP'02)*, Paris, France, 2002, pp. 78-89.
- [17] Oscar F. Gonzalez, Godwin Ansa, Michael Howarth and George Pavlou. "Detection and Accusation of Packet Forwarding Misbehavior in Mobile Ad-Hoc networks", in *Journal of Internet Engineering*, 2:1, 2008.
- [18] S. Marti, T. J. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proceedings of the 6th annual international conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, United States, 2000, pp. 255-265.
- [19] N. Pissinou, T. Ghosh and K. Makki, "Collaborative trust-Based Secure Routing in Multihop Ad Hoc Networks," in *Third International IFIP-TC6 Networking Conference*, Athens, Greece, 2004, pp. 1446-1451.
- [20] S. Buchegger and J.-Y. L. Boudec, "Performance Analysis of the CONFIDANT Protocol," *Cooperation Of Nodes-Fairness In Dynamic Ad-hoc NeTworks*. Technical Report (IC/2002/01), EPFL I&C, Lausanne, Jan 21 2002.
- [21] Tao Li, Min Song and Mansoor Alam. "Compromized sensor node detection: A quantitative approach," in *IEEE International Conference on Distributed Computing Systems*, 2008, pp. 352-357.
- [22] D. Johnson, D. A. Maltz, and Broch. "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *Mobile Ad-hoc Network (MANET) Working Group*, IETF, October 1999.