
ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

# EFFECTS OF SECURITY ATTACKS ON PERFORMANCES OF MANET MULTICAST ROUTING PROTOCOL

**By**

**Kalechristos Abebe**

A thesis submitted to the school of Graduate studies of Addis Ababa University, Addis Ababa Institute of Technology in partial fulfillment of the requirements for the degree of Masters of Science in Computer Engineering

**Advisor**

**Dr. Sreenivas Nune**

JUNE 2014

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL ELECTRICAL AND COMPUTER ENGINEERING

# EFFECTS OF SECURITY ATTACKS ON PERFORMANCES OF MANET MULTICAST ROUTING PROTOCOL

**By**
**Kalechristos Abebe**

**Advisor**

**Dr. Sreenivas Nune**

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL ELECTRICAL AND COMPUTER ENGINEERING

# EFFECTS OF SECURITY ATTACKS ON PERFORMANCES OF MANET MULTICAST ROUTING PROTOCOL

**By**
**Kalechristos Abebe**

APPROVAL BY BOARD OF EXAMINERS

| | |
|---|---|
| Chairman Dept. Graduate Committee | _____ Signature |
| Dr.Sreenivas Nune<br>Advisor | _____ Signature |
| _____<br>Internal Examiner | _____ Signature |
| _____<br>External Examiner | _____ Signature |

# ABSTRACT

Ad hoc networks are showing wider acceptance in the future trend of wireless system. It became clear that group-oriented communication is one of the key application classes in Mobile Ad Hoc Network environments, where several multi cast routing protocols are proposed. These routing protocols assume non adversarial environment and do not take security issues into account in their design The demand put on the wireless system is challenging the current design of secured system.

Mobile ad hoc networks are prone to security attack than other networks due to its unique characteristics. Research works are going on which help protect ad hoc networks from malicious node behaviors, the demand put on security mechanisms is a challenge due its bandwidth and energy requirements. This study considered to model attack in the simulator for varied properties of multicast communication and performed analysis

We arranged simulation set up to show increase in group communication size by increasing number of receiver nodes decreases the impact of attacker nodes. The effect of attacker node position being near sender and near receive is also analyzed where being near sender more serious than near receiver

In this thesis we thoroughly analyzed the performance of multi cast routing protocol under security attack. We incorporated black hole and jelly fish attacks in Protocol for Unified Multicasting through Announcements by modifying its source code. The design is tested on Network Simulator NS-2. Simulation results show that the presence of attacker node has serious effect on the performance protocol, where analysis showed the packet delivery fraction, good put and end to end delay were affected, the more the number of attacker nodes , the more impact of attack

***Keywords*** ***Mobile Ad Hoc Network, Security Attack, Multicast Routing***

## Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Dr.Sreenivas Nune for the continuous support of my thesis work for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank Addis Ababa University, Addis Ababa Institute of Technology for giving me a chance to study.

## Declaration

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been fully acknowledged.

Name: Kalechristos Abebe Signature: _____

Place: Addis Ababa

Date of submission: June 2014

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Sreenivas Nune Signature: _____

Advisor's Name

## Table of Contents

## List of Tables

# List of Figures

# Acronyms

| | |
|---|---|
| CBR | Continuous bit rate |
| CMU | Carnegie Mellon University |
| DCF | Distributed Coordination Function |
| DM | Dynamic mesh |
| DSDV | Destination sequenced distance vector routing |
| ERS | Expanded Ring Search |
| FG | Forwarding Group |
| FIFO | First in first out |
| FSR | Fisheye state routing |
| FTP | File Transfer Protocol |
| GPS | Global positioning system |
| ID | Identification |
| IP | Internet Protocol |
| LAN | Local Area Network |
| MA | Multicast Announcement |
| MAC | Medium access control |
| MANET | Mobile ad hoc network |
| MAODV | Multicast Ad hoc on demand distance vector routing |
| MIT | Massachusetts Institute of technology |
| NRL | Normalized Routing Load |
| NS | Network simulator |
| ODMRP | On demand multi cast routing protocol |
| PAST | Progressively Adapted Sub-Tree |
| PBM | Positon based multicasting |
| PDF | Packet Delivery Fraction |
| PUMA | Protocol for Unified Multicasting through Announcements |
| RREP | Route reply |
| RREQ | Route request |
| SRMP | Source Routing-based Multicast |
| TCP | Transmission control protocol |

| | |
|---|---|
| TTL | Time To Live |
| UDP | User datagram protocol |

# Chapter One
# Introduction

Wireless mobile ad hoc networks consist of mobile nodes interconnected by wireless multi-hop communication paths. Unlike conventional wireless networks, ad hoc networks have no fixed network infrastructure or administrative support. The topology of such networks changes dynamically as mobile nodes join or depart the network or radio links between nodes become unusable.

Designing a perfect security protocol for ad hoc network is a challenging task due to its unique characteristics such as, lack of central authority, frequent topology changes, rapid node mobility, shared radio channel and limited availability of resources

This thesis is a contribution in the field of security analysis on mobile ad-hoc networks multicast routing protocol. Limitations of the mobile nodes have been studied in order to design a secure multi cast routing protocol that thwarts different kinds of attacks. Our approach is based on the one mesh based multi cast routing protocol PUMA; the most popular multi cast routing protocol.

In this chapter, we will introduce wireless ad hoc networks, and discuss their applications and general overview of the thesis

## 1.1 Ad hoc networking

Conventional wireless networks require as prerequisites a fixed network infrastructure with centralized administration for their operation. In contrast, so called (wireless) mobile ad hoc networks, consisting of a collection of wireless nodes, all of which may be mobile, dynamically create wireless network amongst themselves without using any such infrastructure or administrative support [1]. Ad hoc wireless networks are self-creating, self-organizing, and self-administering. They come into being solely by interactions among their constituent wireless mobile nodes, and it is only such interactions that are used to provide the necessary control and administration functions supporting such networks. Mobile ad hoc networks offer unique benefits and versatility for certain environments and certain applications. Since no fixed infrastructure, including base stations, is prerequisite, they can be created and used

‘‘any time, anywhere.’’ Such networks could be intrinsically fault-resilient, for they do not operate under the limitations of a fixed topology. Indeed, since all nodes are allowed to be mobile, the composition of such networks is necessarily time varying. Addition and deletion of nodes occur only by interactions with other nodes; no other agency is involved. Such perceived advantages elicited immediate interest in the early days among military, police, and rescue agencies in the use of such networks, especially under disorganized or hostile environments, including isolated scenes of natural disaster and armed conflict. See Fig. 1.1 for a conceptual representation. In recent days, home or small office networking and collaborative computing with laptop computers in a small area (e.g., a conference or classroom, single building, convention center, etc.) have emerged as other major areas of application. These include commercial applications based on progressively developing standards such as Bluetooth [32] as well as other frameworks such as Piconet, HomeRF Shared Wireless Access Protocol, etc. In addition, people have recognized from the beginning that ad hoc networking has obvious potential use in all the traditional areas of interest for mobile computing.

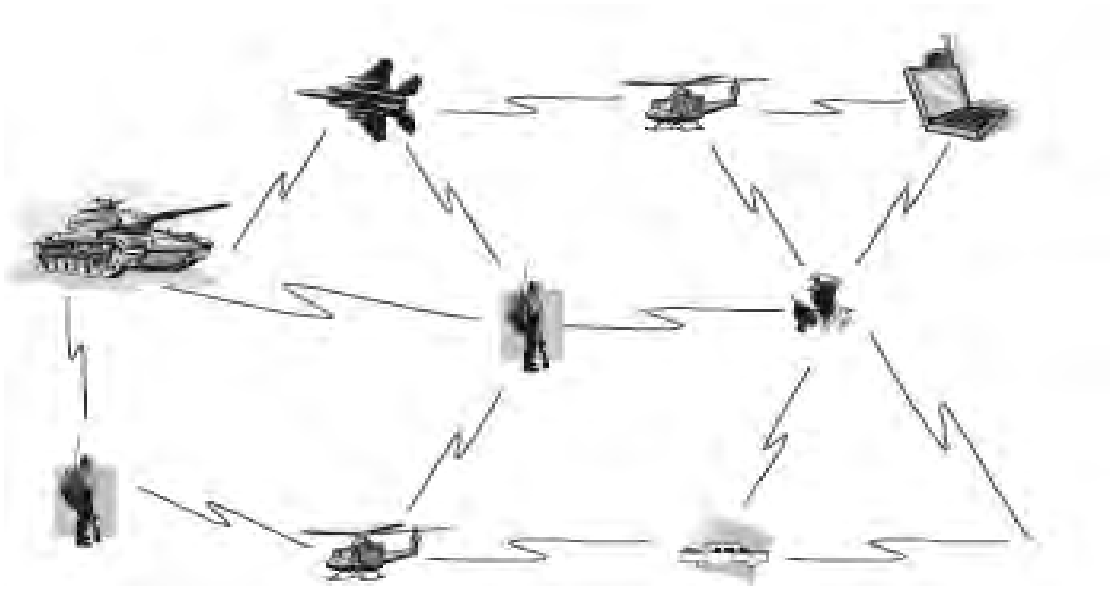


Fig. 1.1 Conceptual representation of ad-hoc network

Mobile ad hoc networks are increasingly being considered for complex multimedia applications, where various quality of service (QoS) attributes for these applications

must be satisfied as a set of predetermined service requirements. In addition, because of the use of the ad hoc networks for military or police use, and of increasingly common commercial applications, various security threats and measures need to be addressed.

Mobile ad hoc networking emerged from studies on extending traditional Internet services to the wireless mobile environment. All current works, as well as this thesis, consider the ad hoc networks as a wireless extension to the Internet, based on the ubiquitous IP networking mechanisms and protocols. Today's Internet possesses an essentially static infrastructure where network elements are interconnected over traditional wire-line technology, and these elements, especially the elements providing the routing or switching functions, do not move. In a mobile ad hoc network, by definition, all the network elements move. As a result, numerous more stringent challenges must be overcome to realize the practical benefits of ad hoc networking.

The absence of a fixed infrastructure for ad hoc networks means that the nodes communicate directly with one another in a peer-to-peer fashion. The mobility of these nodes imposes limitations on their power capacity, and hence, on their transmission range; indeed, these nodes must often satisfy stringent weight limitations for portability. Mobile hosts are no longer just end systems; to relay packets generated by other nodes, each node must be able to function as a router as well. As the nodes move in and out of range with respect to other nodes, including those that are operating as routers, the resulting topology changes must somehow be communicated to all other nodes, as appropriate. In accommodating the communication needs of the user applications, the limited bandwidth of wireless channels and their generally hostile transmission characteristics impose additional constraints on how much administrative and control information may be exchanged, and how often. Ensuring effective routing with improved performance is one of the great challenges for ad hoc networking.

In MANET, a wireless node can be the source of data transmission, destination or intermediate node. When a wireless node plays the role of intermediate node, it serves as a router that can receive and forward data packets to its neighbor closer to the

destination node. Due to the nature of an ad-hoc network, wireless nodes keep moving rather than staying still, the network topology changes from time to time. A node playing the role of a router may get out of the route between source and destination then the route is disconnected, and route discovery process has to be restarted. Thus, the main goal of routing protocol in MANET is to find a correct route efficiently

The lack of fixed base stations in ad hoc networks means that there is no dedicated agency for managing the channel resources for the network nodes. Instead, carefully designed distributed medium access techniques must be used for channel resources, and, hence, mechanisms must be available to recover efficiently from the inevitable packet collisions [15]. An effectively designed protocol for medium access control (MAC) and information's at the lower layers are essential to the quest for performance improvement in ad-hoc network routing protocols.

### 1.1.1 The ad hoc wireless network: operating principles

Mobile node A communicates with another, such as node B directly (single-hop) whenever a radio channel with adequate propagation characteristics is available between them

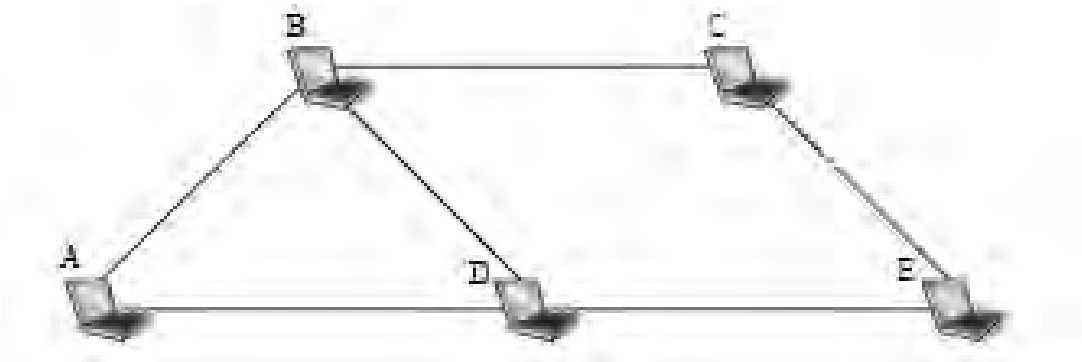



Fig. 1.2 Example of an ad-hoc network

.Otherwise, multi-hop communication is necessary where one or more intermediate nodes must act as a relay (router) between the communicating nodes. For example, there is no direct radio channel (shown by the lines) between A and C or A and E in Fig. 1.2. Nodes B and D must, therefore, serve as intermediate routers for communication between A and C, and A and E, respectively. Indeed, a distinguishing feature of ad hoc networks is that all nodes must be able to function as routers on demand. To prevent packets from traversing infinitely long paths, an obvious essential

requirement for choosing a path is that the path must be loop-free. A loop-free path between a pair of nodes is called a route.

An ad hoc network begins with at least two nodes broadcasting their presence (beaconing) with their respective address information. As discussed later, they may also include their location information, obtained, for example, by using a system such as the Global Positioning System (GPS), for more effective routing. If node A is able to establish direct connection with node B in Fig. 1.2, verified by exchanging suitable control messages between them, they both update their routing tables. When a third node, C, joins the network with its beacon signal, two scenarios are possible. The first one is where both A and B determine that single-hop communication with C is feasible. In the second scenario, only one of the nodes, say B, recognizes the beacon signal from C and establishes the availability of direct connection with C. The distinct topology updates, consisting of both address and route updates, are made in all three nodes afterwards. In the first case, all routes are direct. For the other, shown in Fig. 1.3, the route update first happens between B and C, then between B and A, and then again between B and C, confirming the mutual reachability between A and C via B.

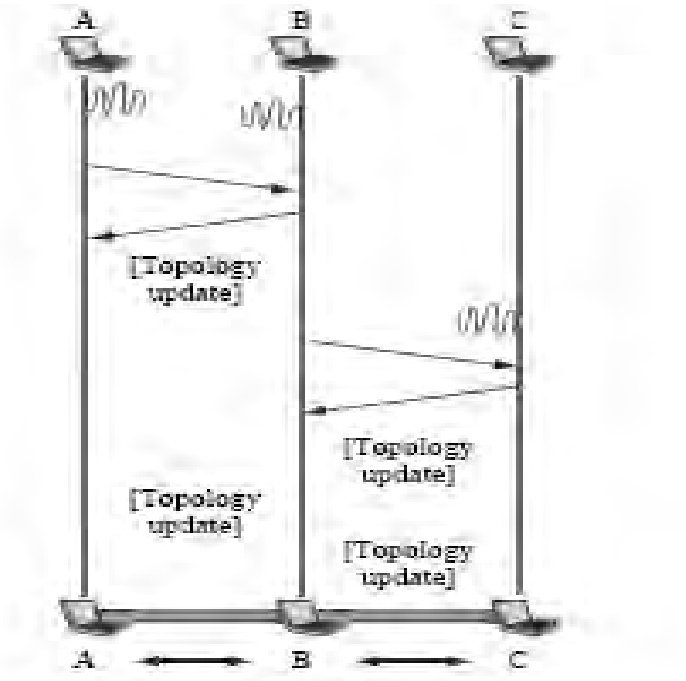



Fig. 1.3 bringing up ad-hoc network

The mobility of nodes may cause the reachability relations to change in time, requiring route updates. Assume that for some reason, the link between B and C is no longer available, as shown in Fig. 1.4. Nodes A and C can still reach each other, although this time only via nodes D and E. Equivalently, the original loop-free route A<->B<->C is now replaced by the new loop-free route A<->D<->E<->C. All five nodes in the network are required to update their routing tables appropriately to reflect this topology change, which will be first detected by nodes B and C, then communicated to A and E, and then to D.

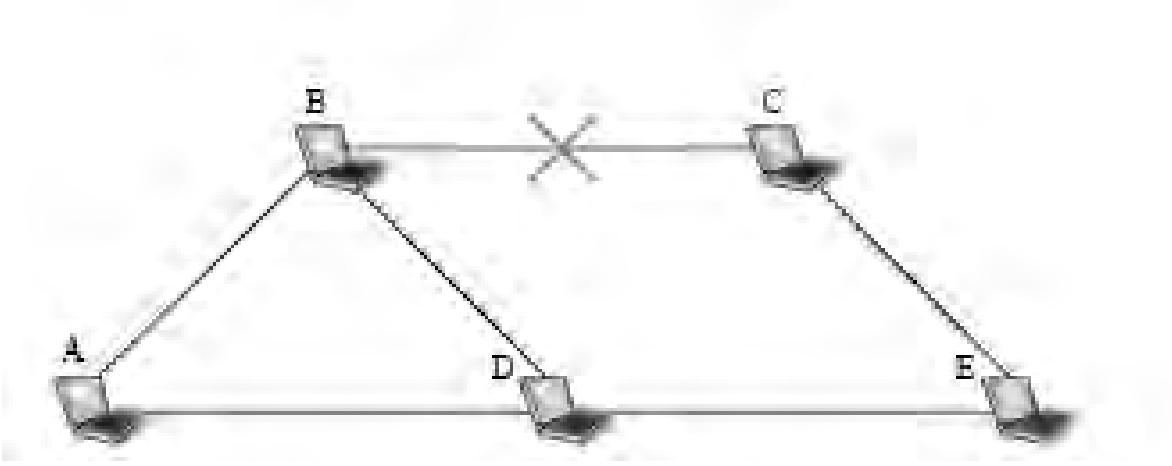



Fig. 1.4 Topology update due to link failure

The reachability relation among the nodes may also change for other reasons. For example, a node may wander too far out of range, its battery may be depleted, or it may suffer a software or hardware failure. As more nodes join the network or some of the existing nodes leave, the topology updates become more numerous, complex, and, usually, more frequent, thus diminishing the network resources available for exchanging user information.

### 1.2 Multi Cast Routing in MANETS

MANETs require fundamental changes to conventional routing protocols for both unicast and multicast communication owing to its unique features. With the rapid growth of group communication services, the multicast routing in MANET has attracted a lot of attention recently [2]. In multicast routing, a path is set up connecting all group members so that bandwidth is not wasted.

Based on the complexity of multi cast routing in ad hoc network, only few propositions are made to be used as multi cast routing protocols for ad hoc. Globally two main categories are recognized. Tree based protocols (MAODV, ABAM, ADMR) and mesh based (PUMA, ODMRP, SRMP) and common multi cast routing protocols for ad hoc network.

## 1.3 Security Vulnerabilities of Ad-Hoc Network

Mobile ad hoc networks are more prone to suffer from the malicious behaviors than the traditional wired networks [28]. Therefore, we need to pay more attention to the security issues in the mobile ad hoc networks.

A vulnerability is a weakness in security system. A particular system may be vulnerable to unauthorized data manipulation because the system does not verify a user's identity before allowing data access[10]. MANET is more vulnerable than wired network. Some of the vulnerabilities are as follows:

### 1.3.1 Lack of centralized management

MANET doesn't have a centralized monitoring server. The absence of management makes the detection of attacks difficult because it is not easy to monitor the traffic in a highly dynamic and large scale ad-hoc network. Lack of centralized management will impede trust management for nodes.

### 1.3.2 Resource availability

Resource availability is a major issue in MANET. Providing secure communication in such changing environment as well as protection against specific threats and attacks, leads to development of various security schemes and architectures. Collaborative ad-hoc environments also allow implementation of self-organized security mechanism.

### 1.3.3 Scalability

Due to mobility of nodes, the scale of ad-hoc network changes all the time. So scalability is a major issue concerning security. Security mechanism should be capable of handling a large network as well as small ones.

### 1.3.4 Cooperativeness

Routing algorithm for MANETs usually assume that nodes are cooperative and non-mailicious. As a result a malicious attacker can easily become an important routing agent and disrupt network operation by disobeying the protocol specifications.

### 1.3.5 Dynamic topology

Dynamic topology and changeable nodes membership may disturb the trust relationship among nodes. The trust may also be disturbed if some nodes are detected as compromised. This dynamic behavior could be better protected with distributed and adaptive security mechanisms.

### 1.3.6 Limited Resource

Resource constraints are a further vulnerability. There can be a variety of devices on MANETs, ranging from laptops to handheld devices such as PDAs and mobile phones. These will generally have different computing and storage capacities that can be the focus of new attacks. For example, mobile nodes generally run on battery power [23].

## 1.4 Statement of the problem

Most of the multicast routing protocols proposed for ad hoc networks assume a trusted, non-adversarial environment and do not take security issues into account in their design. Providing security in mobile ad-hoc networks has been a major issue over the recent year. Many scholarly articles are being suggested which help to standardize the multicast routing protocol to take into account the security issues. Of the recent concerns to standardize the protocols is to study how they perform under different type of security attacks. We showed, via simulation, that these attacks can have a significant impact on the performance of multi cast routing protocols

## 1.5 Research objectives

### 1.5.1 General Objective

The general objective of this thesis is to analyze the performance of MANETs multi cast routing protocol under the presence and absence of security attack for varied properties of multicast communication.

### 1.5.2 Specific Objectives

The specific objectives of this study are:

- Design network topology and create traffic to run simulation setup
- Analyse the performance of multi cast routing protocols on the existed traffic and topology
- Analyse the performance of multicast communications for different number sender and receivers nodes
- Create Black hole attack model and incorporate into ns2 simulator
- Create jelly fish attack model and incorporate into ns2 simulator
- Run simulations setup in the presence of attacks
- Compare results of attacked and non-attacked simulations setup

## 1.6 Scope

The scope of this study is limited to only analysis on the performance of multi cast routing protocols in the presence of attacks and without attacks. Design of attack models and incorporating the same in ns2 are in scope. We will analyse the simulation results to only PUMA protocols. Counter measures solution implementations are out of scope.

## 1.7 Methodology

The methodology to be followed to accomplish the study are :

### 1.7.1 Literature review

Review of literature related to ad hoc network and multi cast communication being the first to consider and study of different security attack in ad hoc network which can be outlined as follows:

- Study the basic principles of mobile ad-hoc networks.
- Review of the existing multi cast routing protocols, especially PUMA protocol
- Identifying the major attacks in routing and the major points of vulnerability.

### 1.7.2 Attack modelling

Attack modelling is related to how incorporate attack models in ns2, possible functions and architecture which help to mode attack will be studied which can be summarized as :

- Design of a set of security attack for the case of black hole and jellyfish
- Incorporate attacks in ns2

### 1.7.3 Simulation setup

The simulation setup considers creating traffic pattern and scenario files, choice parameters needed for simulation collecting trace

### 1.7.4 Analysis and interpretation of findings

This sections considers comparative analysis of results obtained from simulations of using graphs for

## 1.8 Thesis organization

The thesis is organized as follows. In Chapter 2, we discussed routing protocols in mobile ad hoc networks with detailed classification and operating principles by giving more emphasis on multicast routing protocols. Chapter3 briefly describes types of security attacks in ad hoc network. Chapter4 deals with related works in terms of security attacks on ad hoc routing protocols performance and with their achieved results. Chapter5 discusses our attack modeling and designing for simulation. Chapter6 illustrates the simulation results and discussion. Finally, conclusion and future work are discussed in chapter 7

# Chapter Two
# Routing Protocols in Mobile Ad Hoc Networks

Now we have a general overview of an ad hoc network and its operating principles, we move on to take a deeper look into the various approaches for routing in ad hoc networks proposed so far.

Several approaches toward routing in ad hoc networks have been proposed with the goal of achieving efficient routing. With the ever-changing topology, an intuitive approach of routing messages could be that the sender of the message specifies the exact path that the message should take from the sender to the receiver. But this assumes that the sender knows the entire topology of the network, which is not quite a realistic assumption. Another alternative could be to forward the message to a neighbor in the general direction of the destination, and the neighbor then makes a similar decision regarding how to route the message [32].

Based on the above-mentioned approaches, many routing algorithms have been put forward. As we have mentioned earlier, most of the routing algorithms proposed so far for ad hoc networks have been adapted from the routing techniques employed in wireline networks that have fixed network topology. But the routing algorithms devised for wire-line networks are based on the determination of the shortest path between the source and the destination. Hence these routing algorithms cannot be applied to ad hoc networks without modification as the network topology in ad hoc networks is always in a state of flux with changes in graph topology in one region of the network altering the shortest path between several pairs of nodes. Also, in ad hoc networks, this information takes time to propagate to other nodes in the system. If the topology information is not updated promptly, the nodes may continue to route messages based on the outdated information, which may lead to packet losses

## 2.1 Desirable properties of an efficient routing algorithm

Many routing protocols for ad hoc networks have been proposed so far, each one offering some advantage over the previous approach. But in general, there are some

common desirable properties that any routing protocol for an ad hoc network should possess as mentioned in [7]. These are:

- **Loop free:** Presence of loops in the path from the source to the destination result in inefficient routing. In the worst-case, the packets may keep traversing the loop indefinitely and never reach their destination.
- **Distributed control**: In a centralized routing scheme, one node stores all the topological information and makes all routing decisions; therefore, it is neither robust, nor scalable. The central router can be a single point of failure; also, the network in the vicinity of the central router may get congested with routing queries and responses.
- **Fast routing**: The quicker the routing decisions are made, the sooner the packets can be routed towards the destination, as the probability that the packets take the chosen route before it gets disrupted because of node mobility is quite high.
- **Localized reaction to topological changes**: Topological changes in one part of the network should lead to minimal changes in routing strategy in other distant parts of the network. This will keep the routing update overheads in check and make the algorithm scalable
- **Multiplicity of routes**: Even if node mobility results in disruption of some routes, other routes should be available for packet delivery.
- **Power efficient**: A routing protocol should be power efficient. That is the protocol should distribute the load; otherwise shut-off nodes may cause partitioned topologies that may result in inaccessible routes.
- **Secure:** A routing protocol should be secure. We need authentication for communicating nodes, non-repudiation and encryption for private networking to avoid routing deceptions.
- **QoS aware**: A routing protocol should also be aware of Quality of Service. It should know about the delay and throughput for a source destination pair, and must be able to verify its longevity so that a real-time application may rely on it.

## 2.2 Broad Classification of Routing Algorithms for Ad Hoc Networks

Having seen the composition of mobile ad hoc networks, a centralized routing scheme is completely ruled out, as this might lead to a single point of failure. This leads to the requirement of a distributed routing protocol where every node takes part in making routing decision and maintaining the topology by sharing information among them. These distributed routing protocols can be classified broadly, first by how they intend to determine the topology of the network, and second by when they make a decision to find a route to a destination. The first category of topology intended routing can further be classified into hierarchical and flat routing, and the second category can further be divided into proactive and reactive routing.

### 2.2.1 Hierarchical Routing

In hierarchical routing algorithms as described in [3] a set of nodes are divided into clusters. Each cluster has a node, which is designated as the cluster- head. So, every node is either a cluster head or one wireless hop away from the cluster head as shown in fig.2.1. A node that is not a cluster head, but adjacent to more than one cluster head, is referred to as a gateway. Packets between cluster-heads are routed through gateways. Finally, nodes that are neither cluster heads nor gateways are referred to as ordinary nodes. The subnet comprising the cluster heads and gateways is referred to as the backbone network. Here, each cluster head maintains information about other nodes in its cluster, and from time to time, this information is exchanged between cluster heads over the network. Thus, the cluster heads gather network topology information. A node that has a packet to send to another node can obtain routing information from its cluster head. It is not necessary for a packet to be routed through the backbone network, as data packets may be routed along other more efficient routes in the network.

This approach is also termed as topology aware routing as the nodes use the knowledge of the network topology to route messages. There are several ways of implementing this approach. The first possibility is that each node determines the optimal path to every node in the system and stores this information

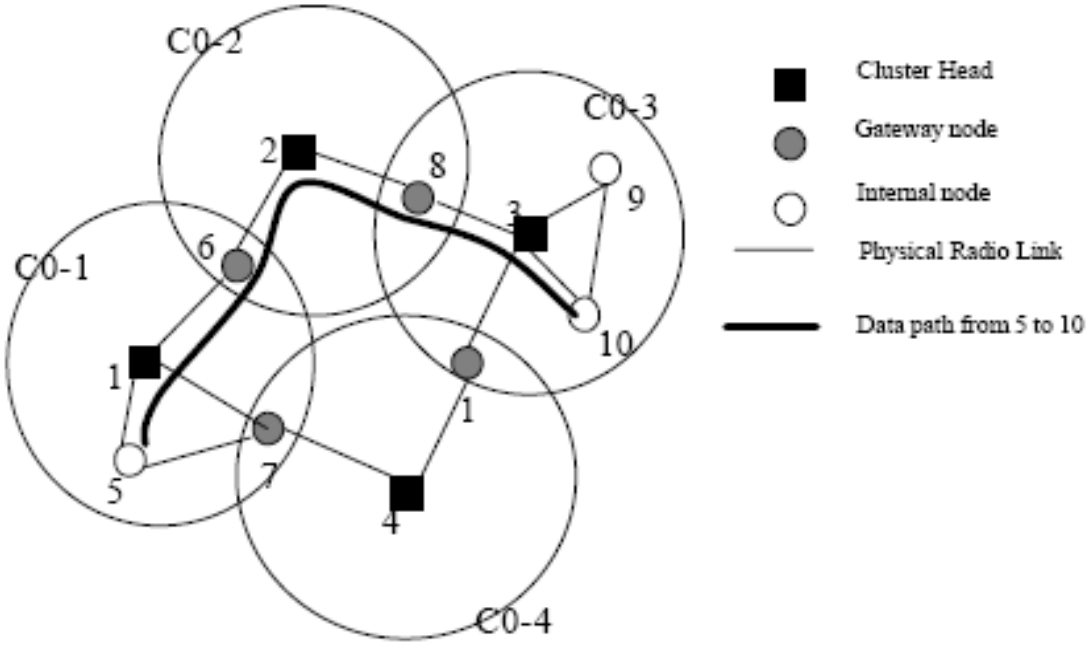



Fig. 2.1 Example of Hierarchical routing

Each time a stream of packets has to be sent from a source to a destination, a connection is established between two end-points and all the packets follow this path. However, with a changing topology, nodes will have to update their routing information and reestablish paths that were broken during communication. If the network topology does not change very often, it is likely that the path establishment costs are incurred once in the beginning and every subsequent packet is routed without additional overhead. The second possibility is connectionless routing, where a route is determined on the fly for every packet as it moves from one node to another. This method will require nodes to store less information about network topology. However, every packet incurs the routing overhead

## 2.2.2 Flat Routing

In case of flat routing algorithms all nodes act as routers and share the responsibility of forwarding packets destined to other nodes. Thus, there is no need to elect cluster heads and periodically reorganize the network. Most flat routing algorithms try to implement a distributed version of the shortest path algorithm or flooding where the general approach is that the sender sends a copy of the message to every neighbor.

The neighbors then propagate copies of the message to all their neighbors except the one from which they received the message. This process is repeated until the network is entirely flooded with the message. If the destination node is in the same partition as the source, the message is sure to reach the destination

#### 2.2.2.1 Proactive Routing

In proactive routing algorithms, each node maintains a routing table containing the next hop information for every other node in the network, and hence a route between the source node and the destination node is always available making the approach proactive. Examples of proactive protocols include Destination Sequenced Distance Vector (DSDV) and the Fisheye State Routing (FSR) [9]

### 2.2.2.2 Reactive Routing Algorithms

In reactive routing algorithms, a path discovery process determines the path to the destination only when the node has a packet to forward that is it reacts to a request to send data to a host. These types of routing algorithms are also referred to as on-demand routing protocols..

## 2.3 Multi Cast Routing in Ad Hoc Network

Multicasting is intended for group communication that supports the dissemination of information from a sender to all the receivers in a group.

When it became clear that group-oriented communication is one of the key application classes in MANET environments, a number of MANET multicast routing protocols have been proposed [16]. These protocols can be classified according to two different criteria. The first criterion has to do with maintaining routing state and classifies routing mechanisms into two types: proactive and reactive. Proactive protocols maintain routing state, while the reactive - reduce the impact of frequent topology changes by acquiring routes on demand.

The second criterion [16] classifies protocols according to the global data structure used to forward multicast packets. Existing protocols are either tree- or mesh-based.

As in fixed (non-mobile) multicast routing, tree-based protocols build a tree over which multicast data is forwarded. Although bandwidth-efficient, tree-based protocols do not always offer sufficient robustness. Certain key features of MANETs, such as fast deployment, make them well-suited for critical environments (e.g., battle field or disaster recovery) where robustness and reliability are essential. Thus, one of the main challenges for multicast routing in MANETs is the need to achieve robustness in the presence of universal mobility and frequent node outages. For this purpose, mesh-based protocols build a mesh for forwarding multicast data and thus address robustness and reliability requirements with path redundancy inherent to meshes.

### 2.3.1 Mesh- and Tree-Based Multicast Overview

In this section we review the operation of mesh- and tree-based multicast routing using ODMRP and MAODV as examples of mesh- and tree-based protocols, respectively.

#### 2.3.1.1 Multicast Ad hoc On-demand Distance Vector- MAODV

Royer and Perkins proposed MAODV in 1999 [8]. Here protocol discovers multicast routes on demand using a broadcast route discovery mechanism. When a node wishes to join a multicast group or it has data to send to the group but does not has a route to that group, it originates a route request (RREQ) message. Only the members of the multicast group respond to the join RREQ. If an intermediate node receives a join RREQ for a multicast group of which it is not a member or it receives a route RREQ and it does not have a route to that group, it rebroadcast the RREQ to its neighbors. But if the RREQ is not a join request any node of the multicast group may respond. Figure 2.1 depicts the propagation of RREQ

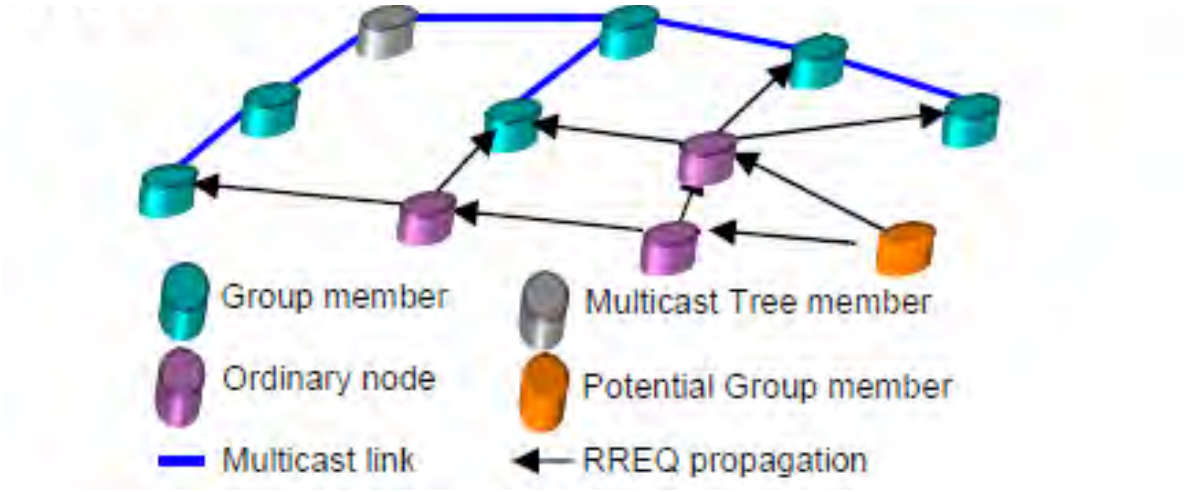



Fig. 2.1: RREQ Propagation

Every node sets up pointers to determine the reverse route in its routing table upon receiving a RREQ. This entry may later be used to relay a response back to the route requester. This entry is not activated until or unless it gets multicast activation message from the requester. The responding node unicasts the route response RREP as shown in fig. 2.2 back to the route requester after the completion of necessary updates on it routing table.

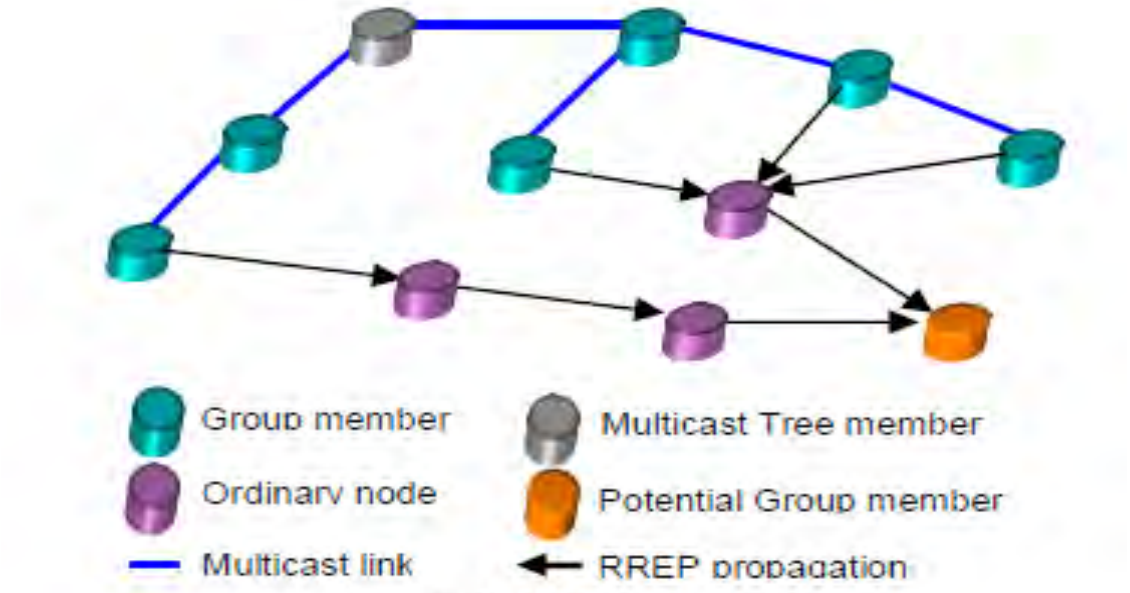



Fig. 2.2: RREP Propagation

A node may receive multiple route reply for a route request. Usually node selects a route with the greatest sequence number and the shortest hop distance to the member of the multicast group and discards other routes. After that, node enables the selected next hop in its routing table and unicasts an activation message to that node. Upon receiving this message it activates the entry for that node in its multicast routing table. It does not forward the message further if it is a member of the multicast group otherwise it does. On the other hand, if it is not a member of the multicast group it may have multiple options to forward this activation message due to multiple route responses. It chooses best next hop and unicasts this activation message to the next hop. This process continues until activation message reached to the source of the route responder. Figure 2.3 represents the final multicast tree

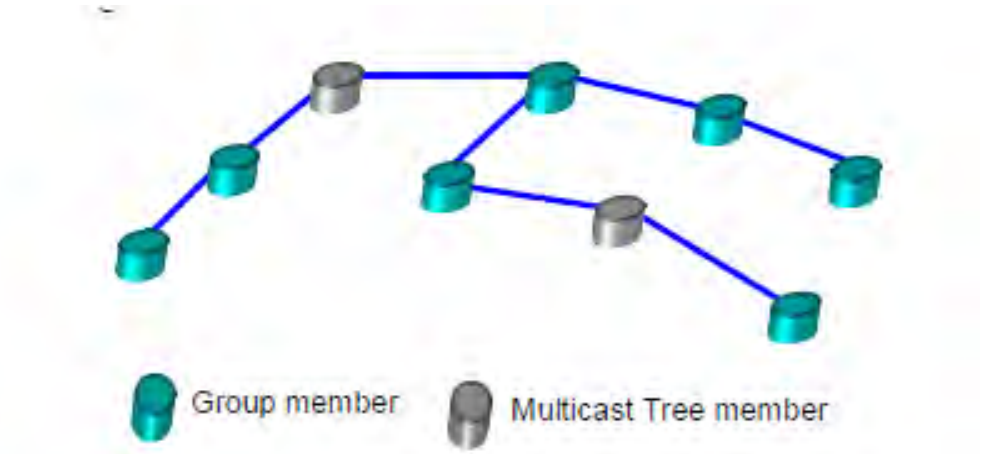



Fig. 2.3: Multi cast tree formed by MAODV protocol

For maintenance purpose MAODV uses group leader. The member who joins the multicast group first becomes the leader of that group. It periodically broadcasts hello message containing group sequence number to the multicast group. Using this hello message nodes refresh their routing tables.

MAODV routing protocol requires to actively follow and respond to the changes in the multicast tree as it maintains hard state in its routing table. In order to terminate from the multicast group MAODV requires pruning. It allows a node to quit from the group if it is a leaf node in the tree otherwise it must remain in the tree as a non group multicast member. Links are checked to detect link failures. When link failure is detected, downstream node is responsible for repairing the link

### 2.3.1.2 On-Demand Multicast Routing Protocol – ODMRP

In 2000, Bae et al. proposed a mesh based, rather than a conventional tree based, multicasting routing protocol, named On-Demand Multicast Routing Protocol (ODMRP) [24]. To carry multicast data via scoped flooding it uses forwarding group concept

The source, in ODMRP, establishes and maintains group membership. If source wishes to send packet to a multicast group but has no route to that group, it simply broadcasts JOIN_DATA control packet to the entire network. When an intermediate node receives the JOIN_DATA packet it stores source address and sequence number in its cache to detect duplicate. It performs necessary routing table updates for reverse path back to the source. Non duplicate message is rebroadcasted if TTL value is greater than zero as shown if figure 2.4

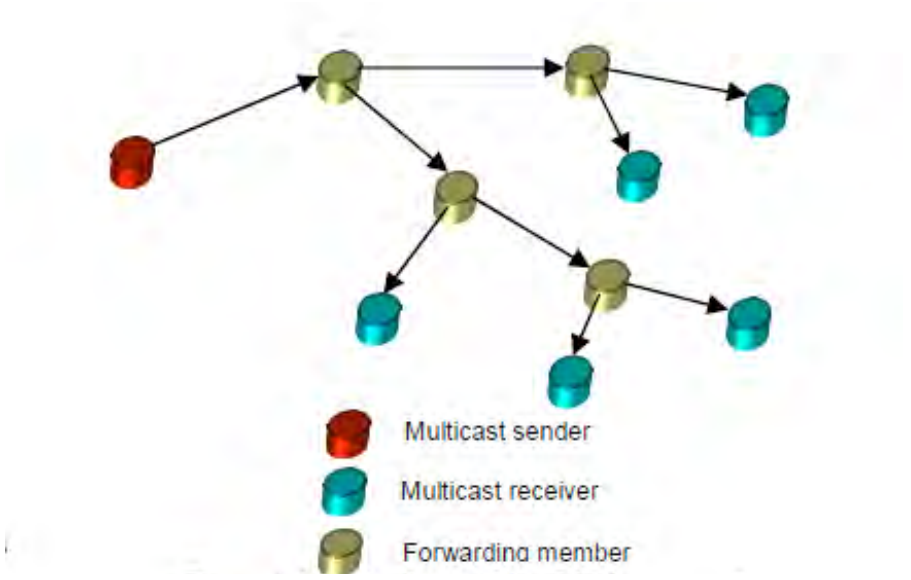



Fig. 2.4: Non duplicate JOIN_DATA propagation

A multicast receiver constructs a JOIN_TABLE upon getting JOIN_DATA packet and broadcasts it to its neighbours. When a node receives a JOIN_TABLE, it resolves whether it is on the way to the source by consulting earlier cached data. If it realizes it is the part of forwarding group it sets forwarding group flag(FG_FLAG).

Considering the matched entry this node builds new join table and broadcasts it. In this way JOIN_TABLE is propagated with the help of forwarding group members and ultimately it reaches to the multicast source as shown in fig 2.5. A multicast table is built on each node to carry multicast data as shown in fig 2.6. This process either constructs or revises the routes from sources to receivers and forms a mesh.

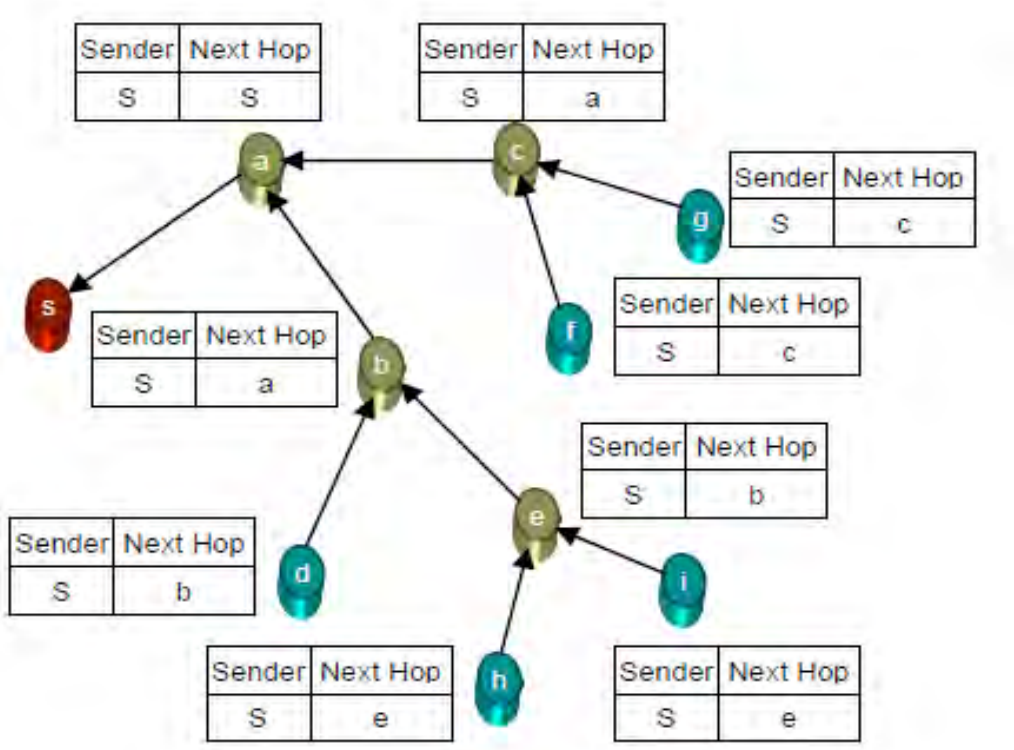



Fig. 2.5: JOIN_TABLE propagation

Group maintenance in ODMRP is quite simple as it uses soft state approach. No explicit control packets are required to join or leave the group. If a multicast source wishes to leave the group, it simply stops sending JOIN_DATA packets. On the other hand if a multicast receiver wants to escape from the group it just stops responding to the join reply.

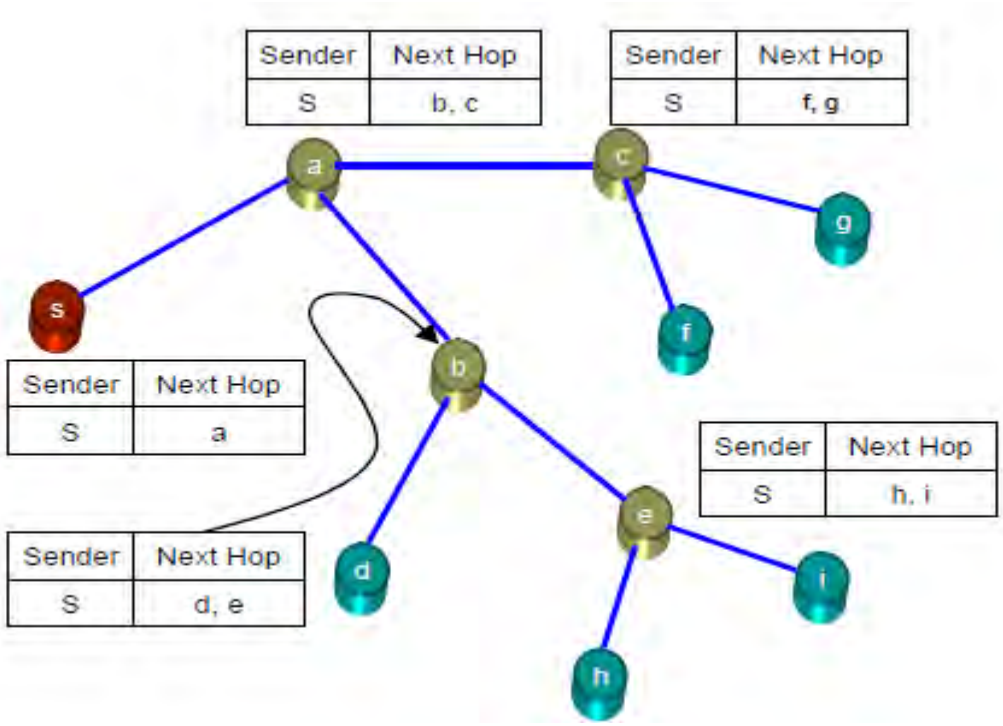



Fig.2.6: Multi cast table formation of ODMRP

The robustness of mesh configuration is in depicted in the fig. 2.7 Let, three multicast sources, S1, S2, and S3, are sending multicast packets to the receivers, R1, R2 and R3. In doing the transportation there are three forwarding group members namely, A, B and C. In a tree configuration if a link fails between any path between sender to receiver, data forwarding is stopped instantly until tree is reconfigured. But in this mesh configuration there may be some redundant paths between senders to receivers. And hence ensures some sorts of robustness by exploiting redundant paths.

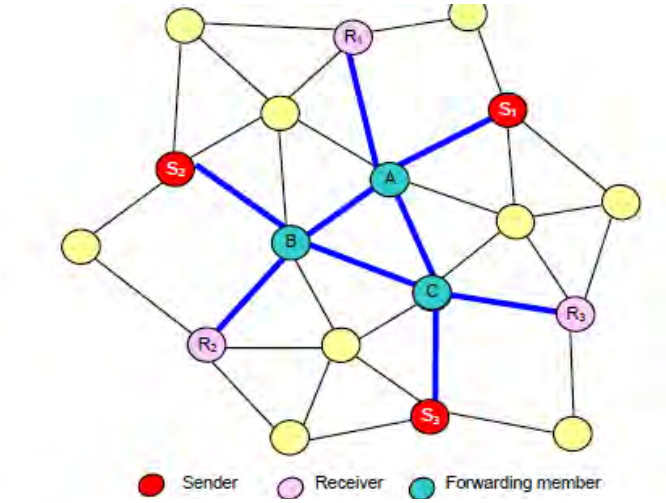



Fig. 2.7: Mesh formation in ODMRP

### 2.3.1.3 Position Based Multi Cast Routing (PBM)

Position Based Multicast (PBM) routing protocol uses geographical position of the nodes to make forwarding decision [13]. One of the key features of PBM approach is that it neither requires to maintain a data distribution structure such as tree or mesh nor resorts the flooding. Actually it is a generalization of existing position based unicast routing protocol such as Greedy Perimeter Stateless Routing (GPSR) or Face.

The protocol assumes the position of the destinations are known to the sender by means of location service, the position of its own by the use of GPS and the position of its direct neighbour through periodic beacons. Two key problems have to be solved to adapt PBM unicast routing to multicast routing. On of them is to decide when and where at a particular node and a particular multicast packet has to split into multiple copies to reach all the destinations. Another is the recovery strategy used to escape from a local optimum to reach multiple destinations

Two distinct cases can occur when forwarding node selects the next hop nodes. In the first case, for each destination there is at least one neighbour exists that is closer to the destination than the current forwarding node itself. Greedy multicasting is used in this case. Otherwise perimeter multicasting is deployed.

### 2.3.1.4 Overlay Multicast – PAST-DM

Progressively Adapted Sub-Tree in Dynamic Mesh (PAST-DM) is an overlay multicast routing protocol that builds a virtual mesh spanning all the members of a multicast group [30]. In order to carry packets it uses unicast routing protocol. But this algorithm gradually adapts to the changes of the physical topology in a distributed manner. The advantages of this approach are the robustness and the low overhead.

The advantages of overlay multicast are at the cost of low efficiency of packet delivery and long delay. When constructing the virtual topology, it is very hard to prevent different unicast tunnels from sharing physical links, which results in redundant traffic on the physical links. Fig 2.8 is an example of such a scenario.

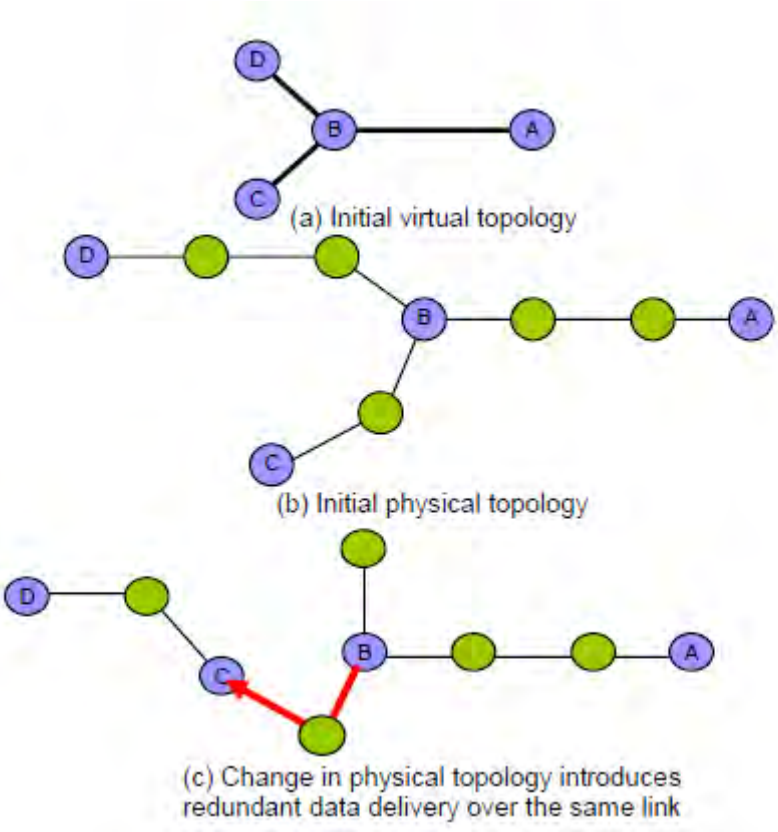



Fig. 2.8: Side effect of overlay multicast

A multicast session begins with the formation of a virtual mesh spanning all group members. Each member node makes the use of expanded ring search (ERS) technique to discover neighbours. When a node ***I*** receives a group request message from node ***J***, along with the hop distance to node ***J***, node ***I*** records node ***J*** as its neighbour in the virtual mesh and then sends back a group response message to ***J***, so that node ***J*** will record the same. This virtual topology has a maximum degree for each node. The node stops the neighbour discovery phase when the number of virtual neighbours of a node reaches the upper limit. If a node fails to discover any neighbour using the expanded ring search technique, in that case it can use flooding to locate neighbours.

By exploiting unicast routing table each node keeps the track of other nodes in its locality. Each node records its virtual neighbours as a virtual link. Topology map is represented as a link state table. The entries are the link state information of all group nodes obtained from virtual neighbours. Every node periodically exchanges this link state table with its virtual neighbour nodes only. Through this link state table node has a local view of the entire virtual topology.

#### 2.3.1.5 Source Routing-based Multicast Protocol

SRMP is a mesh-based multicast routing protocol. A mesh structure (an arbitrary sub-network) is established on demand to connect group members [25], providing richer connectivity among multicast members. By building a mesh, packets can be efficiently delivered to multicast receivers in the case of node movements and topology changes. In addition, drawbacks of multicast trees can be avoided (ex,intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree).

SRMP is based on a new source routing approach, in which the source route accumulates in the reply packet. The source routing concept is used by DSR unicast protocol , allowing each data packet to carry in its header the list of nodes' addresses through which this packet must be transmitted.

During mesh establishment, SRMP uses the Forwarding Group (FG) nodes concept . The FG is a set of nodes responsible for forwarding multicast data between any member pairs. This scheme can be viewed as a "limited scope" flooding within a

properly selected forwarding set. The key innovation of SRMP is to handle effective criteria in selecting FG nodes in order to achieve a compromise between the number of the selected nodes, the availability and the stability of the selected paths. Four metrics are considered to establish the mesh structure: association stability, link signal strength, link availability, and higher battery life.

#### 2.3.1.6 PUMA Protocol

PUMA [29] is a reactive routing protocol which discovers route only when it is required. Its multicast connectivity is established and maintained by means of receiver initialization approach in which the receivers joins into the multicast group by using address of core node without the need for network-wide flooding of control or data packets from all the sources of the group. Each group has exactly has one special node which is called core node in the group. PUMA's uses the shared mesh based multicast topology for constructing routes to the members of the multicast group without depending upon any unicast routing protocol. Multicast group maintenance of PUMA is achieved by using the soft state approach where in which the multicast group membership and its associated routes are refreshed periodically by flooding its Multicast Announcement (MA) packet

##### 2.3.1.6.1 Control Packet

PUMA uses a single control packet called Multicast Announcement (MA) to create and maintain its multicast topology in MANET whose format is shown in table 2.1.

**Mesh membership code** – this field is set to 1 when a node wants to join into the group; else it is unset.

**Distance to core** – hop count from the current node to core node.

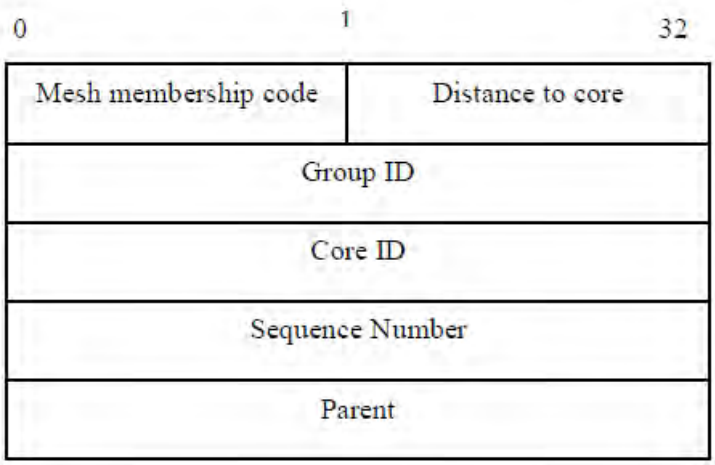
**Group ID** – address of the group.

**Core ID** – address of the core node.

**Sequence number** – sequence number of the group.

**Parent ID** – address of the neighbor to reach the core.

Table 2.1 : Multicast Announcement




This multicast announcement packet is used to elect the core, find out the sources outside a multicast group to unicast data packets towards the group, join and leave the mesh of a group, and maintain the mesh of the group.

**2.3.1.6.2 Core Election**

PUMA chooses a core for each multicast group in the network. Each connected component has only one core. If one receiver joins the group before other receivers, then it becomes the core of the group. If several receivers join the group at the same time, then the one with highest ID becomes the core of the group.

When a receiver needs to join a multicast group, it first determines whether it has received a multicast announcement for that group. If the node has received, then it takes on the core specified in the announcement it has received, and it transmits the multicast announcements that specifies the same core for the group. Otherwise it assumes itself as the core of the group and starts transmitting multicast announcement periodically to its neighbors stating itself as the core of the group and a hop count of 0 distance to itself. Nodes propagate multicast announcements based on the best multicast announcement they receive from their neighbors.

A node that believes itself to be the core of a group, it transmits multicast announcements periodically for that group. As the multicast announcement pass

through the network, it establishes a connectivity list at every node in the network[4]. Connectivity list is used to form a mesh structure and to route data packets from receivers to the core.

A node keep tracks of the data from all the multicast announcements it receives from its neighbors in the connectivity list. Fresher multicast announcements from a neighbor overwrite entries with lower sequence numbers for the same group. Hence all the nodes in a group store the recent information about a neighbor for each core in the group

Each entry in the connectivity list also stores the time when it was received, and the neighbor from which it was received. The node then generates its own multicast announcement based on the best entry in the connectivity list.

While electing core, on receiving a multicast announcement with higher core ID, all entries in the connectivity list with a lower core ID are erased. Hence all suitable entries in the connectivity list at any point of time have the same core ID and sequence number. Among these suitable entries, the entries with a shortest distance to core be qualified as best entries, and the neighbors corresponding to these entries are called parents.

After selecting the best multicast announcement, the node generates the fields of its own multicast announcement in the following way:

- **Core ID:** The core ID in the best multicast announcement.
- **Group ID:** The group ID in the best multicast announcement.
- **Sequence number:** The sequence number in the best multicast announcement
- **Distance to core:** One plus the distance to core in the best multicast announcement.
- **Parent:** The neighbor from which it received the best multicast announcement.
- **Mesh member:** A node sets its membership code field based on whether it is a mesh member or non-member.

  After generating its own multicast announcement, it will broadcast to its entire neighbor as shown in fig.2.9

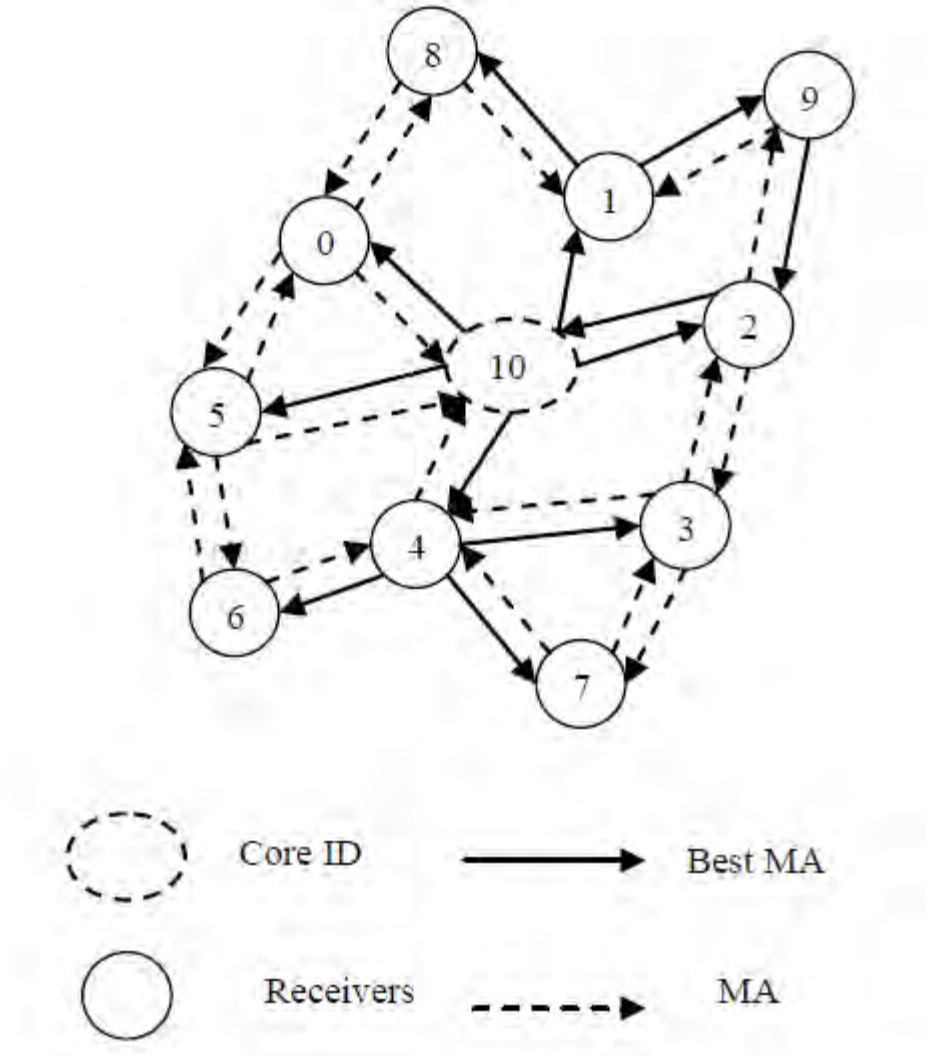



Fig: 2.9 Dissemination of Multicast Announcement

The solid arrows in fig 2.10 indicate the neighbor from which a node receives its best multicast announcement. Node 5 has three entries in its connectivity list as shown below in table 2.2 for neighbors 10, 0, and 6. However it chooses the entry it receives from 10 as the best entry, because it has the shortest distance to core. Node 5 uses this entry to generate its own multicast announcement, which specifies Core ID = 10, Group ID = 224.0.0.1, Sequence Number = 79, Distance to Core = 2 as table 2.2 indicated and Parent = 10. When a node wants to send data packets to the group it forwards it to the node from which it received its best multicast announcement. If that link is broken then it tries its next best and so on. Hence each node in the network has

one or more routes to the core. The multicast announcement sent by the core has distance to core set to zero and parent field set to INVALID-ADDRESS, because the core has no parents.

Multicast announcements are generated by the core for every three seconds. After receiving a multicast announcement with a highest sequence number, nodes wait for a short period (e.g. 100 ms) to collect multicast announcements from multiple neighbors before generating their own multicast announcement.

Table 2.2 Sample Connectivity List

| Neighbor | Multicast Announcement | | Time (ms) |
|---|---|---|---|
| | Distance to core | Parent | |
| 10 | 1 | 10 | **12152** |
| 0 | 2 | 10 | **12183** |
| 6 | 3 | 4 | **12270** |

## 2.4 Quantitative Performance Metrics of MANET Routing Protocols

MANET routing protocols have the following quantitative performance metrics [19]

### 2.4.1 Packet Delivery Fraction(PDF)

It is the total number of packets received at their intended destination divided by the total number of generated packets

$$PDF = \frac{\text{Packet received}}{\text{Packet sent}}$$

### 2.4.2 Normalized Routing Load (NRL)

Normalized routing load is the ratio of the number of control packets propagated by every node in the network and the number of data packets received by the destination nodes.

### 2.4.3 End –to-End Delay (E_E Delay)

It is the average delay time of all successfully delivered packets

### 2.4.4 Jitter

It is an average variation from the end-to-end delay

### 2.4.5 GoodPut

It is the total number of data packets that reached correctly to destination. It can be measured in terms of bytes. Total number of packets will multiplied by maximum packet size to get the total number data bits reached to destination.

# Chapter Three
# Security Attacks in Ad-Hoc Network

Security in mobile Ad-Hoc network is a big challenge as there is no centralized authority which can supervise the individual nodes operating in the network. The attacks can come from both inside the network and from the outside MANET has no clear line of defense, so, it is accessible to both legitimate network users and malicious attackers. In the presence of malicious nodes, one of the main challenges in MANET is to design the robust security solution that can protect MANET from various routing attacks [20]. However, these solution are not suitable for MANET resource constraints, i.e., limited bandwidth and battery power, because they introduce heavy traffic load to exchange and verifying keys. MANET can operate in isolation or in coordination with a wired infrastructure, often through a gateway node participating in both networks for traffic relay. This flexibility, along with their self-organizing capabilities, are some of MANET's biggest strengths, as well as their biggest security weaknesses. In this chapter different routing attacks, such as active (flooding, black hole, spoofing, and wormhole) and passive (eavesdropping, traffic monitoring, and traffic analysis) will be described.

## 3.1 Security Goals

Security involves a set of investments that are adequately funded. In MANET, all networking functions such as routing and packet forwarding, are performed by nodes themselves in a self-organizing manner. For these reasons, securing a mobile ad hoc network is very challenging. The goals to evaluate if mobile ad hoc network is secure or not are as follows[20]:

### 3.1.1 Availability

Availability means the assets are accessible to authorized parties at appropriate times. Availability applies both to data and to services. It ensures the survivability of network service despite denial of service attack.

### 3.1.2 Confidentiality

Confidentiality ensures that resources or data are accessed only by authorized parties. That is only those who should have access to something will actually get that access.

To maintain confidentiality of some confidential information, we need to keep them secret from all entities that do not have privilege to access them. Confidentiality is sometimes called secrecy or privacy.[5]

### 3.1. 3 Integrity

Integrity means that assets can be modified only by authorized parties or only in authorized way. Modification includes writing, changing status, deleting and creating. Integrity assures that a message being transferred is never corrupted.

### 3.1.4 Authentication

Authentication enables a node to ensure the identity of peer node it is communicating with. Authentication is essentially assurance that participants in communication are authenticated and not impersonators. Authenticity is ensured because only the legitimate sender can produce a message that will decrypt properly with the shared key.

### 3.1.5 Nonrepudiation

Nonrepudiation ensures that sender and receiver of a message cannot deny that they have ever sent or received such a message .This is helpful when we need to discriminate if a node with some undesired function is compromised or not.

### 3.1.6 Anonymity

Anonymity means all information that can be used to identify owner or current user of node should default be kept private and not be distributed by node itself or the system software

## 3.2 Security Attacks

Securing wireless ad hoc networks is a highly challenging issue. Understanding possible form of attacks is always the first step towards developing good security solutions. Security of communication in MANET is important for secure transmission of information. Absence of any central co-ordination mechanism and shared wireless medium makes MANET more vulnerable than wired network. These attacks can be classified into two types:

### 3.2.1 Passive Attacks

Passive attacks does not disrupt proper operation of network. Attackers snoop data exchanged in network without altering it. Requirement of confidentiality can be

violated if an attacker is also able to interpret data gathered through snooping .Detection of these attack is difficult since the operation of network itself does not get affected.

### 3.2.2 Active Attacks

Active attacks are the attacks that are performed by the malicious nodes that bear some energy cost in order to perform the attacks. Active attacks involve some modification of data stream or creation of false stream. Active attacks can be internal or external

## 3.3 Active Attacks

Different types of active attacks are available in MANET. Here we will discuss only the most common types of active attacks

### 3.3.1 Black Hole Attack

In this attack, an attacker advertises a zero metric for all destinations causing all nodes around it to route packets towards it. A malicious node sends fake routing information, claiming that it has an optimum route and causes other good nodes to route data packets through the malicious one. A malicious node drops all packet that it receive instead of normally forwarding those packets. An attacker listen the requests in a flooding based protocol

### 3.3.2 Worm Hole Attack

In a wormhole attack, an attacker receives packets at one point in the network, "tunnels" them to another point in the network ,and then replays them into the network from that point. Routing can be disrupted when routing control message are tunnelled.This tunnel between two colluding attacks is known as a wormhole .In reactive routing protocol, this attack could prevent discovery of any routes and may create a wormhole even for packet not address to itself because of broadcasting. Wormholes are hard to detect because the path that is used to pass on information is usually not part of the actual network. Wormholes are dangerous because they can do damage without even knowing the network

### 3.3.3 Jelly Fish Attack

Jellyfish attack is somewhat different from Black-Hole & Worm-Hole attack. Instead of blindly dropping the data packets, it delays them before finally delivering them. It

may even scramble the order of packets in which they are received and sends it in random order. This disrupts the normal flow control mechanism used by nodes for reliable transmission. Jellyfish attack can result in significant end to end delay and thereby degrading QoS[11] .

### 3.3.4 Rushing Attack

In reactive routing protocol, each node before transmitting its data, first establishes a valid route to destination. Sender node broadcasts a RREQ (route request) message in neighborhood and valid routes replies with RREP (route reply) with proper route information. Some of the protocols use duplicate suppression mechanism to limit the route request and reply chatter in the network. Rushing attack exploits this duplicate suppression mechanism. Rushing attacker quickly forwards with a malicious RREP on behalf of some other node skipping any proper processing. Due to duplicate suppression, actual valid RREP message from valid node will be discarded and consequently the attacking node becomes part of the route.

### 3.3.5 Neighbor Attack

Make two or more hop away nodes think that they are one-hop away (neighbors). Neighbor attackers don't update their IP addresses in the last-hop field of the Join Query packet

### 3.3.6 Gray-hole attack

This attack is also known as routing misbehavior attack which leads to dropping of messages. Gray hole attack has two phases. In the first phase the node advertise itself as having a valid route to destination while in second phase, nodes drops intercepted packets with a certain probability.

## 3.4 Passive Attacks

### 3.4.1 Traffic Monitoring

It can be developed to identify the communication parties and functionality which could provide information to launch further attacks .It is not specific to MANET, other wireless network such as cellular, satellite and WLAN also suffer from these potential vulnerabilities.

### 3.4.2 Syn flooding

This attack is denial of service attack. An attacker may repeatedly make new connection request until the resources required by each connection are exhausted or reach a maximum limit. It produces severe resource constraints for legitimate nodes.

This paper attempts to analyze the effect of two of the active attacks namely black hole and jelly fish on performances of MANET's multi cast routing protocols

# Chapter Four

# Related Works

## Introduction

In Mobile Ad-Hoc Networks (MANETs), security is one of the most important concerns because a MANETs system is much more vulnerable to attacks than a wired or infrastructure-based wireless network.

Considering the vulnerability nature of ad-hoc for security attacks, lots of research works are going on to study the impact of security attack on performances of MANETs routing protocols each addressing its own particular issue of interest.

Since this paper aims to study the impact of security attack on multi cast routing protocols, the related works that we are going to review in this chapter only considers those that address multi cast routing protocols

## 4.1 Study of Different Attacks on Multicast Mobile Ad Hoc Network

Study of different types of attacks on multi cast mobile ad hoc network was carried out[17], this paper, presented simulation-based study of the effects of different types of attacks on tree based multicast in MANETS. The most common types of attacks namely Gray whole attack and Wormhole attack were considered.

The research conducted simulation using NS-2 simulator, a scalable simulation environment for wireless network systems. The simulated network consists of 100 nodes placed randomly with in 1500x300m area. Each node has a transmission range of 250m and moves at a speed of 10m/s. The total sending rate of all the senders of the multicast group, i.e., the traffic load is 1Mbps. They use a high traffic load value, highlight the effects of the attacks on the packet loss rate, as opposed to packet loss due to congestion and collisions from a high traffic load.

This set of simulations compares the performance of multicast operation over AODV protocol during its normal operation, introducing both gray hole attack and worm hole attack, into the network using MAODV protocol, performance of measures against

these attacks by varying the number of multicast receivers as 10, 20, 40 and 60 respectively. The number of multicast sender is one. Different performance metrics like Packet Delivery ratio Packet Latency Packet Consumed energy

From this paper it can be seen that the simulation considers node moving with constant speed which is not the common scenario most of the time. Since Grey hole and worm whole attack might have similar impacts, different attack types might have been useful for the study. Once of the common features of multicast communication is the ability to vary number of senders and compare performance of multi cast routing protocol, this paper considered only one sender and use MAODV routing protocol

**4.2 Neighbor Attack and Detection Mechanisms in Mobile Ad hoc Network**

In this paper [21], researchers presented simulation based study of the impact of neighbor attack on mesh-based Mobile Ad-Hoc Network (MANET). And also the study considered the number of attackers and position affects the performance metrics such as packet delivery ratio and throughput.

The research conducted experiments using Glomosim simiulator version 2.03. The simulated network consists of 50 mobile nodes placed randomly within a 1000 m x 1000 m area. Each node has a transmission range of 250 m and moves at a speed of 1 m/s. The total sending rate of all the senders of the multicast group, i.e. the traffic load, is 1 packet/s. They used a low traffic load value to highlight the effects of the attacks on packet loss rate, as opposed to packet loss due to congestion and collisions resulting from a high traffic load.

The mobility model chosen for a mobile node was the random way-point model. A mobile node begins by staying in one location for a pause time of 30 seconds. Once this time expires, the mobile node chooses a random destination in the simulation area and then travels toward the newly chosen destination. Upon arrival, the mobile pauses for 30 seconds before starting the process again. The attackers were positioned around the center of the multicast mesh in all experiments. In these experiments, they simulated four scenarios. In the first three scenarios, the attacker group was placed near

the senders, near the receivers, and around the mesh center, respectively. In the fourth scenario, the attackers were uniformly distributed over the whole network. The duration of each experiment was 300 seconds in simulated time.

The simulation was carried out for ODMRP routing protocol and for performance metrics of packet delivery fraction, which showed lowering of values due the presence of attack

Since the simulation setup uses node movement with 1m/s which is of pedestrian scenario, high speed scenario is not considered in the study

### 4.3 Performance Evaluation of Mesh based Multicast Reactive Routing Protocol under Black Hole Attack

This paper discusses the impact of black hole attack on ODMRP under various scenarios [7].

The simulation settings were as follows. The network consists of 50 nodes placed randomly within an area of 1000m x 1000 m. Each node moves randomly and has a transmission range of 250m. 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 50 m/s. The simulations were carried out with 2, 5, 7 and 9 attackers for different number of receivers. The malicious nodes were selected randomly.

The performance is evaluated using metrics such as packet delivery ratio and end to end delay for various numbers of senders and receivers via simulation. Simulations are carried out using network simulator ns-2. The results show that black hole attack has impact dropping packets reaching to a node labeled as malicious

This research considers only Black hole attack and the protocol used is ODMRP routing protocol

### 4.4 Impact of Rushing Attack on Multicast in Mobile Ad Hoc Network

This paper is based on Rushing attack[27]. In Rushing attack, the attacker exploits the duplicate suppression mechanism by quickly forwarding route discovery packets in order to gain access to the forwarding group and this will affect the Average Attack Success Rate. In this paper, the researchers attempt to measure the impact of Rushing attack and their node positions which affect the performance metrics of Average

Attack Success Rate with respect to three scenarios: near sender, near receiver and anywhere within the network. The performance of the Attack Success Rate with respect to above three scenarios is also compared

The experiment runs simulations under Linux, using the network simulator NS2 version ns-allinone-2.26. The simulation environment is composed of: area: 500*500 meters, number of nodes 50 – 100, simulation duration: 1000s, physical/Mac layer: IEEE 802.11 at 2Mbps, 250 meters transmission range, mobility model: random waypoint model with no pause time, and mode, movement speed 0m/s, 1m/s and 10m/s, using routing protocol MAODV under NS2.26

The research considered only packet delivery ratio as a metric low speed node movement, where decrease in the value of the metric is observed for near sender attacker position.

**4.5 Multi cast security attacks and its countermeasures for PUMA protocol**

This paper considered the vulnerabilities of PUMA which is a representative of mesh based routing protocol. In this the researchers carried out simulation based study of black hole and wormhole attack [28].

The study considered black hole scenario in the protocol. The performance of packet delivery ratio is evaluated by computer simulation using ns 2.34. The research showed the difference between the normal PUMA operation and PUMA under attacks, in which some nodes are made to play the role of attackers at different simulation time.

The nodes in the computer simulation move according to the Random Waypoint Algorithm. The scenario was defined with a set of parameters as follows. Number of nodes: 50, Simulation tool: Ns 2.34, Data rate: 11Mb, Packet Size: 1000, Traffic Type: CBR, Simulation Duration: 250 sec

Packet delivery ratio was seen to decrease with increase in number of attacker nodes for constant speed movement of nodes. The paper did not consider the impact of increasing or decreasing number of multi cast senders and receivers , the effect of other performance metrics is not considered.

### 4.6 Securing MAODV: Attacks and Countermeasures

The research assess the vulnerability of MAODV to attacks launched by both insider and outsider nodes [14]. In particular, the research identify attacks on multicast tree formation and maintenance that have no counterpart in unicast routing protocols

Simulations were written using the ns-2 simulator (version 2.26) with CMU Monarch extensions. In simulations, researchers used the two-ray ground reflection model to model radio propagation, the IEEE 802.11 Distributed Coordination Function (DCF) as the MAC layer protocol, and the random waypoint model as the node mobility model.

The scenario simulated consists of a network with 50 nodes and a single multicast group with 10 members. All group members join the multicast group at the beginning of simulation leading to the construction of the multicast tree. Each simulation run corresponds to 750 seconds of simulated time. After 30 seconds, one group member starts transmitting data packets at Constant Bit Rate (CBR) flow of 2 packets per second.

The metric used to evaluate the impact of partitioning attacks is the Packet Delivery Ratio (PDR) which is defined as the ratio of the total number of data packets received by multicast group members to the product of the number of data packets sent and the number of group members.

In this paper, researchers investigated the security of MAODV, which is a representative of tree-based multicast routing protocols for ad hoc networks. The study identified several attacks on MAODV. The goal of these attacks is either to create a partition in the multicast tree or to build an energy inefficient multicast tree. Simulation results confirm that these attacks disrupt the normal operation of MAODV to a large extent.

Several research works are currently going related to security aspect of multi cast routing protocol. Here we tried to present only those papers that are strongly related to our study only

# Chapter Five

# Attack Modeling in Ns2

## Introduction

As MANET is vulnerable to various attacks in different layers of protocol stack, modeling attack requires focusing a specific layer, designing and implementing the network components so that an analysis of performance metrics from the simulation result can be obtained. It has been a big challenge for a network layer routing protocol to function correctly and efficiently in the presence of malicious node which attempts to disrupt the routing service. Routing attacks can generally be characterized into routing, disruption and resource consumption by not forwarding the packets or adding and modifying some parameters of routing messages [22]. In this chapter, we will describe model of attack and how to incorporate it into ns2 simulator and specific functions to be modified in PUMA protocol.

### 5.1 Attack Modeling

A node in NS-2 is a compound object which is composed of a node entry object and classifiers [22] as shown in Fig-4.1.

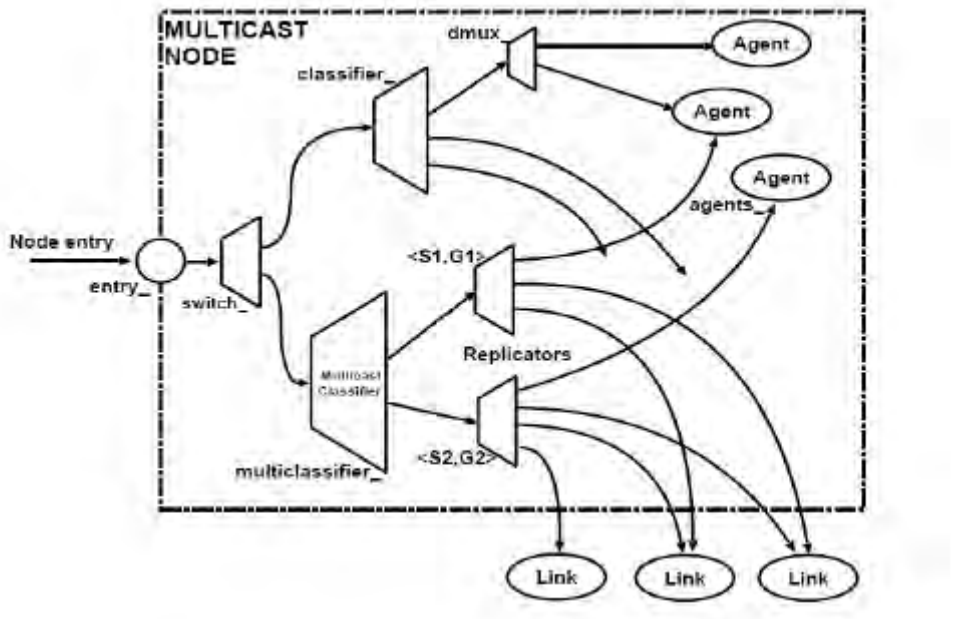



Fig. 4.1: Internal Structure of Multicast Node

NS -2 has an address classifier that does unicast routing and port classifier. A multicast node shown in Fig-4.1, in addition has a classifier that classify multicast packets from unicast packets and a multicast classifier that performs multicast routing

Unlike the real network packet, an NS-2 packet is composed of a stack of headers, and an optional pay load. A packet header format is initialized when a simulator object is created where a stack of all registered headers such as common header that is commonly used by any objects as needed. IP header, TCP header, FTP header and trace header is defined and the offset of each header in the stack is recorded so that any network object can access any header in the stack of a packet using the corresponding offset value

The packet header is analyzed by the classifier and forwarded to an outgoing interface which is the next downstream object in the network. The actual processing of the packet received by the node is done by the agent. An agent is a service or connection such as TCP/UDP with which two nodes in the network are connected. An agent's functionalities such as send, receive, forward and drop can be manipulated to launch an attack. Given below is the list of actions that are taken by a node agent upon receiving a packet;

- Extract the IP header of the packet to determine source and destination address.
- Extract the common header to determine packet type, size, next hop, previous hop etc
- Extract protocol specific header of the packet
- If the packet has already been received or has information that is older than it currently has or the packet has been generated by itself, then discard the packet by dropping
- If the packet has latest information then forward the packet to the next hop if it has a route to the next hop.
- If the destination of the control packet is the node itself then generate a route reply packet and send it to the previous hop in the packet.

In our design, we created a UDP agent and attached it to each node allowed to send with a specific multi cast address 0xE000000 and destination port of 100 as shown below. Every sender node is allowed to start sending at specific time.

```
set udp_(0) [new Agent/UDP]
$udp_(0) set dst_addr_ 0xE000000
$udp_(0) set dst_port_ 100
$ns_ attach-agent $node_(0) $udp_(0)


set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.1
$cbr_(0) set random_ 2112
$cbr_(0) set maxpkts_ 10000
$cbr_(0) set dst_ 0xE000000
$cbr_(0) attach-agent $udp_(0)
$ns_ at 2.00000000000000 "$cbr_(0) start"
```

Since it is a multi-cast communication, nodes wishing to join a multi cast group are supposed to call join-group and leave-group functions in our TCL script which will latter invoke corresponding functions in puma.cc files as show below( complete TCL script we created will be annexed)

```
Node instproc join { group } {
    $self instvar ragent_
    set group [expr $group]

    $ragent_ join $group
}

Node instproc leave { group } {
    $self instvar ragent_
    set group [expr $group] ;

    $ragent_ leave $group
}
```

Once multi cast group formation is enabled, we identified selected nodes as attacker in our TCL script as shown below and for each node set as an attacker , the simulator does appropriate action enabled in the source code of the protocol

```
$ns_ at 0.0 "[$node_(9) set ragent_] hacker"
$ns_ at 0.0 "[$node_(13) set ragent_] hacker"
```

**5.2 Modification of Codes**

The header file of PUMA puma.h contains the class definition and we add a Boolean variable as follows. This variable helps us to identify whether a node is set as an attacker or not

class PUMA: public Agent {

...

bool malicious;

...

};

The constructor of the class shown below initializes the member variable malicious to false initially

PUMA::PUMA(nsaddr_t new_id) : Agent(PT_PUMA), message_cache()

Once packets are received at each node, the agent has to determine whether the node is set as an attacker. If the node is set as attacker, the agent has to take an action corresponding to each attack model, this requires writing set of codes which represent an action and incorporate it into functions available in puma.cc file.

The first is to enable malicious property if a node is set as hacker node as

```
PUMA::command(int argc, const char*const* argv) {

      . . . . . .

          if (argc == 2) {
           Tcl& tcl = Tcl::instance();


           if (strncasecmp(argv[1], "id", 2) == 0) {
               tcl.resultf("%d", id);
               return TCL_OK;
           }
       if(strcmp(argv[1],"hacker") == 0) {
          malicious = true;
         return TCL_OK;

   . . . . . .

  }
```

Then if a nodes attacker property is set , we have to enable actions of black hole attacker as

```
PUMA::recv(Packet* p, Handler*) {
. . . . .
              hdr_cmn *ch = HDR_CMN(p);
              hdr_ip  *ih = HDR_IP(p);
              if (malicious == true ) {
           drop(p, DROP_RTR_ROUTE_LOOP);

        }
              else if ((HDR_CMN(p)->ptype() == PT_PUMA))
                     handle_protocol_packet(p);
              else if ((ih->saddr() == id) && (ch->num_forwards() == 0))
                     handle_data_from_transport(p);
              else
                     handle_data_packet_from_network(p);

. . . .
}
```

Since jelly fish attacker has effect of adding random extra delay before actually forwarding, it has to first check if a node is set as attacker and then modify the delay variable by adding random value on it

```
PUMA::routing_set_timer(RoutingEvent* event, double delay)
{
 . . . .

Scheduler::instance().schedule(routing_timer, event, delay+jellydelay);


. . . .
}
```

# Chapter Six

# Simulation Result and Discussion

## 6.1 Introduction

The simulations are implemented using Network Simulator *NS-2* [30], which is an open- source simulation tool that runs on Unix/Linux/Windows. *NS-2* is an object-oriented, discrete event driven network simulator developed at UC Berkley written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT). It implements network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR (continuous bit rate), and route queue algorithms include fair queuing, deficit round-robin and FIFO, routing algorithms such as Dijkstra, and more. *NS-2* has provisions for implementing multicasting and some of the MAC layer protocols and LAN simulations. Current *NS-2* has already included wireless ad hoc networking protocols such as PUMA which needs additional work to simulate multi cast features [20]

## 6.2 Generating Traffic and Mobility Models

To run the simulation, we generated traffic and mobility models, which are appropriate for multi cast communication which are called by tcl script simulation code at *NS-2*.

### 6.2.1 Traffic Models

Random traffic connections of TCP and CBR can be setup between mobile nodes using CMU's traffic-scenario generator scripts [12][26]. This traffic generator script is available under ~ns/indep-utils/cmu-scen-gen and is called cbrgen.tcl. It can be used to create CBR and TCP traffics connections between wireless mobile nodes. The command line looks like the following:

ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]

We created a CBR connection files, having 25 nodes, which are allowed to join the multi cast group by specifying multi cast address in the call to join at the specified time

```
$ns_ at 0.010000000000000 "$node_(0) join 0xE000000"
$ns_ at 0.010000000000000 "$node_(1) join 0xE000000"
$ns_ at 0.010000000000000 "$node_(2) join 0xE000000"
```

These nodes are allowed to leave by calling leave group function as

```
$ns_ at 50.000000000000000 "$node_(21) leave 0xE000000"
```

$ns_ at 50.000000000000000 "$node_(22) leave 0xE000000"

### 6.2.2 Mobility models

The node-movement generator script for generating the Random Waypoint mobility model is available under ~ns/indep-utils/cmu-scen-gen/setdest directory. Run setdest with arguments as shown below:

./setdest [-n num_of_nodes] [-p pausetime] [-s speedtype] [-m minspeed] [-M maxspeed] [-t simtime] [ -P puasetype] [-p pausetime] [-x maxx] [-y maxy] \

We created scenarios consisting of 25 nodes moving with variable speed which is 5-10m/s, 10-15/ms, 15-20m/s and 20-25m/s. It is realistic scenario that node may move with variable speed, that is why our scenario considered variable speed case that can be specified between range of values. In each of the scenarios the number of nodes will remain fixed and the node movement speed will vary. The simulation will stop after 500 s and the topology boundary is defined as 1000 meters X 1000 meters.

Table 6.1 Scenario table

| **Scenario** | **Number of Node** | | | | **Speed(m/s)** |
|---|---|---|---|---|---|
| | **Total** | **Sender** | **Receiver** | **Attacker** | |
| Black Hole attack with varied number of receivers | 25 | 5 | 10,15,20 | 1 | 5-10<br>10-15<br>15-20<br>20-25 |
| Multicast communications with varied number of receivers | 25 | 5 | 10,15,20 | 0 | |
| Black Hole with varied number of attacker nodes | 25 | 5 | 15 | 1,2,3 | |
| Jelly fish attack | 25 | 5 | 15 | 1 | |
| Impact of attacker positions | 25 | 5 | 15 | 1 | |

### 6.3 Simulation and Parsing of trace files

After arranging the necessary parameters, we run the simulation and the trace files are recorded the traffic and node movements are generated after the simulation. Details of information are organized by writing a C++ code and organized into table format these files need to be parsed in order to extract the information needed to measure the performance metrics.

### 6.4 Results and discussion

Simulation results of all scenarios under different parameters are discussed by showing the graph for each case

#### 6.4.1 Comparison of attack and non-attack scenarios

To demonstrate the result of our design, in each of the scenarios we simulated both the attacked and non-attacked cases.

##### 6.4.1.1 Impact of Black hole attack with varying number of receivers

**One attacker and varying number of receivers**

- **PDF versus speed**

Here the simulation is performed for 5 sender nodes and varying number of receivers. The nodes are moving with variable speed of ranges of values as described in the simulation setup section. The simulation consists of one attacker node at random position in the network and receiver nodes vary as 10, 15, and 20 nodes. As can be seen in above graph, as the number of receiver node increases the impact of attacker node decreases on average taken at each speedby 15.76 % decrease in PDF in going from 20 to 15 receivers and on average 15.36 % decrease in PDF in going from 15 to receivers. This is due to the fact that more number of receivers results in a denser routing mesh providing alternate paths for the data packets

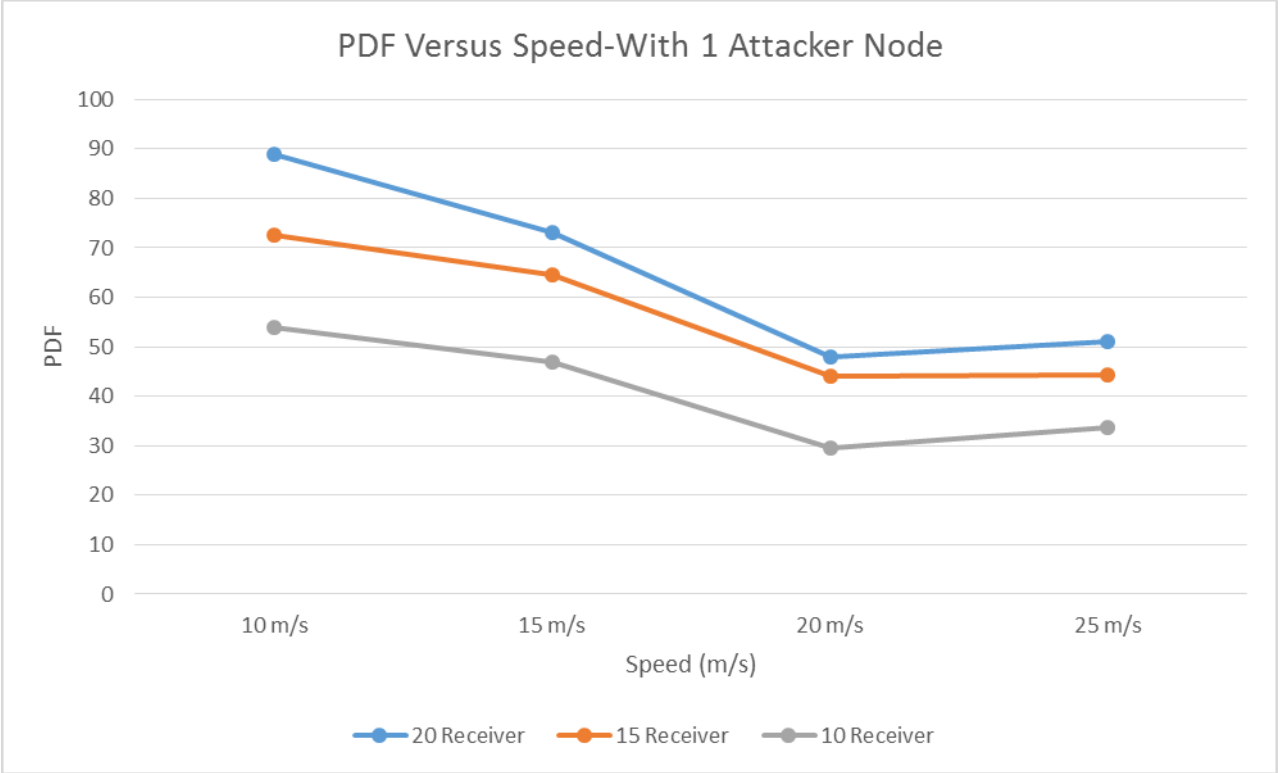



Fig. 6.1 PDF Versus Speed-one attacker varied number of receivers

.

- **End to End delay versus speed**

The average end to end delay has not showed any noticeable effect due to the presence of attacker node with increase in number of receivers nodes increases as shown in the below figures

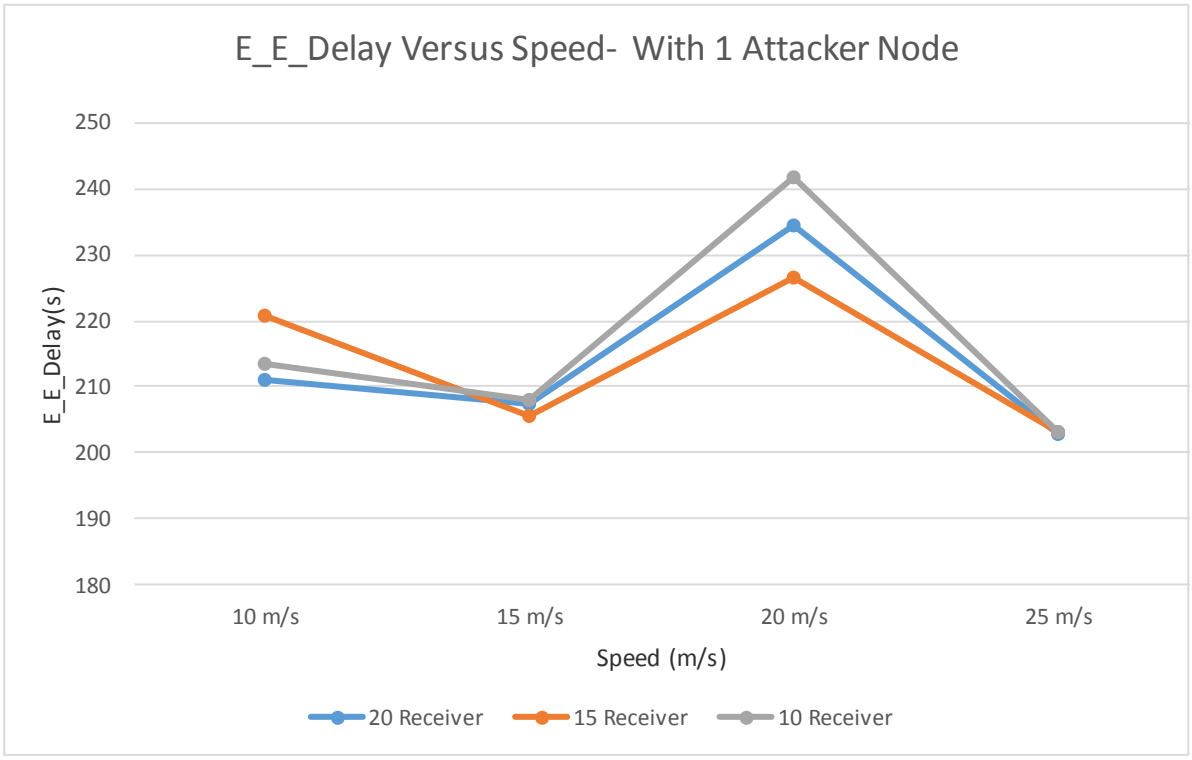



Fig. 6.2 End to End delay versus Speed- One attacker varied number of receivers

- **GoodPut Versus Speed**

The increase in number of receiver node, decreases the impact of attacker node which leads to increases in total number of packets reaching to destination. 16% increase in good put in going from 15 to 20 receivers in the presence of attacker node. There is 13.1% increase in Goodput as the number of receiver increases from 10 to 15 nodes in the presence of one attacker node as shown in the below figure

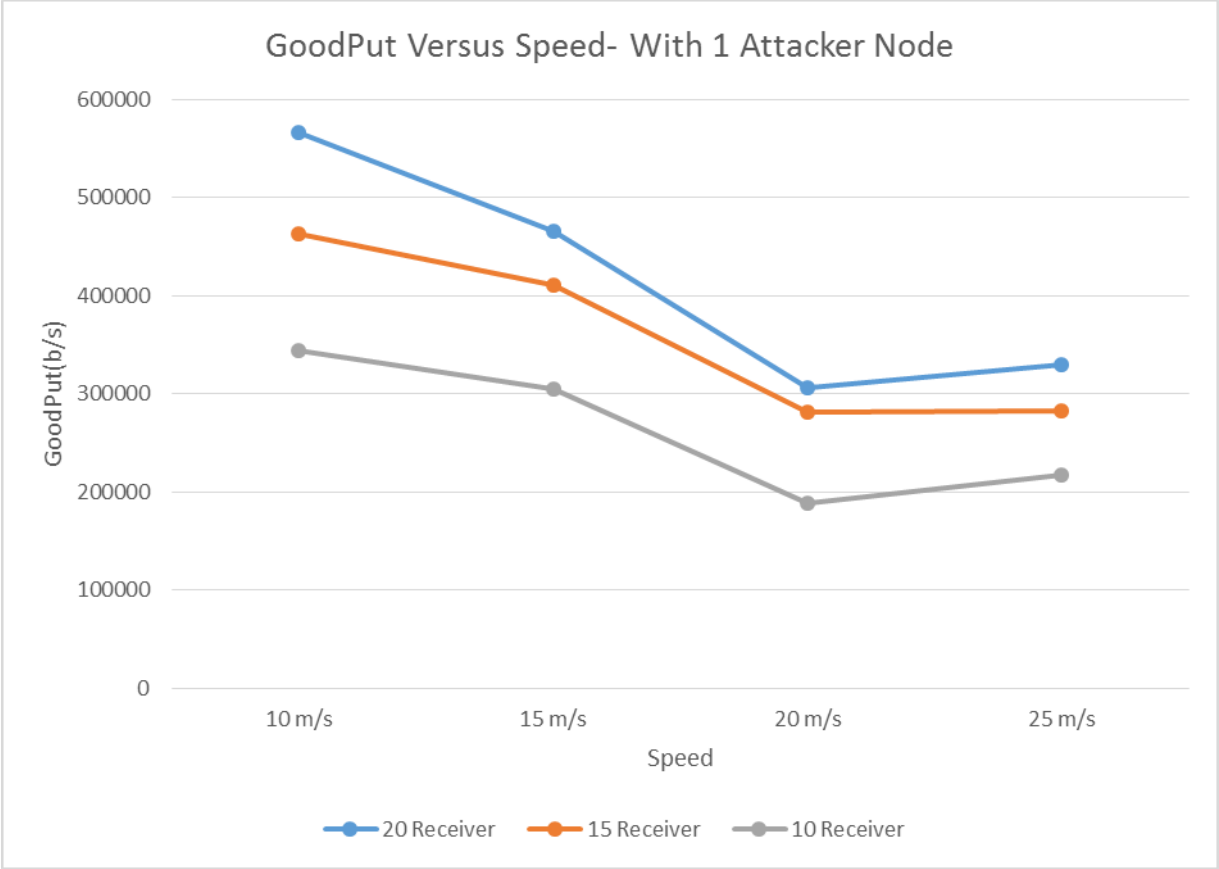



Fig 6.3 GoodPut Versus Speed- one attacker varied number of receivers

Here the simulation is performed for 5 sender nodes and varying number of receivers. The nodes are moving with variable speed of ranges of values as described in the simulation setup section.

**6.4.1.2 Multi cast communication with, effect of varying receiver**

Multi cast communication is a form of one to many or many to many, here we tried to show that, an increase in number of receiver nodes for the same number of sender increases the total number of packets reaching to destination. This has the overall effect of increasing the PDF

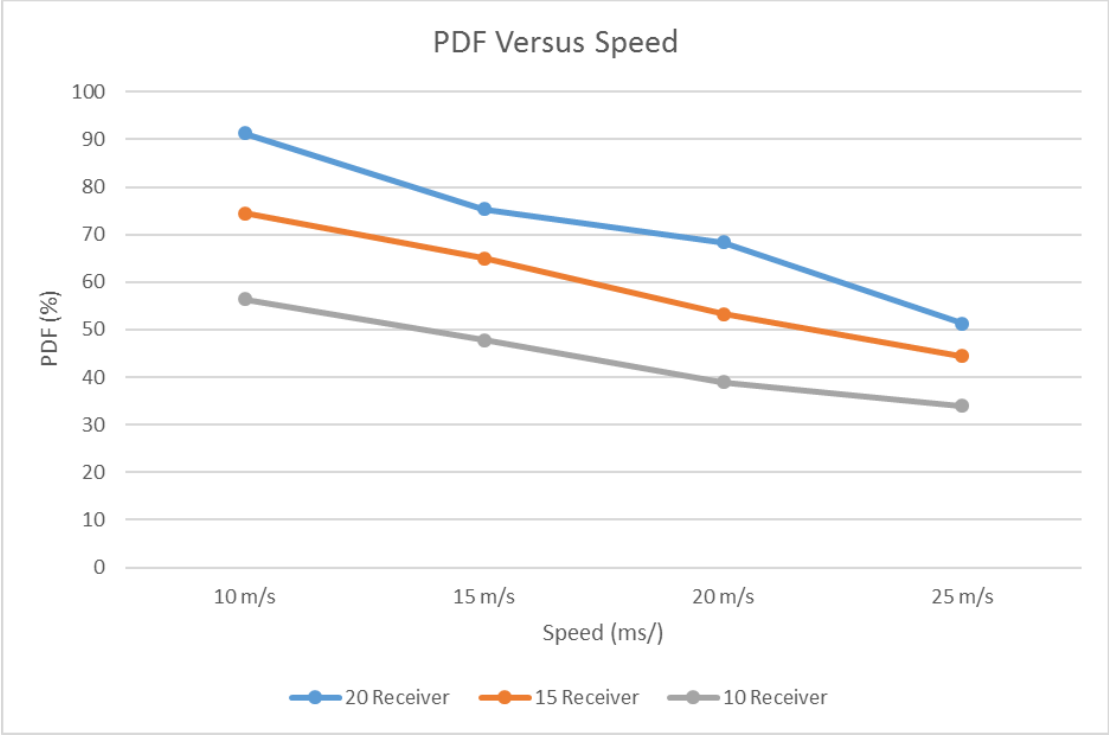



Fig 6.4 PDF versus Speed- Effects of varying receivers

Goodput increases from our simulation for 5 sender nodes and receivers vary from 10, 15 and 20 nodes. The comparison the below two figures shows the results obtained from the simulation. The more the number of receivers the more data packets successfully reaching to destination which increases the goodput at each case as shown in fig 6.5

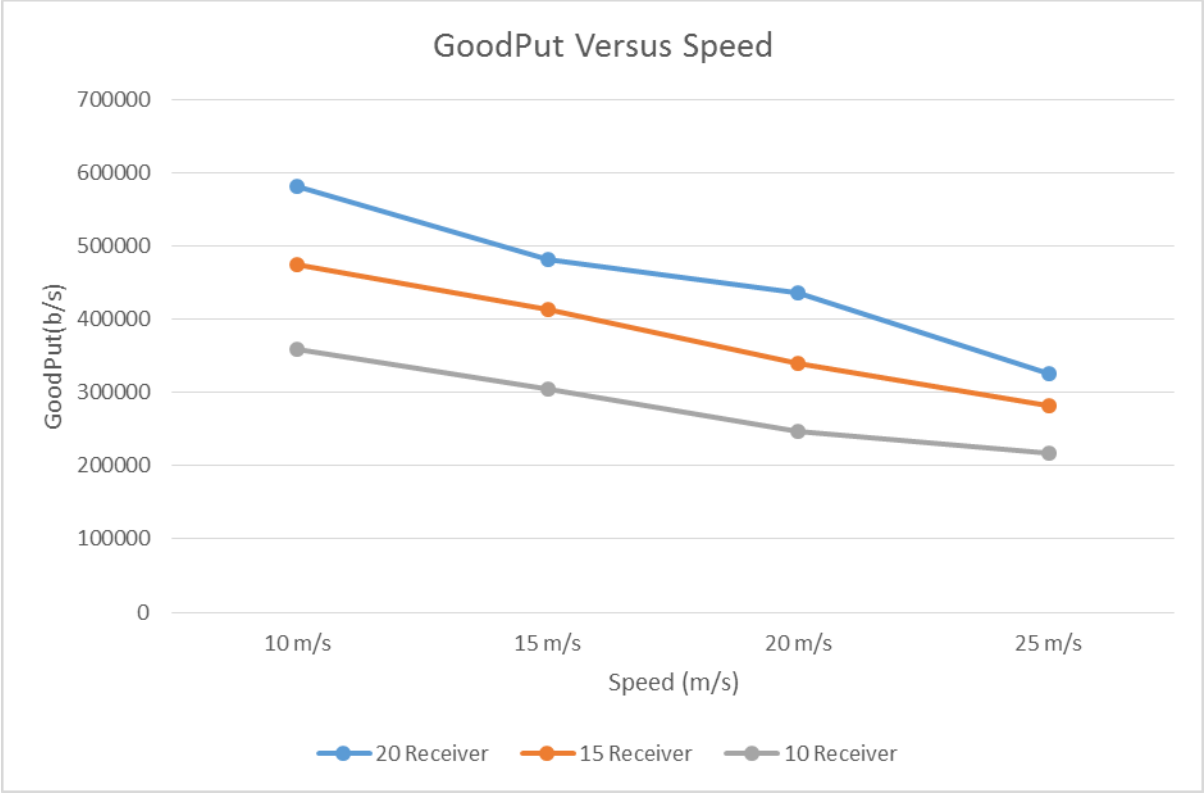



Fig. 6.5 Goodput Versus Speed- Effects of varying receivers

#### 6.4.1.3 Black hole attack with varying number of attacker nodes

In this scenario we considered the impact of attack with increase in number of attacker nodes. The simulation is performed keeping the number of sender nodes to 5 and receiver nodes to 15.

- **PDF Versus speed**

Here we run simulation for 1, 2 and 3 attacker nodes and results of PDF is shown in the below graph

From the simulation results it is observed that, the packet delivery fraction reduces with increased mobility of the nodes and also with increased number of black hole nodes and affect the performance of the network. As the above graphs shows, the more the number of attacker nodes, the lesser the packet delivery fraction due more number of packets dropped before reaching to destination. The numeric results shows that there is an overall 7-8 % average decrease in PDF with increase in number of attacker node from 1 to 2 and from 2 to three

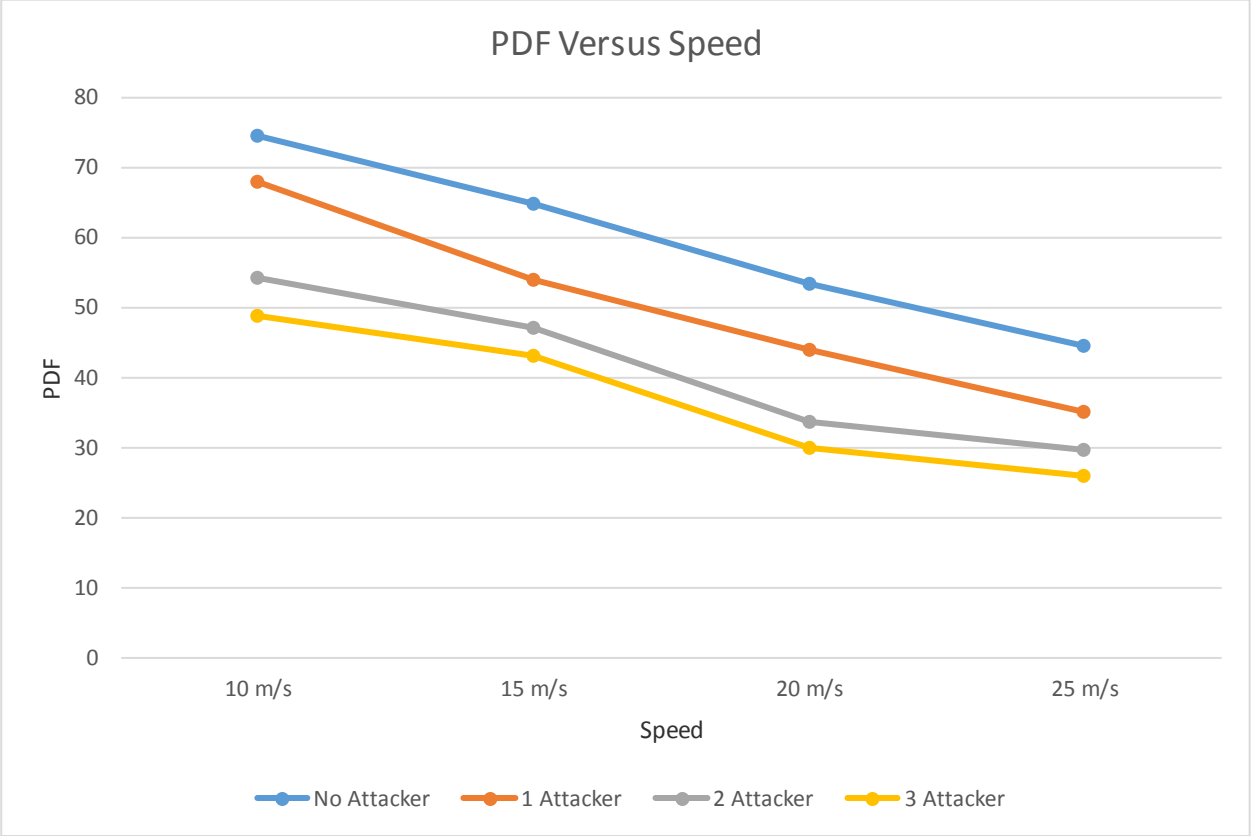



Fig. 6.6 PDF versus speed- Varied number of attacker nodes

- **Goodput versus speed**

By the similar reasoning given to PDF, the more the number of attacker nodes the lesser the goodput. The increase in number of attacker node increase the total number of packets dropped before reaching to destination. This will result in decrease in goodput values as shown in the graph below

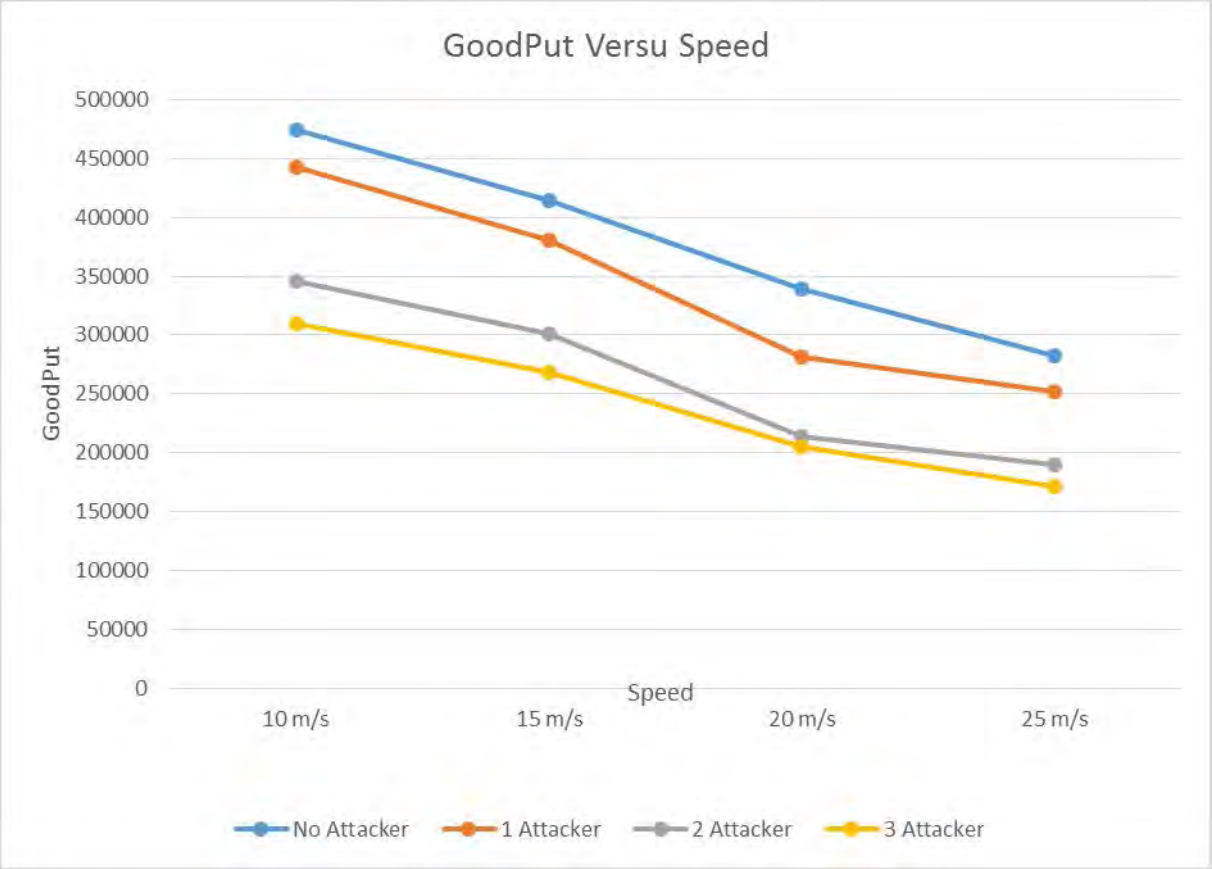



Fig. 6.7 Goodput versus speed- Varied number of attacker nodes

- **End to end delay versus speed**

The below figure(h) shows the variation of end to end delay for different number of attackers in the presence of 5 sender and 15 receivers. There seems to be an increase in the delay with increase in the number of attacker nodes from 1 to 2 an from 2 to three. This effect of black hole attack on end to end delay is relatively lesser compared to PDF and Goodput. At a speed of between 15m/s to 25m/s the end to delay is seen to increase with increase in number of attacker nodes

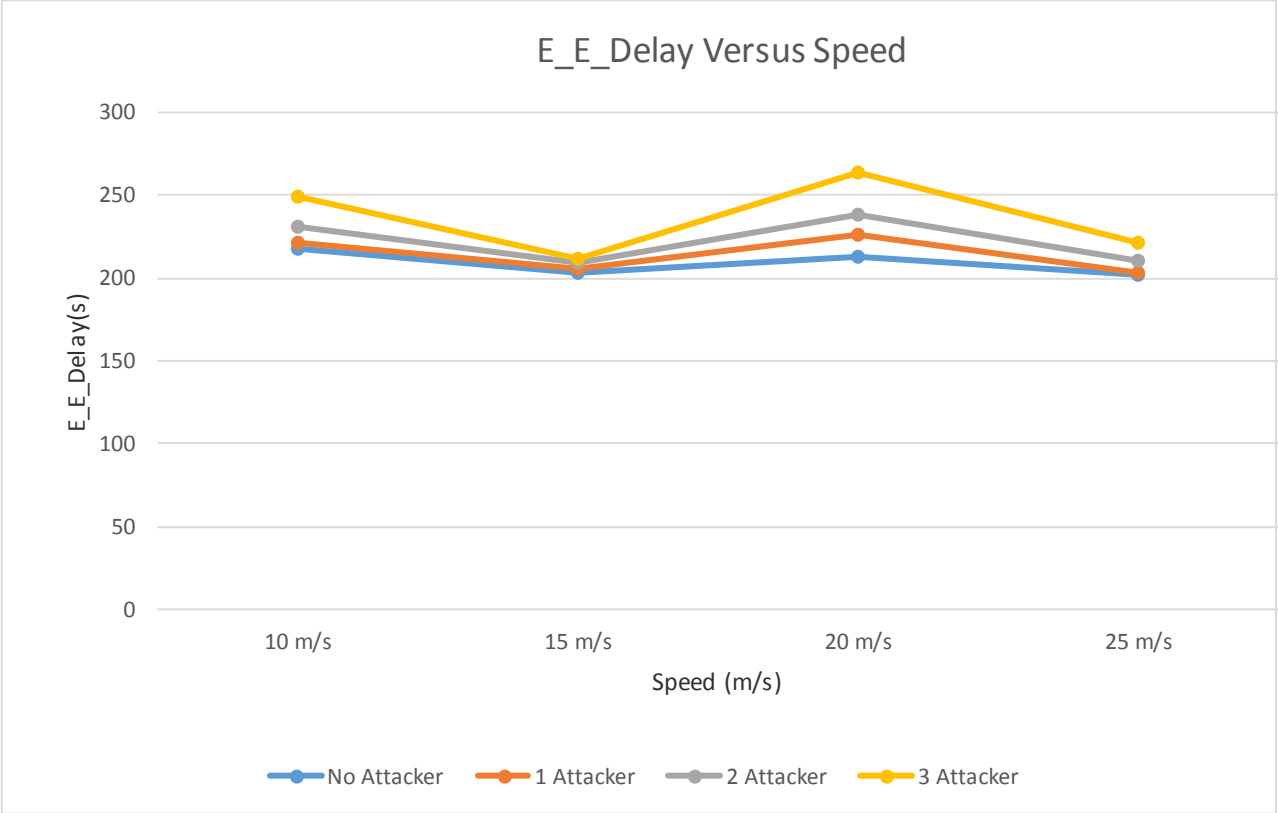



Fig. 6.8 End to End delay versus speed-Varied number of attacker

**6.4.1.4 Jellyfish attack**

Jelly fish attacker node after successfully invading into the network, it delay data packets for random period of time. This we have seen using 5 sender and 15 receiver node. Here the number of attacker node is set to one. The effect can be seen in the fig 6.9 shown below.

Other performance metrics simulated didn't show any reasonable change in their values due to the presence of jelly fish attacker nodes.

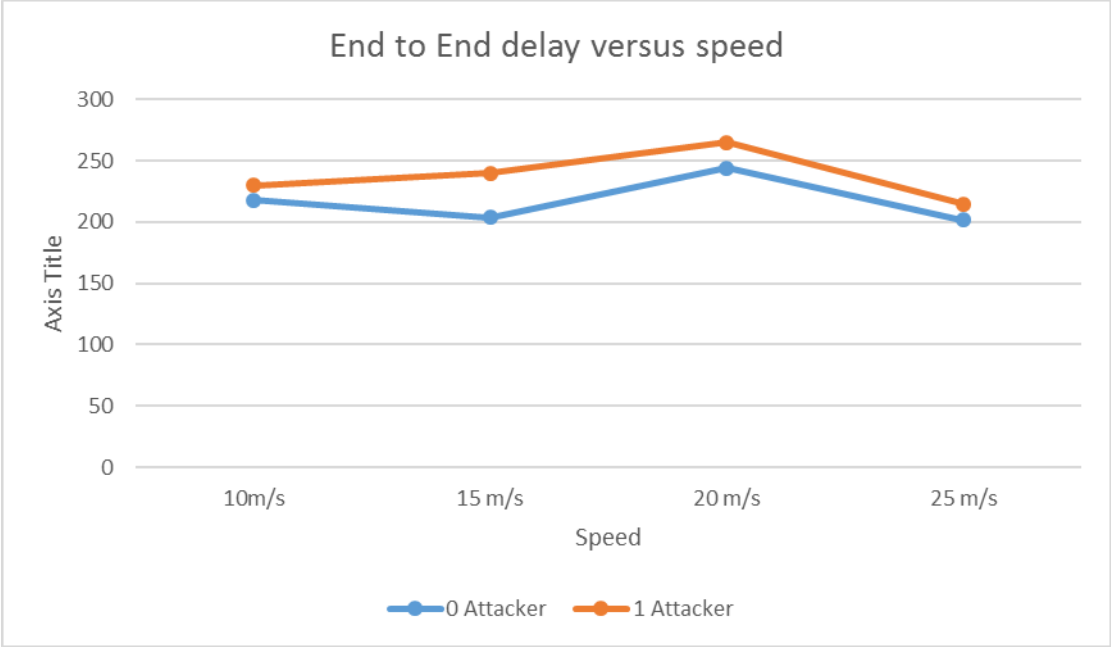



Fig. 6.9 End to End Delay versus Speed- Jelly fish attack

### 6.4.1.5 Impact of attacker position

The previous scenarios take into consideration of attacker node positioned randomly over the network. There are situation that the attacker node position might have its own factor on attacking. In our simulation we considered attacker node positioned near sender and attacker near receiver which is one hop away. We have simulation consisting of 5 sender and 15 receiver nodes. Attacker near sender has higher probability of getting access to packets coming out of sender nodes which makes PDF Values to decrease at each on average value of 6 % as shown in fig 6.10

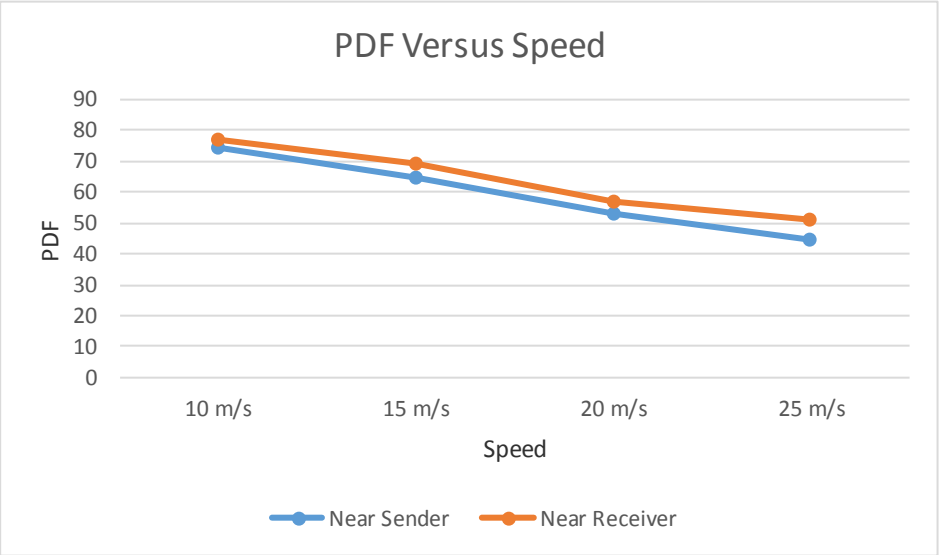



Fig. 6.10 PDF versus speed- Attacker near sender/receiver one hop away

The simulation also considered the impact of attacker node positioned near sender and near receiver one hop away. As the results show that attacker positioned one hop away from the sender has higher probability of getting access to packets coming out of sender nodes, the total numbers of packets reaching successfully to destination decreases. As a result the goodput value decreases as well. The case of attacker positioned one hop away from the receiver node has lesser impact of dropping packets compared with that of near sender cases

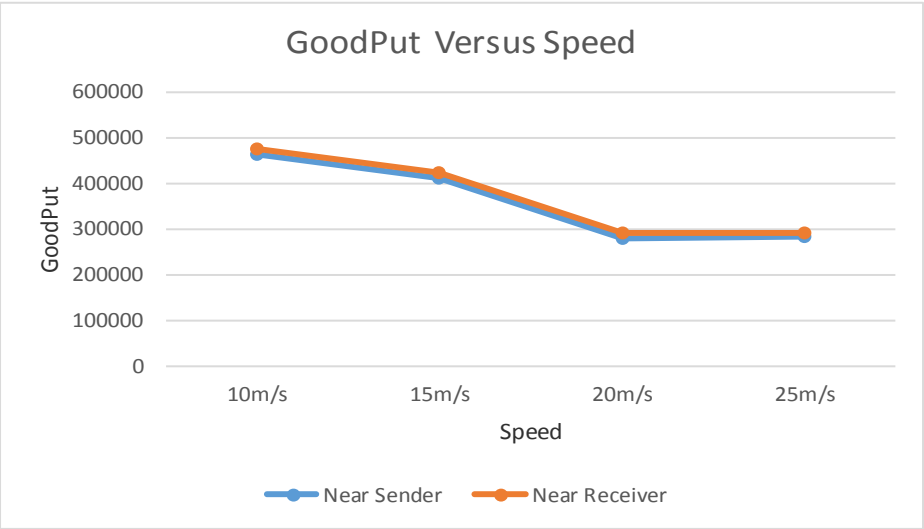



Fig 6.11 Goodput Versus Speed- Attacker near sender/receiver one hop away

# Chapter Seven

# Conclusion and Recommendation

In this thesis we thoroughly analyzed the performance issues of the MANETS multi cast routing protocol with respect to security and tried show how the performance is affected due to the presence of black hole and jellyfish attacks

The research on MANET security is still in its early stage. The existing proposals are typically attack-oriented in that they first identify several security threats and then enhance the existing protocol or propose a new protocol to thwart such threats. Because the solutions are designed explicitly with certain attack models in mind, they work well in the presence of designated attacks but may collapse under unanticipated attacks. Therefore, a more ambitious goal for ad hoc network security is to develop a multi-fence security solution that is embedded into possibly every component in the network, resulting in depth protection that offer multiple line of defense against many both known and unknown security threats

In our study a complete analysis is done on the PUMA protocol and the vulnerabilities of the protocol are identified such as black hole attack, jelly fish attack.

Simulation results show the presence of attacker node in the network has the effect of decreasing essential performance metrics of the protocol like PDF, End to end delay and Goodput.

Since this paper is focusing on multicast routing protocol, we attempted to show the increase in group size by increasing the number of receivers decrease the impact of attacker node. Ad hoc network size can vary dynamically and number of attacker node can also increase or decrease. In our simulations the more the number of attacker node, the more the impact

of attack which leads to considerable percentage of decrease in PDF and goodput values

Random position of attacker node and specific position of attacker node were also considered in our simulation. Since dynamic change of topology is one of the feature of ad hoc network, attacker node can be near sender node at one point of time and near receiver at some other time, where results show that attacker positioned near sender drops more packets than attacker near receiver which leads an overall decrease in PDF and good put

Since little work has been done so far on the analysis of PUMA protocol compared with other tress and mesh based routing protocol , we believe this study contributes it own share of performance analysis of the protocol due attacker nodes. We also tried to show simulation that node are moving with variable speed which is the most likely scenario in the real world from the pedestrian to high speed scenarios

Future work can be done to study the effect of other types of attacks and comparison of PUMA protocol with other mesh based routing protocols and secure solution to thwart against those attacks, mitigation of attacks with band width and energy consumption

# Bibliography

[1] C. Gui and P. Mohapatra, "Efficient Overlay Multicast for Mobile Ad Hoc Networks," *In the Proceedings of IEEE WCNC'03, New Orleans*, LA, Mar., 2003

[2] Khalid A. Farhan, "Network sender multicast routing protocol", *Proceedings of seventh IEEE International conference on networking*, 2008, pp. 60-65

[3] R. Ramanathan and M. Steenstrup, "Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support," *ACM/Baltzer Mobile Networks and Applications*, vol. 3, no. 1, Jun. 1998, pp. 101-119

[4] Ravindra Vaishampayan, J.J. Garcia-Luna-Aceves, "Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks", proc. of IEEE Conference, pp: 304-313, 2004

[5] S. Desilva, and R. V. Boppana, "Mitigating Malicious Control Packet Floods in Ad Hoc Networks," Proc. IEEE Wireless Commun. and Networking Conf., New Orleans, LA, 2005

[6] A. Banerjee and P. Dutta, "A Survey of Multicast Routing Protocols For," *International Journal of Engineering Science and Technology*, 2010

[7] E. Anita and V.Vasudevan, "Performance Evaluation of Mesh based Multicast Reactive Routing Protocol under Black Hole Attack," *International Journal of Computer Science and Information Security,* vol. 3, no. 1, 2009

[8] E. Royer, and C. E. Perkins, "Multicast operation of the ad hoc ondemand distance vector routing protocol", *In the proceedings of MobiCom,* 1999

[9] H. Labiod and H. Moustafa, "The Source Routing-based Multicast Protocol for Mobile Ad Hoc Networks (SRMP)", *Internet draft, IETF*, November 2001

[10] H.Deng, H.Li, and D.P.Ararwal, "Routing security in wireless Ad hoc networks", IEEE Communication magazine. Vol. 40, No.10. Oct.2002

[11] Hoang Lan Nguyen: A Study of Security Attacks on Multicast in Mobile Ad Hoc Networks, York University, 2002

[12] http ://www.isi.edu/nsnam/ns/ns-ducumentation. Html

[13] J. L. Sobrinho and A. S. Krishnakumar, ''Quality-of-service in ad hoc carrier sense multiple access wireless networks,'' *IEEE J.* Select. Areas, Commun.vol. 17, No. 8, Aug. 1999, pp. 1353–1414

[14] M. A. Amuthan and D. N. Abirami, "Multicast security attack and its countermeasures for PUMA protocol," *Int. J. Comp. Tech.,* vol. 2, no. 3, 2009.

[15] Mauve, M., Füßler, H., Widmer, J., Lang, T., "Poster: Position-Based Multicast Routing for Mobile Ad-Hoc Networks", *In Proceedings of Fourth ACM International Symposium on Mobile Ad Hoc Networking And Computing: MobiHoc*, 2003.

[16] Mehul Revankar "Attacks in Ad-Hoc Networks and Modeling in NS-2" University of Mumbai,2004

[17] N.SHANTHI, DR.LGANESAN and DR.K.RAMAR, "Study Of Different Attacks On Multicast Mobile Ad Hoc Network," *Journal of Theoretical and Applied Information Technology,* 2009.

[18] P. Chaporkar, A. Bhat, and S. Sarkar, "An Adaptive Strategy for Maximizing Throughput in layer Wireless Multicast," *MobiHoc'04,* pp.256-267 , Japan, Roppongi, May 2004

[19] P. Goyal, S. Batra and A. Singh, "A Literature Review of Security Attack in Mobile Ad-hoc," *International Journal of Computer Applications (0975 – 8887),* November 2010.

[20] ] S. C. Mandhata, S. Patro and S. Mohanty, "Exploration of security threats and its performance impact on Mobile Ad-Hoc Networks using NS-2," *International Journal of Computer Applications,* vol. 45, no. 8, May 2012

[21] S. Parthiban, A. Amuthan, N.Shanmugam and K. Joseph, "Neighbor Attack and Detection Mechanisms in Mobile Ad hoc Network*," Advanced Computing: An International Journal ( ACIJ ),*, vol. 3, no. 2, March 2012.

[22] S. Roy, V. G. Addada, S. Setia and S. Jajodia, "Securing MAODV: Attacks and Countermeasures," *Center for Secure Information Systems, George Mason University, Fairfax, VA 22030,* 2005

[23] S. Sen, J. A, C. Juan and E. Tapiador, "Security Threats in Mobile Ad Hoc . Networks," *Department of Computer Science,*, 2010

[24] Sung-Ju Lee, William Su, and Mario Gerla, "On-demand multicast routing protocol (ODMRP) for ad hoc networks", *Internet Draft*, draftietfmanet-odmrp-02.txt, 2000.

[25] T.-W.Chen and M.Gerla, ``Global State Routing: A New Routing Scheme for Adhoc Wireless Networks," *In Proceedings of IEEE ICC'98*, Atlanta, GA, pages 171-175, Jun. 1998

[26] The Monarch Project of Carnegie Mellon University, http://monarch.cs.cmu.edu/

[27] V. PALANISAMY and P.ANNADURAI, "Impact of Rushing attack on Multicast in Mobile Ad Hoc Network," *International Journal of Computer Science and Information Security,* vol. 4, no. 1, 2009

[28] W. L. a. A. Joshi, "Security Issues in Mobile Ad Hoc Networks," *Department of Computer Science and Electrical Engineering,* 2010.

[29] Wei Gong, Zhiyang You, Danning Chen, Xibin Zhao, Ming Gu and Kwok-Yan Lam, " Trust Based Malicious Nodes Detection in MANET", *IEEE*, 2009.

[30] Z. J. Haas, M. Gerla, D. B. Johnson, et al., ''Guest editorial,'' *IEEE J*. Select. Areas Commun., Special issue on wireless networks, vol. 17, no. 8, Aug. 2003, pp. 1329–1332

[31] Zhijiang Chang, Georgi aydadjiev ;Routing Protocols for Mobile Ad-hoc Networks: Current Development and Evaluation, Stamatis Vassiliadis *Computer Engineering laboratory, EEMCS, Delft University of Technology*, 2003

[32] Zhijiang Chang, Georgi aydadjiev, Stamatis Vassiliadis ;Routing Protocols for Mobile Ad-hoc Networks: Current Development and Evaluation Computer Engineering laboratory, *EEMCS, Delft University of Technology*,2003

## Appendices

## Appendix A:TCL script

```
# ======================================================================
# Define options
# ======================================================================

set val(chan)       Channel/WirelessChannel
set val(prop)       Propagation/TwoRayGround
set val(netif)      Phy/WirelessPhy
set val(mac)        Mac/802_11
set val(ifq)        Queue/DropTail/PriQueue
set val(ll)         LL
set val(ant)        Antenna/OmniAntenna
set val(x)              1000   ;# X dimension of the topography
set val(y)              1000  ;# Y dimension of the topography
set val(ifqlen)         50            ;# max packet in ifq
set val(seed)           1.0
set val(adhocRouting)       PUMA
set val(nn)             25           ;# how many nodes are simulated
set val(cp)             "pumacbr"
set val(sc)           "./pumascen/pumascena10"
set val(stop)           500.0           ;# simulation time

# ======================================================================
# Main Program
# ======================================================================

#
# Initialize Global Variables
#

# create simulator instance

set ns_              [new Simulator]
set topo      [new Topography]
#$topo load_flatgrid $X $Y


# create trace object for ns and nam

set tracefd   [open wk.tr w]
set namtrace    [open puma.nam w]
$ns_ use-newtrace
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)


# define topology
$topo load_flatgrid $val(x) $val(y)

#
# Create God
#
set god_ [create-god $val(nn)]

#
# define how node should be created
#

#global node setting
```

```
$ns_ node-config -adhocRouting $val(adhocRouting) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -channelType $val(chan) \
               -topoInstance $topo \
               -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF
               #-channel $chan

#
#  Create the specified number of nodes [$val(nn)] and "attach" them
#  to the channel.

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


#
# Define node movement model
#
puts "Loading connection pattern..."
source $val(cp)

#
# Define traffic model
#
puts "Loading scenario file..."
source $val(sc)

# Define node initial position in nam

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

    # 20 defines the node size in nam, must adjust it according to your
scenario
    # The function must be called after mobility model is defined

     $ns_ initial_node_pos $node_($i) 20
 }


#here
#$ns_ at 0.0  "$node_(9) hacker"
#$ns_ at 0.0  "[$node_(14) set ragent_] hacker"
#$ns_ at 0.0 "[$node_(9) set ragent_] hacker"
#$ns_ at 0.0 "[$node_(11) set ragent_] hacker"
#$ns_ at 0.0 "[$node_(13) set ragent_] hacker"

#use node 4 as attacker during 1 sender and 15 receiver


#
# Tell nodes when the simulation ends
#
```

```
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}



Node instproc join { group } {
    $self instvar ragent_
    set group [expr $group]

    $ragent_ join $group
}

Node instproc leave { group } {
    $self instvar ragent_
    set group [expr $group] ;

    $ragent_ leave $group
}



$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"

puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp $val(adhocRouting)"
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

puts "Starting Simulation..."
$ns_ run
```

## Appendix B: C++ code for parsing trace file and calculating parameters

```
BEGIN {

             sends=0;
             flag=0;

             recvs=0;

             routing_packets=0.0;

             droppedBytes=0;
            check=0;
             droppedPackets=0;

             highest_packet_id =0;

             sum=0;
             summean=0.0;
             count=0;
            avghop=0;
             recvnum=0;
             noerrors=0;
             nodied=0;
             firsttime=0.0;
              EnergyConsume=0.0
              variance=0.0;


          }



        {

        time = $3;

        packet_id = $41;
        NodeId=$9;
        NodeEnergy=$17;



        # CALCULATE PACKET DELIVERY FRACTION

        if (( $1 == "s") &&  ( $35 == "cbr" ) && ( $19=="AGT" )) {
sends++; }


             if (( $1 == "s") &&  ( $35 == "cbr" ) && ( $19=="MAC" )) {
        avghop++; }


        if (( $1 == "r") &&  ( $35 == "cbr" ) && ( $19=="AGT" ))   {
recvs++; }



        # CALCULATE DELAY

        if ( start_time[packet_id] ==0  )
{
 start_time[packet_id] = time;
```

```
#printf("   %.2f",start_time[packet_id]);
}

        if (( $1 == "r") &&  ( $35 == "cbr" ) && ( $19=="RTR" )) {
end_time[packet_id] = time;
#printf("   %.2f",end_time[packet_id]);
 }

             else {  end_time[packet_id] = -1;  }


        # CALCULATE TOTAL AODV OVERHEAD


       if (($1 == "s" || $1 == "f") && $19== "RTR" && $35 =="PUMA")
           routing_packets++;


        # DROPPED AODV PACKETS

        if (( $1 == "d" ) && ( $35 == "cbr" )  && ( $3 > 0 ))

             {

                   droppedBytes=droppedBytes+$37;

                   droppedPackets=droppedPackets+1;

             }


            #find the number of packets in the simulation

                if (packet_id > highest_packet_id)

                   highest_packet_id = packet_id;



        }



        END {


        for ( i in end_time )

        {

        start = start_time[i];

        end = end_time[i];

        packet_duration =  end-start;
```

```
        if ( packet_duration > 0 )

        {    sum += packet_duration;
       duration[i]=packet_duration;


             recvnum++;

        }

        }

               delay=sum/recvnum;

    for(i in duration)
      {
       variance+=(duration[i]-delay)*(duration[i]-delay);
      }
         variance=variance/recvnum;




           NRL = routing_packets/recvs;  #normalized routing load

          PDF = (recvs/sends)*100;  #packet delivery ratio[fraction]
          printf("  SEND          RECEIVE         PDF       E_E_DELAY
JITTER       GOODPUT       DROPPKT     DROPBYTES\n");
          printf("__________    ____________    _____      __________
__________   __________   _______    _________\n");
           printf(" %.2f",sends);

           printf("       %.2f",recvs);

          # printf("         %.2f",routing_packets++);

           printf("        %.2f",PDF);

          # printf("   %.2f",NRL);


           printf("      %.2f",delay);

           printf("       %.8f",variance);

           printf("     %.2f bps",recvs*512/20);
              #droped packet
          printf("      %d",droppedPackets);
          printf("        %d\n",droppedBytes);
#printf("avrage hop count   = %d\n",avghop);


        }
```