



UNDERGRADUATE THESIS REPORT

ON

**Simulation-Based Proportional Study of Routing  
Protocols for MANET**

By

Md. Nazmul Hasan (2012-3-80-038)

Md. Emil Sadekin (2013-1-80-027)

Faisal Ahmed (2013-1-80-026)

Submitted to the

Department of Electrical and Electronic Engineering

Faculty of Science and Engineering

East West University

In partial fulfillment of the requirements for the degree of Bachelor of Science in

Electrical and Electronics Engineering

(B.Sc in EEE)

Spring, 2017

# **Simulation-Based Proportional Study of Routing Protocols for MANET**

By

Md. Nazmul Hasan (2012-3-80-038)

Md. Emil Sadekin (2013-1-80-027)

Faisal Ahmed (2013-1-80-026)

Submitted to the

Department of Electrical and Electronic Engineering

Faculty of Science and Engineering

East West University

In partial fulfillment of the requirements for the degree of Bachelor of Science in  
Electrical and Electronics Engineering

(B.Sc in EEE)

Spring, 2017

Approved by

---

Thesis Supervisor

Fakir Mashuque Alamgir

---

Chairperson

Dr. Muhammed Mazharul Islam

Department of Electrical and Electronic Engineering

East West University

## **Approval**

This paper titled ‘Simulation-Based Proportional Study of Routing Protocols for MANET’ has been submitted by Md. Emil Sadekin (2013-1-80-027), Faisal Ahmed (2013-1-80-026) and Md. Nazmul Hasan (2012-3-80-038), session Spring, 2017 has been accepted as satisfactory in partial fulfillment of requirement of the degree of Bachelor of Science in Electrical and Electronics Engineering on April, 2017.

---

Dr. Muhammed Mazharul Islam

Assistant Professor and Chairperson

Department of Electrical and Electronic Engineering

East West University

Dhaka, Bangladesh

## **Abstract**

Wireless ad-hoc networks have recently gained significant research attention due to their vast potential of applications in numerous fields. Multihop routing is a significantly important aspect which determines, to a large extent, the overall performance of the network. A number of routing protocols have been proposed for routing in wireless ad-hoc networks with focus on optimizing different aspects of the network routing. This report focuses on studying two popular protocols for wireless networks: Ad-hoc On Demand Distance Vector (AODV) and Optimized Link-State Routing (OLSR). The two protocols belong to different classes of routing categorization. AODV is a popular on-demand (reactive) routing protocol whereas the OLSR is a popular link-state based proactive routing protocol. The technical aspects of the two protocols shall be studied while highlighting the differences between the two and simulation based performance comparison of the two protocols shall be carried out under varying traffic and network conditions using the Network Simulator.

**Keyword:** Wireless Ad-hoc network, routing protocol, Network Simulator.

## **Acknowledgement**

At first, we would like to be so thankful to our creator Almighty Allah for providing us mental and physical strength to conduct the research works and letting us successfully complete the entire research.

Secondly, we are so grateful to our honorable supervisor Mr. Fakir Mashuque Alamgir, Senior Lecturer, Dept. of Electrical and Electronics Engineering, East West University for his extensive supervision, guidance, patience from very early stage our work.

Yes, we have put forward a lot of effort here. However, we have very frequently sought help from many individuals and organizations. We thankfully acknowledge their enormous contributions.

Lastly, we are also thankful to our family members and our friends for their constant and silent mental support and encouragement.

## Authorization Page

This is to certify that this thesis is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

\_\_\_\_\_

Md. Emil Sadekin

\_\_\_\_\_

Md. Faisal Ahmed

\_\_\_\_\_

Md. Nazmul Hasan

We further authorize East West University to reproduce this project report by photocopy or other means, in total or in part, at the request of other institutions or individuals for purpose of scholarly research.

\_\_\_\_\_

Md. Emil Sadekin

\_\_\_\_\_

Md. Faisal Ahmed

\_\_\_\_\_

Md. Nazmul Hasan

East West University

Authors

Dhaka, Bangladesh,

April, 2017

## TABLE OF CONTENTS

	<b>Page Number</b>
Approval.....	1
Abstract.....	2
Acknowledgement.....	3
Authorization Page.....	4
Table of contents.....	5
List of Figures.....	9
List of Tables.....	11
CHAPTER 1: Introduction.....	12
1.1 Objective.....	12
1.2 MANETS.....	12
1.2.1 Types of MANETS.....	13
1.2.2 Initial configuration mechanisms of MANETS.....	13
1.2.3 MANETS characteristics.....	14
1.2.4 Advantages of MANETS.....	14
1.2.5 Disadvantages of MANETS.....	15
1.2.6 Infrastructure of ad-hoc networks.....	15
1.2.7 Dynamic topology of ad-hoc networks.....	15
1.2.8 Problems associated with wireless communication.....	15
1.2.9 Implicit trust relationship between neighbors.....	16
1.2.10 MANETS Applications.....	16

Undergraduate Thesis Report

1.3 Trust.....	17
1.3.1 Defining Trust Mechanism.....	18
1.3.2 Conceptual model.....	26
1.3.3 Establishing Trust.....	27
1.3.4 Purpose of a trust model.....	28
1.3.5 Probabilistic models for trust.....	28
1.4 NODES.....	29
1.4.1 Node Client.....	30
1.4.2 Node Specifications.....	31
CHAPTER 2: Literature Survey.....	32
2.1.1 Mitigating Routing Misbehavior in Mobile Ad Hoc Networks.....	32
2.1.2 Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes-fairness In Dynamic Ad Hoc Networks.....	33
2.1.3 CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks.....	33
2.1.4 Trust Computations and Trust Dynamics In Mobile adhoc Networks: Survey.....	34
2.1.5 Dynamic Source Routing in Ad Hoc Wireless Networks.....	35
2.1.6 A Survey of Secure Wireless Ad Hoc Routing.....	36
2.1.7 A Survey on Trust Management for Mobile Ad Hoc Networks.....	37
2.1.8 Secure Routing for Mobile Ad-hoc Networks.....	38
CHPTER 3 : Architecture Overview.....	39
3.1 Confidant.....	39
3.1.1 Architecture.....	39
3.1.2 Description.....	39



Undergraduate Thesis Report

3.1.3 Performance Analysis.....	40
3.2 Watchdog Pathrater .....	41
3.2.1 Architecture.....	41
3.2.2 Description.....	41
3.2.3 Performance Analysis.....	42
3.3 CORE.....	43
3.3.1 Description.....	43
3.3.2 Architecture.....	43
3.3.3 The CORE Scheme.....	44
3.3.4 Components.....	44
3.3.5 Protocol.....	46
3.3.6 Request Made by a Misbehaving Entity.....	47
3.3.7 RT Updates and Distribution.....	47
3.4 COOPERATION ENFORCEMENT.....	48
3.4.1 The CORE Node Operation.....	49
3.4.2. Vulnerabilities of CORE.....	50
3.4.3 Performance Analysis.....	51
CHAPTER 4: Implementation Details.....	52
4.1 Implementation.....	52
4.2 Simulation Setting Files.....	57
4.3 Shell Script File.....	59
4.4 Simulation Settings.....	60
4.5 Simulation Topologies.....	62
4.6 simulation Modeling.....	63
4.7 Model Design and Implementation.....	64
4.7.1 Node Movement.....	65

Undergraduate Thesis Report

4.7.2 Node Transmission Range.....	65
4.7.3 Physical and MAC Layers.....	68
4.7.4 Radio Propagation Model.....	68
4.7.5 Omni-directional Antenna.....	69
4.7.6 Topology and Traffic Settings.....	69
4.7.7 Routing and Transport Protocols.....	69
4.8 Evaluation and Matrices.....	70
CHAPTER 5: Simulation Scenarios.....	71
5.1 Description and Motivation about Scenarios.....	71
CHAPTER 6: Results and discussions.....	73
6.1 Performance Evaluation Results.....	73
6.1.1 Performance Comparison of AODV and OLSR with Varying Network Size.....	73
6.2 Performance Comparison of AODV and OLSR with Varying Traffic.....	78
6.3 Performance Comparison of AODV and OLSR for Varying Mobility.....	82
CHAPTER 7: Conclusion and Future Work.....	87
REFERENCES.....	88

## LIST OF FIGURES

	<b>Page Number</b>
Figure 1.1: Ad-hoc Routing.....	12
Figure 1.2: TRUST in Internet Mechanism.....	17
Figure 1.3: Route Discovery in AODV1.....	25
Figure 1.4: Route Discovery in AODV2.....	25
Figure 1.5: NODES in Mobile Network.....	30
Figure 4.1 : Screenshot of Simulation.tcl 1.....	52
Figure 4.2 :Screenshot of Simulation.tcl 2.....	53
Figure 4.3: Screenshot of Simulation.tcl 3.....	54
Figure 4.4: Screenshot of Simulation.tcl 4.....	55
Figure 4.5: Screenshot of Simulation.tcl 5.....	55
Figure 4.6: Screenshot of Simulation.tcl 6.....	56
Figure 4.7: Screenshot of Simulation.tcl 7.....	56
Figure 4.8: Screenshot of Simulation.tcl 8.....	57
Figure 4.9: Screenshot of Simulation.tcl 9.....	58
Figure 4.10: 25 Node Topology.....	61
Figure 4.11: 50 Node Topology.....	61
Figure 4.12: 75 Node Topology.....	62
Figure 4.13: 100 Node Topology.....	62
Figure 4.14: Data Flow For a Single Simulation.....	63
Figure 4.15: Screenshot of Simulation.tcl 10.....	64
Figure 4.16: 50 Nodes Before Simulation.....	65

Undergraduate Thesis Report

Figure 4.17: 50 Nodes After Simulation.....	66
Figure 4.18: 100 Nodes Before Simulation.....	66
Figure 4.19: 100 Nodes After Simulation.....	67
Figure 4.20: Screenshot of Evaluation Program.....	68
Figure 6.1: Packet Delivery Percentage for AODV and OLSR.....	74
Figure 6.2: Packet Loss Percentage for AODV and OLSR.....	75
Figure 6.3: End-to-End Delay for AODV and OLSR.....	76
Figure 6.4: Routing Overhead for AODV and OLSR.....	77
Figure 6.5: Throughput for AODV and OLSR.....	77
Figure 6.6: Performance Comparison of AODV and OLSR with Varying Traffic.....	78
Figure 6.7: Packet Loss Percentage for AODV and OLSR.....	79
Figure 6.8: Delay AODV and OLSR.....	80
Figure 6.9: Routing Overhead Comparison for AODV and OLSR.....	80
Figure 6.10: Throughput Comparison for AODV and OLSR.....	81
Figure 6.11: Throughput Comparison for AODV and OLSR.....	82
Figure 6.12: End-to-End Delay Comparison for AODV and OLSR.....	83
Figure 6.13: Routing Overhead Comparison for AODV and OLSR.....	84
Figure 6.14: Packet Delivery Comparison for AODV and OLSR.....	85
Figure 6.15: Packet Loss Comparison for AODV and OLSR.....	86

## LIST OF TABLES

### Page Number

Table 1: Simulation Parameters.....	60
-------------------------------------	----

# CHAPTER 1

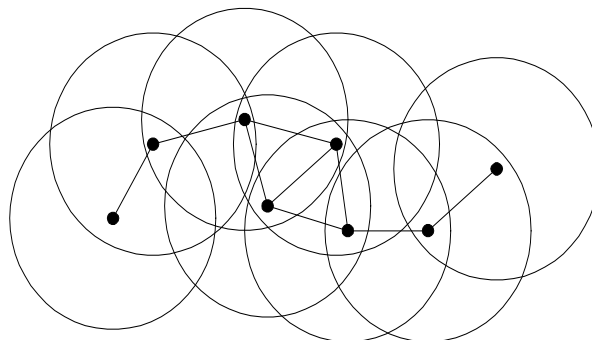
## Introduction

### 1.1 Objective

Mobile ad-hoc networks are very popular type of networks which have a great variety of application in various fields. Due to the versatility of the network configuration topology and the need for faster and secure communication leave it a challenging job to provide reliant and secure routing protocol. So mobile Ad-hoc networks are often more vulnerable than other type of networks .the overall performance of a mobile ad-hoc network depends on the routing protocol. The main objective of this paper is to study two different type of routing protocols: Ad-hoc On Demand Distance Vector (AODV) and Optimized Link-State Routing (OLSR), observing their role and performance in a network and compare their performances. For this purpose, we have performed some computer aided simulation.

### 1.2 MANETs

A mobile ad hoc network (MANET) is generally defined as a network that has many free or autonomous nodes, often composed of mobile devices or other mobile pieces that can arrange themselves in various ways and operate without strict top-down network administration. There are many different types of setups that could be called MANETs and the potential for this sort of network is still being studied.



**Figure 1.1: Ad-hoc Routing**

### 1.2.1 Types of MANETS

- **Vehicular ad hoc Networks (VANETs):** VANETs are used for communication between vehicles and roadside equipment. Intelligent vehicular ad hoc networks (InVANETs) are a kind of artificial intelligence that helps vehicles to behave in intelligent manners during vehicle-to-vehicle collisions, accidents.
- **Smart Phone ad hoc Networks (SPANs):** SPANs leverage the existing hardware (primarily Bluetooth and Wi-Fi) in commercially available smart phones to create peer-to-peer networks without relying on cellular carrier networks, wireless access points, or traditional network infrastructure. SPANs differ from traditional hub and spoke networks, such as Wi-Fi Direct, in that they support multi-hop relays and there is no notion of a group leader so peers can join and leave at will without destroying the network.
- **Internet Based Mobile ad hoc Networks (IMANETS):** IMANETS are ad hoc networks that link mobile nodes and fixed Internet-gateway nodes. For example, multiple sub-MANETs may be connected in a classic Hub-Spoke VPN to create a geographically distributed MANET. In such type of networks normal ad hoc routing algorithms don't apply directly. One implementation of this is Persistent System's CloudRelay.

### 1.2.2 Initial Configuration Mechanisms

In stateful solutions, addresses are assigned by the network; therefore the network should maintain the status information of addresses that have been assigned and/or released.

In stateless solutions, the addresses are assigned by the same node that enters the MANET. This node should run a test for duplicate address detection (DAD) in order to determine the uniqueness of the assigned address.

The hybrid solutions combine aspects of both previous types of solutions to improve the scalability and reliability of auto-configuration mechanisms.

### 1.2.3 MANETs Characteristics

- **Distributed Operation:** There is no background network for the central control of the network operations; the control of the network is distributed among the nodes. The nodes involved in a MANET should cooperate with each other and communicate among themselves and each node acts as a relay as needed, to implement specific functions such as routing and security.
- **Multi-hop Routing:** When a node tries to send information to other nodes which is out of its communication range, the packet should be forwarded via one or more intermediate nodes.
- **Autonomous Terminal:** In MANET, each mobile node is an independent node, which could function as both a host and a router.
- **Dynamic Topology:** Nodes are free to move arbitrarily with different speeds; thus, the network topology may change randomly and at unpredictable time. The nodes in the MANET dynamically establish routing among themselves as they travel around, establishing their own network.
- **Light-Weight Terminals:** In maximum cases, the nodes at MANET are mobile with less CPU capability, low power storage and small memory size.
- **Shared Physical Medium:** The wireless communication medium is accessible to any entity with the appropriate equipment and adequate resources. Accordingly, access to the channel cannot be restricted.

### 1.2.4 Advantages of MANETs

The advantages of an Ad-Hoc network include the following:

- They provide access to information and services regardless of geographic position.
- Independent from central network administration. Self-configuring network, nodes are also act as routers. Less expensive as compared to wired network.
- Scalable—accommodates the addition of more nodes.
- Improved Flexibility.



- Robust due to decentralize administration.
- The network can be set up at any place and time.

### **1.2.5 Disadvantages of MANETs**

In ad-hoc routing protocols, nodes exchange information with each other about the network topology, because the nodes are also routers. This fact is also an important weakness because a compromised node could give bad information to redirect traffic or simply stop it. Moreover, we can say that routing protocols are very brittle in term of security. This part aims to provide a description of the causes of the problems with adhoc routing protocols.

### **1.2.6 Infrastructure of ad-hoc Networks**

Ad-hoc networks have no predetermined fixed infrastructure, that's why the nodes themselves have to deal with the routing of packets. Each node relies on the other neighboring nodes to route packets for them.

### **1.2.7 Dynamic Topology of ad-hoc Networks**

The organization of the nodes may change because of the mobility-aspect of ad-hoc networks: they contain nodes that may frequently change their locations. Because of this fact, we talk about the dynamic topology of these networks, which is a main characteristic that causes problems: when several adhoc networks mix together, there can be duplications of IP addresses, and resolving it is not so simple. Then, attacks can easily occur by using this duplication of IP address (cf. attacks using impersonation).

### **1.2.8 Problems Associated With Wireless Communication**

Wireless channels have a poor protection to noise and signal interferences, therefore routing related control messages can be tampered. A malicious intruder can just spy on the line, jam, interrupt or distort the information circulating within this network.

### 1.2.9 Implicit Trust Relationship between Neighbors

Actual ad-hoc routing protocols suppose that all participants are honest. Then, this directly allows malicious nodes to operate and try to paralyze the whole network, just by providing wrong information.

### 1.2.10 MANETs Applications

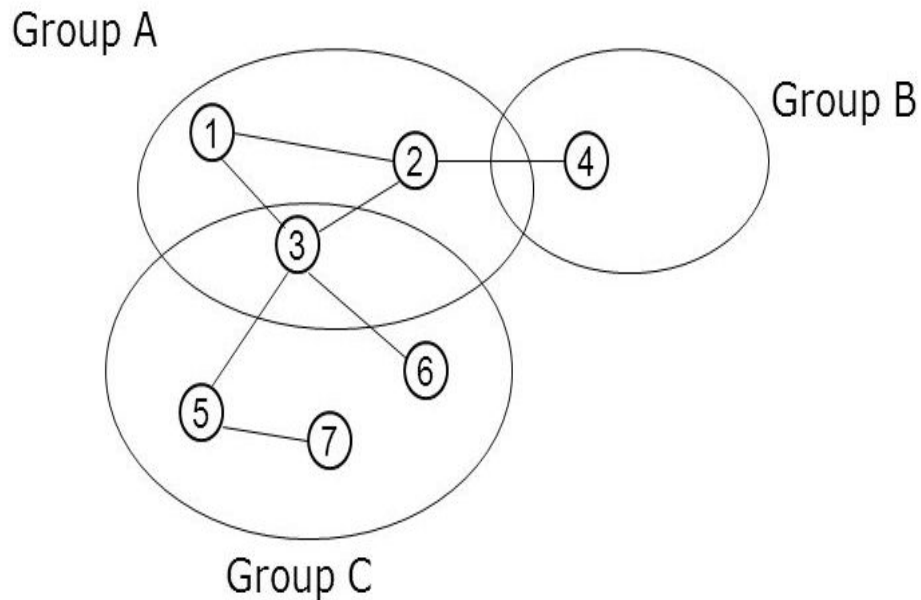
Some of the typical applications include:

- **Military Battlefield:** Ad-Hoc networking would allow the military to take advantage of commonplace network technology to maintain an information network between the soldiers, vehicles, and military information head quarter.
- **Collaborative Work:** For some business environments, the need for collaborative computing might be more important outside office environments than inside and where people do need to have outside meetings to cooperate and exchange information on a given project.
- **Local Level:** Ad-Hoc networks can autonomously link an instant and temporary multimedia network using notebook computers to spread and share information among participants at a e.g. conference or classroom. Another appropriate local level application might be in home networks where devices can communicate directly to exchange information.
- **Personal Area Network and Bluetooth :** A personal area network is a short range, localized network where nodes are usually associated with a given person. Short-range MANET such as Bluetooth can simplify the inter communication between various mobile devices such as a laptop, and a mobile phone.
- **Commercial Sector:** Ad hoc can be used in emergency/rescue operations for disaster relief efforts, e.g. in fire, flood, or earthquake. Emergency rescue operations must take place where non-existing or damaged communications infrastructure and rapid deployment of a communication network is needed.

### 1.3 Trust

Every security system depends on trust, in one form or another, among users of the system. In general, different forms of trust exist to address different types of problems and mitigate risk in certain conditions. Which form of trust to apply in a given circumstance is generally dictated by corporate policy.

A trusted Internet takes into account security, transaction protection, and identity assertion and management. Given the network dependence on unique numbers and the escalating amount of geo location data being gathered, the privacy implications of the current Internet represent a significant and growing concern.



**Figure 1.2: TRUST in Internet Mechanism**

### **1.3.1 Defining Trust Mechanism**

Trust is defined as a binary relationship, or set of compounded binary relationships, based on individual identity or unique characteristic validation. That is, trust is the establishment of a trust relationship through a validation process and the subsequent use of that relationship in some transactional context.

A mobile ad-hoc network is a type of network with high mobility. It has no particular infrastructure and central administration. This type of network is widely used in military services and emergency civil communication services. Since there is no definite infrastructure and that any node can join and leave the network at any time, providing security to this system is a little bit challenging. That is why MANET systems are vulnerable to malicious attacks.

Routing protocols can be classified into 3 types:

- 1) Reactive protocol,
- 2) Proactive protocol and
- 3) Hybrid protocol.

Reactive protocols among all are more efficient because of low computation costs with no additional requirement to exchange routing information to maintain route tables. Some of the routing protocols under this concept are DSR, TORA and AODV. AODV has better experience than others. Security is provided to AODV routing system in two ways. They are the cryptographic mechanism and the trust based mechanism. Cryptographic mechanism uses encryption method, public key method or another's cryptography method. Trust mechanism calculates the trust level of each node before establishing communication. Compared to cryptographic method, trust based method is more effective, since cryptographic method include some disadvantages like network overhead due to additional information exchanged.

The trust based method that has been proposed here is called AODV. Here the trust level is calculated based on the successful and failed communication. Here probability approach is used to compute the trust opinion. For example, node A needs to assess trust level of node B using the following set of equations:

$$b_B^A = \frac{p}{p+n+2} \dots\dots\dots(1.1)$$

$$d_B^A = \frac{n}{p+n+2} \dots\dots\dots(1.2)$$

$$u_B^A = \frac{2}{p+n+2} \dots\dots\dots(1.3)$$

Where  $b_B^A = \frac{p}{p+n+2}$  is the probability that a node B can be trusted by a node A,  $d_B^A = \frac{n}{p+n+2}$  is the probability that a node B cannot be trusted by a node A,  $u_B^A = \frac{2}{p+n+2}$  is the uncertainty of both belief and disbelief, p and n are the positive and negative events respectively. Belief, disbelief and uncertainty are calculated using probabilistic approach based on the successful and unsuccessful packet sending between nodes. The mechanism needs to perform three steps of computation before sending the packets i.e. trust calculation procedure, trust combination procedure and trust judging procedure. The mechanism cannot detect the attack during the route discovery process. The nodes that have direct connection between each other are more trusted to calculate the trust level of its neighbors than node that does not have a direct connection. Equation 1.4 shows the calculation of trust accumulation opinions.

$$T_{VBOi} = \sum_{i=0}^N DTVOi (90\% \text{ of } T_{VBOi}) + \sum_{i=0}^N ITVOi (10\% \text{ of } T_{VBOi}) \dots\dots\dots(1.4)$$

Where  $DTVOi$  is direct trust opinion of other nodes which are the direct neighbors of the Guard node A.  $ITVOi$  is indirect trust opinion of other nodes about a specific node. In another proposal, trust opinion was calculated based on the connection behaviors among the nodes. The trust proportions is represented as  $\omega$ , where the value are  $0 < \omega < 1$ . Total trust opinion is calculated with Equation 1.5

$$T = \omega Td + (1 - \omega), 0 < \omega < 1 \dots\dots\dots(1.5)$$

Where  $T$  is the total trust value for a particular collaborator,  $Td$  is the direct trust value,  $Tid$  is the indirect trust value, and  $\omega$  represents the importance proportion of direct trust to the total trust.

Another proposal approached the route trust calculation by detecting the success level of the packet arrives in the destination. Trust node and trust route are combined to choose the secure path to destination. The route trust is calculated using Equation 1.6

$$\text{Route Trust} = (\text{No. of Packets Sent by the Node} - \text{No. of Packets Received by Destination}) \dots \dots \dots (1.6)$$

The perfect condition is when the route trust equal 0. Route is trusted if the differences between the sent packet and received packet is small and almost zero. To exchange the neighbor list and route trust value, these information are put in the RREP packet. That makes the packet size of RREP increases.

Trust calculation based on the level of successful packet exchanges is also used to compute the trust level among the nodes.

The success ratio of the trust level is calculated with Equation 1.7 and Equation 1.8.

$$Rr = Rrs - Rrf \text{ where } Rrs + Rrf \neq 0, \text{ otherwise } Rr = 0 \dots \dots \dots (1.7)$$

$$Rf = Rfs - Rff \text{ where } Rfs + Rff \neq 0, \text{ otherwise } Rf = 0 \dots \dots \dots (1.8)$$

Where  $Rr$  is the packet routing credence,  $Rrf$  is the number of routing packets that are failing to forward and  $Rrs$  is the number of routing packets that are forwarded successfully.  $Rf$  is the value of forwarding credence category,  $Rfs$  is a number of data packets that are forwarded successfully, and  $Rff$  is a number of data packets that are failing to forward.

Another proposal proposed Trust Cross Layer Secure protocol (TCLS) routing protocol. Security mechanisms in TCLS also uses packet routing success ratio as a trust parameter. But in the success ratio is calculated based on the total RREQ arriving at the destination node, not the total RREQ between the neighbor nodes. Trust success ratio is calculated with Equation 1.9.

$$SRI = \frac{FC}{ni} \cdot Prec \dots \dots \dots (1.9)$$

Where  $SRI$  is a success ratio value and  $Prec$  is the number of packets received at destination node in specific time interval. Success ratio value will be added on the RREP packet and it is broadcasted to the next neighbor nodes. It is encrypted using cryptography method before sending to the source node. If the intermediate node is failed to verify the digital signature of the destination node then the RREP packet is dropped. The trust values of the node will be increased if the node has a high success ratio value and the packet can be verified by the intermediate nodes. The authentication and encryption process are performed use CBC-X encryption method.

Based on the literature study about the trust mechanism for securing the routing protocol, success ratio becomes an important parameter to calculate the trust level of the nodes. The aim of the trust calculation is to detect the potential attack and mitigate the attacker to avoid its impact to the network. The trust calculation can only perform after communication is established, if the packet data is used as a parameter. The attack cannot be detected by the trust mechanism if it is perform during the route discovery phases, because the nodes only calculate the success ratio of packet data. If the packet routing is used as a parameter to calculate the trust level, the trust mechanism directly starts the detection when the node performs the route discovery phases. This allows the trust mechanism to mitigate the attacker before the communication is established. In our trust calculation, we use routing packets as parameters to calculate the trust level of each node. In the Equation 1.7, the success ratio is the comparison of the difference between success packet and failed packet to the accumulation of success packet and failed packet. In this approach, we cannot detect the detailed behaviors of the each node. If the node is a malicious node, there is a possibility that the malicious nodes only sends or forwards some packets, not all the packets. However, in the Equation 1.9 the trust level of each node is calculated based on the comparison between total RREQ packet arrives in the destination node to the total of packets that have been forwarded by the each node. This approach only uses the total number of RREQ

packet that arrives in the destination. Each time the intermediate node forwards the routing packet, it will duplicate the routing packets based on the number of its neighbors. The total number of RREQ packet forwarded should be bigger than the total accepted RREQ in that node. With this approach, we assume that the total RREQ in the destination cannot be a parameter to calculate the trust level of each node in the network. Our proposed trust calculation computes the node trust level based on the behaviors and activities of each node. The assumptions about the normal activities are:

- a. The node is a normal node if it forwards all the routing packet to its neighbors. Based on this assumption, the total number of packet sending must be equal or more than the total packet receives at the nodes. The total forwarded RREQ depends on the total neighbors of that's node.
- b. If the direct neighbor nodes do not receives the packet that have been forwarded by its neighbors, then this nodes is suspected as a malicious nodes.

Based on these assumptions, the trust behaviors calculation is divided into two kinds of trust i.e. trust local calculation (TL) and trust global calculations (TG). The definition of trust local and trust global as follows:

- a. Trust global (TG) is the trust level calculation based on the total activities of the nodes. The activities are the total number of received routing packets and the total number of sending routing packets.
- b. Trust local (TL) is the node trust calculation based on the total number of routing packets that have been received from a specific node and forward it to its self. 50 Each node in the network will calculate the trust local and trust global of its neighbors. The node must accumulate TL and TG values to compute the total trust level of its neighbor nodes before sending or forwarding the packets. Equation 1.10 is utilized to calculate the trust local, and Equation 1.11 is utilized to calculate the trust global. In these Equations, the node i want to calculate the trust level of node j.

$$TL_{i,j} = \sum Pr_{i,j} \sum Pr_{ifj,k} ;$$

where  $\sum Pr_{ifj} \neq 0$  .....(1.10)

$$TG_i = \sum Pr_j \sum ; here \sum Ps_j \neq 0$$
 .....(1.11)



Where  $TL_i$ , is the trust local opinion of node  $i$  to node  $j$ ,  $TG_i$ , is the trust global opinion of node  $i$  to node  $j$ ,  $Pr$  is the received routing packet,  $Ps$  is the sent routing packets and  $Prifj$ , is the total forwarded routing packet from node  $i$  by the  $j$  that origin from node  $k$ . Trust local (TL) is the comparison of packet routing from the specific nodes. It assesses the specific behaviors of each node. In AODV, the identical routing packet is received only once by the nodes. Because each time node receives the routing packet, the packet id will be checked. If the packet has been received before, then the latest one will be ignored. Based on this assumption, the node is a normal node if the trust local calculation is equal to 1. Otherwise, the node is suspected as a malicious node. If the node is a trusted node, then the TL value is set 1. Otherwise, the TL value is set to 0. Trust local opinion is set by using Equation 1.12 and Equation 1.13.

$$TL_i = 1, \text{henodeistrustedandTLvalueisset } 1 \dots \dots \dots (1.12)$$

$$TL_i \neq 1, \text{henodeisuntrustedandTLvalueisset } 1 \text{ set } 0 \dots \dots \dots (1.13)$$

Trust global (TG) is the comparison between total routing packets that have been received and total routing packet that have been forwarded by the node. This indicates the global behaviors of the nodes. In the AODV protocol, routing packet will be forwarded if the intermediate node is not a destination node. The intermediate node forwards the routing packet to all its neighbors. Based on this condition, the total number of forwarded routing packet by the node is greater than the total of routing packet that has been received. Therefore, in the trust global view, the node is a normal node if the trust global calculation equal or less than 1. Otherwise, the node is suspected as a malicious node. If the node is a trusted node, then the TG value is set 1. Otherwise, the TG value is set to 0. Equation 1.14 and Equation 1.15 show the opinion of trust global calculation.

$$TG_i \leq 1, \text{henodeistrustedandTGvalueisset } 1 \dots \dots \dots (1.14)$$

$$TG_i > 1, \text{henodeisuntrustedandTGvalueisset } 0 \dots \dots \dots (1.15)$$

Nodes conclude the total trust level of its neighbor by accumulating the trust local and trust global values. The node is marked as a trusted node when both result opinions accumulation of TL and TG is trusted. If one of the trust opinions is un-trusted, the node is suspected as a

malicious node. Based on this assumption, the AND logic is utilized to accumulate the trust opinion values. Equation 16 shows the accumulation model.

$$Totaltrustlevel_{i,j} = TL_{i,j} \wedge TG_{i,j} \dots\dots\dots(1.16)$$

Trust mechanism calculation using TL and TG method can be performed only if all the nodes in the network have the ability to hear all the activities of its neighbors. To fulfill this condition, the network must be in the promiscuous mode.

### **A. Destination Sequence Number (DSQ) value control mechanism**

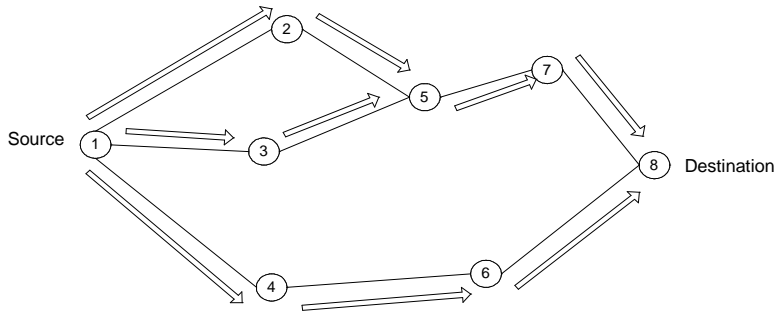
Each node monitors the DSQ value of RREP by calculating the difference in the routing table. When the node sends or forwards the RREQ packets, it records the destination address and the DSQ value in its routing table. When the node receives the RREP packets, it checks the routing table if there is a same destination address. If it does exist, the difference of DSQ is calculated. Otherwise, it forwards the RREP packets. The origin node of RREP is suspected as a malicious node if the DSQ difference value is more than threshold.

### **B. Route discovery phases**

The initial condition of the all node in the network is considered as a trusted node. The default TL and TG values are 1. The source node broadcasts RREQ packet to all neighborhood for finding the communication route to the destination node. In the first time, source node found that all its neighbors are trusted nodes. Therefore, it sends 102 the routing packet directly. When the intermediate node received the RREQ packet, it checks the trust level by calculating the TL and TG of the source node. If it is an un-trusted node, then the RREQ is ignored. Otherwise, the intermediate node calculates the trust level of its next neighbor nodes and forwards the packet routing only to the trusted neighbor nodes. Trust calculation mechanism is performed in two sides i.e. at the sender node and receiver node of packet routing. Once the destination node receives RREQ packet, it generates and broadcasts the RREP packet to the source node through the reverse route that have been created by RREQ packet. During sending the RREP packet, the node does not need to recalculate the trust level of each node in its reverse path because it has been done when RREQ find the path to destination. When the intermediate node receives RREP, it compares the DSQ value by performing the DSQ value control mechanism. When the source

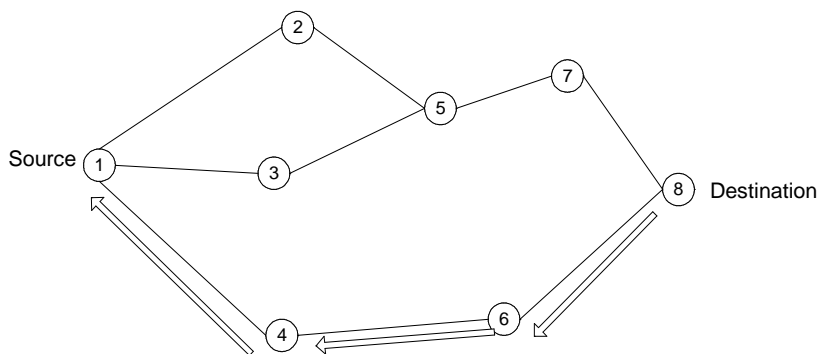
node receives RREP packet, it selects the route from the RREP with a normal DSQ value and the minimum number of hops. Figure 1.3 & 1.4 explains the route discovery procedures.

- **Propagation of Route Request (RREQ) packet**



**Figure 1.3: Route Discovery in AODV1**

- **Path Taken by Route Reply(RREP) Packet**



**Figure 1.4: Route Discovery in AODV2**

Source node broadcasts PREQ to all trusted neighbor nodes. Initial condition for all node is trusted (TL =1 and TG=1)

Node received PREQ, it calculates TL and TG of the previous node.

- a. If the previous node is untrusted, PREQ is ignored.
- b. If the previous node is trusted, nodes creates a reverse route to the origin of the packet.
- c. Node calculates the TL and TG of the next neighbor nodes.
- d. Node forwards the PREQ packet only to the trusted neighbor nodes.

Destination node receives PREQ packet, It generates and send PREP to source node though the reverse route.

When the intermediate node receives PREP, it compares the DSQ values by performing the DSQ value control mechanism.

Once the source node receives the PREP, It selects the communication route bases on the normal DSQ value and the minimum number of hops.

### **C. Route Maintenance Phases**

When there is a broken link during the communication process, the nearest node to the broken link generates and sends the RERR messages to the source node. Once the source node receives the RERR messages, it re-initiates the route discovery phases if the communication is still needed.

## **1.3.2 Conceptual Model**

Trust and reputation model can be characterized as:

- **Cognitive**

In models based on a cognitive approach, Trust and reputation are made up of underlying beliefs and are a function of the degree of these beliefs. The mental states, that lead to trust another agent or to assign a reputation, are an essential part of the model, as well as the mental consequences of the decision and the act of relying on another agent.

- **Game-Theoretical**

Trust and reputation are considered subjective probabilities by which the individual A, expects the individual B to perform a given action on which its welfare depends. In this approach, trust and reputation are not the result of a mental state of the agent in a cognitive sense, but the result of a more pragmatic game with utility functions and numerical aggregation of past interactions.

### **1.3.3 Establishing Trust**

To establish trust or confidence, there must be a binding of unique attributes to a unique identity, and the binding must be able to be tested satisfactorily by a relying entity. When one achieve a satisfactory level of confidence in the attributes provided by an entity, one establish a trust relationship. This element of trust is commonly called authentication.

Trust involves a binary relationship, or a set of compounded binary relationships based on validation of unique individual identity. Consider the following examples of simple trust models:

- A trusts B. (This means A can validate the unique identity of B. It does not mean that B necessarily trusts A.)
- A trusts B, and B trusts A.
- A trusts B, B trusts C, therefore, A trusts C.

A trust model is not the particular security mechanisms utilized within a particular security architecture. Rather, it is the combination of those security mechanisms in conjunction with the security policy when they address all business, technical legal, regulatory, or fiduciary requirements to the satisfaction of a relying entity.

### **1.3.4 Purpose of a Trust Model**

The purpose of a trust model is to respond to a specific threat profile. A threat profile is the set of threats and vulnerabilities identified through a use-case-driven data flow analysis that is particular to an organization. Essentially, a threat profile identifies likely attackers and what they want.

The level of trust necessary for one organization or circumstance may be different from the level of trust required by another organization or circumstance. For example, the level of assurance that an organization needs regarding the authentication of a user may be different in particular use cases.

### **1.3.5 Probabilistic Models for Trust**

The credential based trust has a binary nature. That is a principal is either trusted or untrusted to perform an action. Credentials are issued, as proofs of trust, to a principal if its behavior is well known, by the issuing authorities, to comply with specific security obligations. By exchanging credentials between principals, they can base their mutual interactions on ‘proved’ knowledge about each other’s behavior. This trust approach is appropriate in closed networks, where principals have sufficient information (through credentials) about their peers. Nevertheless, in modern, large-scale networks (e.g. the Internet), interacting principals can have autonomously different behaviors and intentions which are incompletely known by each other. This incomplete knowledge available to principals about each other makes credentials not the appropriate evidence of trust because no principal is ‘perfectly’ trusted, that is guaranteed to conform to interaction-related policies. However, the trustworthiness of a principal can be reflected by the history of its interactions with other principals (interaction history). Based on this idea, an approach of trust has evolved where a principal (truster) ‘X’ evaluates a quantitative measure for its trust in another principal (trustee) ‘Y’ using ‘Y’s interaction history. Note that the trust value here is not binary as the case in credential-based trust, but rather a number expressing the level of trustworthiness. This view of trust is known as the computational trust and also as reputation based trust.

## **1.4 Nodes**

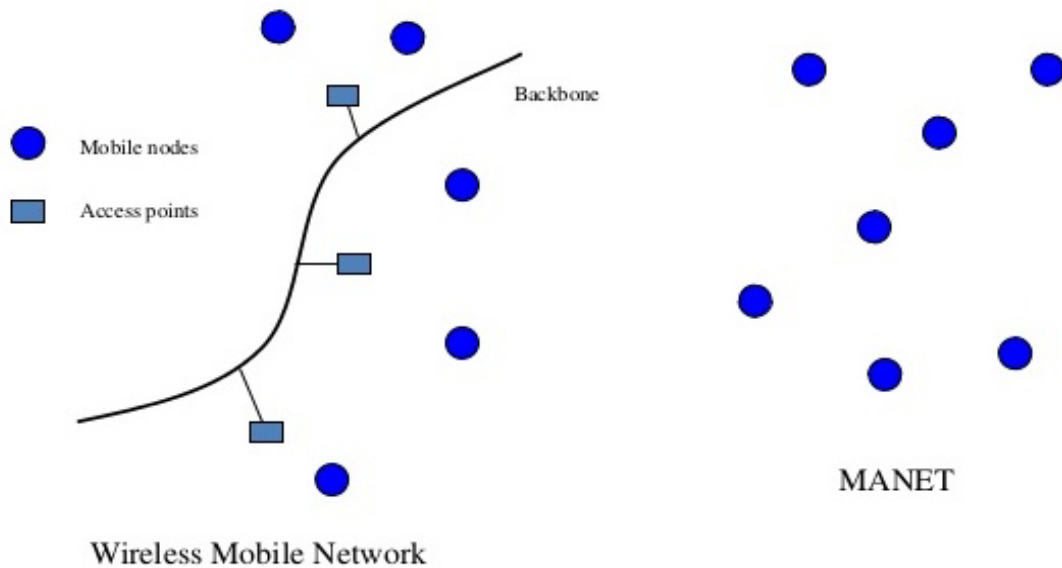
In communication networks, a node (Latin nodus, 'knot') is either a connection point, a redistribution point or a communication endpoint (some terminal equipment). The definition of a node depends on the network and protocol layer.

In a communications network, a network node is a connection point that can receive, create, store or send data along distributed network routes. Each network node - whether it's an endpoint for data transmissions or a redistribution point has either a programmed or engineered capability to recognize, process and forward transmissions to other network nodes.

The concept of network nodes came into being with the use of distributed networks and packet switching. Depending on its application, network nodes perform a variety of functions. In data communications, physical network nodes include data communications equipment or devices that sit between data terminal equipment (DTE) and data transmission circuits.

These include switches, bridges, modems or hubs that perform signal conversion, coding and line clocking. Network nodes in data communications also include data terminal equipment like digital telephone handsets, printers or host computers like routers, servers or workstations.

In internet and intranet networks, most physical network nodes are host computers that are identified by an IP address. Some data link devices like wireless local area network (WLAN) access points do not have IP host addresses and are considered physical network or LAN nodes rather than internet nodes or hosts.



**Figure 1.5: NODES in Mobile Network**

### 1.4.1 Node Client

Some Partners may want to consider using a Node Client as an alternative to a full Node. Like Nodes, Node Clients can submit, request, and receive results from a request on the Network. Network Clients, however, cannot respond to data queries from other Nodes, and therefore cannot publish data on the Exchange Network. Network Clients may be a more cost-effective solution for Partners who do not have a compelling business need to publish their data on the Exchange Network. Review Node or Node Client: To Be or Not To be to decide which may be the right solution right for your agency.



### **1.4.2 Node Specifications**

In order for Nodes to be able to communicate with one another, they must all comply with the same Node Functional Specifications, a document which outlines the minimum specifications and functionality of an Exchange Network Node. The current Node Specifications is Version 2.1

## CHAPTER 2

### Literature Survey

#### 2.1.1 Mitigating Routing Misbehavior in Mobile Ad Hoc Networks [1]

This paper describes two techniques that improve throughput in an ad hoc network in the presence of nodes that agree to forward packets but fail to do so. To mitigate this problem, categorizing nodes based upon their dynamically measured behavior is proposed. Here two software based simulation is done. **Watchdog** that identifies misbehaving nodes and a **pathrater** that helps routing protocols avoid these nodes. Percentage of overhead transmissions and the accuracy of misbehaving node detection was evaluated using simulation in **watchdog** and **pathrater** packets throughput. When used together in network with moderate mobility, the two techniques increase throughput by 17% in presence of 40% misbehaving nodes, while increasing the overhead transmissions from the standard routing protocol's 9% to 17%. During extreme mobility, **watchdog** and **pathrater** can increase network throughput by 27%, while increasing the overhead transmissions from the standard routing protocol's 12% to 24%. The results show that, the benefits of an increased number of routing nodes can be gained, while minimizing the effects of misbehaving nodes. In addition, the paper proposed that, this can be done without a priori trust or excessive overhead.

### **2.1.2 Performance Analysis of the CONFIDANT Protocol[2]**

The paper describes that Mobile ad-hoc networking works properly only if the participating nodes cooperate in routing and forwarding. However, it may be advantageous for individual nodes not to cooperate. A protocol is proposed, called CONFIDANT, for making misbehavior unattractive. The CONFIDANT protocol works as an extension to a reactive source-routing protocol for mobile ad-hoc networks. It is based on selective altruism and utilitarianism. It aims at detecting and isolating misbehaving nodes, thus making it unattractive to deny cooperation. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. The detailed implementation of CONFIDANT assumes that the network layer is based on the Dynamic Source Routing (DSR) protocol. We present a performance analysis of DSR fortified by CONFIDANT and compare it to regular defenseless DSR. It shows that a network with CONFIDANT and up to 60% of misbehaving nodes behaves almost as well as a benign network, in sharp contrast to a defenseless network.

### **2.1.3 CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks [3]**

The paper describes that a collaborative reputation mechanism named CORE is proposed to enforce cooperation among the nodes of a MANET to forestall ungenerous behavior which is the primary focus of this paper. In MANET countermeasure for misbehavior of node and selfishness are a common part. Lack of node activity that is caused by ungenerous behavior cannot be solved by classical security method which aims at verifying the integrity and correctness of an operation. In this mechanism a technique called reputation is used for keeping the tracks of network collaboration. Here each network entity in CORE keeps track of other entities' collaboration by using the reputation technique. The reputation mechanism is computed based on data monitored by the local entity and some information provided by other nodes involved in each operation. Since there's no incentive for a node to maliciously unfold negative info concerning different nodes, denial of service attacks supported malicious broadcasting of negative ratings for legitimate nodes are strongly prevented. Here Qual Net network simulator software is used to represent the simulation result of CORE reputation mechanism. In this research paper researchers are mainly focused on the lack of anteriority trust bonding between

mobile nodes. Prevention of service attacks and countermeasures against immoral behavior of nodes is the very first concern of researcher. A collaborative reputation mechanism named CORE is suggested that can be integrated with any network function like packet forwarding, route discovery, network management, and location management. Collaborative reputation mechanism is used as a basis for the security mechanism that solves the problems due to misbehaving nodes by incorporating a reputation mechanism that provides an automatic method for the social mechanisms of reputation. This mechanism can be smoothly extended to basic network functions with little impact on existing protocols.

#### **2.1.4 Trust Computations and Trust Dynamics In Mobile adhoc Networks: Survey[4]**

The paper describes about Trust play a key role in building the information security. For nodes participated in MANET, they must have confidence that their neighbor nodes are trustworthy and secure. Trust establishment in MANET is an uncovered and challenging field. The behavior of MANET is based on trust neighbor relationships. These relationships initiate, develop and terminate dynamically and have usually short life spans. Due to computational complexity trust management and computations are highly challenging issue in MANET. The independent movement of component nodes and computational complexity of trust prevents the technical application suited to alternative networks. A malicious node in MANET can cause damage and affect the data quality. The main focus of this paper is represented the detailed survey on varied trust computing approaches that area unit engaged towards MANETs. In this paper researchers also focus on analyzing trust dynamics including trust propagation, prediction and aggregation algorithms, the influence of network dynamics on trust dynamics and the impact of trust on security services. The aim of the paper is to supply MANETs designers with multiple views on the idea of trust, an understanding of the properties that ought to be thought of developing a trust metric, and insights on how trust is computed. Trust scheme presented in this paper are based on many different types of mechanism. The aggregation of trust is useful in a computational model but it can increase its complexity making a general solution difficult. Several models are dependent on the characteristics of the environment and a possible solution could be the use of adaptive mechanisms that can modify how to combine different sources of information in a given environment. A lot of trust and reputation definitions have been presented and there are several

works that give meaning to both concepts. It is necessary to contemplate the constraints and also the sort of information which will be used as input by the network while coming up with a brand new trust system. A wide range of application is covered in this paper depending on various types of mechanism that helps to represent the trust scheme.

### **2.1.5 Dynamic Source Routing in Ad Hoc Wireless Networks[5]**

The paper describes that Ad hoc networks are a new wireless networking paradigm for mobile hosts. Unlike traditional mobile wireless networks, ad hoc networks do not rely on any fixed infrastructure. Instead, hosts rely on each other to keep the network connected. To forward a packet to its destination, it might be necessary for one mobile host to enlist the help of alternative hosts because of the range limitation of each mobile host's wireless transmissions. This paper mainly concerns about dynamic source routing, a protocol for routing in mobile ad hoc networks. The protocol adapts quickly to routing changes once host movement is frequent, nevertheless needs very little or no overhead in periods during which hosts move less of times. The paper describes the design and performance of a routing protocol for ad hoc networks that instead uses dynamic source routing of packets between hosts that want to communicate. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which to forward the packet; the sender explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination host. To evaluate the performance of dynamic source routing protocol a packet level simulator is used. In this paper a protocol is proposed for routing packets between wireless mobile hosts in an ad hoc network. Like other protocols such as distance vector or link state algorithms, this protocol uses dynamic source routing that adapts quickly to routing changes once host movement is frequent, nonetheless needs very little or no overhead during times within which hosts move less often. Here researchers assumed that all hosts want to communicate with alternative hosts within the ad hoc network are willing to participate absolutely within the protocols of the network. Particularly, every host collaborating within the network ought to even be willing to forward packets for alternative hosts within the network. Researchers strongly believe that this dynamic source routing protocol offers a number of potential advantages over conventional routing protocols such as distance vector in an ad hoc network.

## 2.1.6 A Survey of Secure Wireless Ad Hoc Routing[6]

According to the paper Ad hoc networks use mobile nodes to enable communication outside wireless transmission range. Attacks on ad hoc network routing protocols disrupt network performance and reliability.

In a multi-hop wireless ad hoc network, mobile nodes cooperate to form a network without using any infrastructure such as access points or base stations. Instead, the mobile nodes forward packets for each other, allowing communication among nodes outside wireless transmission range. The nodes' mobility and the fundamentally limited capacity of the wireless medium, together with wireless transmission effects such as attenuation, multipath propagation, and interference, combine to create significant challenges for routing protocols operating in an ad hoc network.

**Routing-disruption attacks:** The attacker attempts to cause legitimate data packets to be routed in dysfunctional ways.

**Resource-consumption attacks:** The attacker injects packets into the network in an attempt to consume valuable network resources such as bandwidth or to consume.

From an application-layer perspective, both attacks are instances of a denial-of-service (DoS) attack. An attacker might similarly create a routing black hole, which attracts and drops data packets. In a special case of a black hole, an attacker could create a gray hole, in which it selectively drops some packets but not others. An attacker might attempt to make a route through itself appear longer by adding virtual nodes to the route; we call this attack a gratuitous detour because a shorter route exists and would otherwise have been used. A more subtle type of routing-disruption attack is creating a wormhole in the network, using a pair of attacker Nodes A and B linked via a private network connection. Arushing attack is a malicious attack that is targeted against on-demand routing protocols that use duplicate suppression at each node. Establishment of Private Key and Public key distribution. Private-key distribution is more challenging than public-key distribution because protocols for key distribution must ensure the secrecy of such keys.

### **2.1.7 A Survey on Trust Management for Mobile Ad Hoc Networks[7]**

According to the paper managing trust in a distributed Mobile Ad Hoc Network (MANET) is challenging when collaboration or cooperation is critical to achieving mission and system goals such as reliability, availability, scalability, and re-configurability. In defining and managing trust in a military MANET, the interactions between the composite cognitive, social, information and communication networks is considered, and take into account the severe resource constraints (e.g., computing power, energy, bandwidth, time), and dynamics (e.g., topology changes, node mobility, node failure, propagation channel conditions). To combine the notions of “social trust” derived from social networks with “quality-of-service (QoS) trust” derived from information and communication networks to obtain a composite trust metric. A survey of trust management schemes developed for MANETs and discussed generally accepted classifications, potential attacks, performance metrics, and trust metrics in MANETs. The future research areas on trust management in MANETs based on the concept of social and cognitive networks. A composite trust metric that captures aspects of communications and social networks, and corresponding trust measurement, trust distribution, and trust management schemes are interesting research directions. For dynamic networks, such as military MANETs, these schemes should have desirable attributes such as ability to adapt to environmental dynamics, scalability, reliability, and re- configurability.

### **2.1.8 Secure Routing for Mobile Ad-hoc Networks[8]**

The paper describes about mobile Ad Hoc Networks (MANETs) are an emerging type of wireless networking, in which mobile nodes associate on an extemporaneous or ad hoc basis. It has become particularly vulnerable to intrusion, as they operate in open medium, and use cooperative strategies for network communications. The widely accepted existing routing protocols designed to accommodate the needs of such self-organized networks do not address possible threats aiming at the disruption of the protocol itself. In this paper researchers want to show most popular protocol that follow the table-driven and the source-initiated on-demand approaches. Researches are successful to formulate the threat model for ad hoc routing and present several specific attacks that can target the operation of a protocol. This security protocol classified into five categories: solutions based on asymmetric cryptography; solutions based on

symmetric cryptography; hybrid solutions; reputation-based solutions; and a category of add-on mechanisms that satisfy specific security requirements. In this paper add-on mechanisms that address specific security problems in ad hoc routing or techniques and extensions to existing approaches are present. The best part of the paper is has presented the best known protocols for securing the routing function in mobile ad hoc networks. The analysis of the different proposals has demonstrated that inherent characteristics of ad hoc networks, such as lack of infrastructure and rapidly changing topologies, introduce additional difficulties to the already complicated problem of secure routing.



## CHAPTER 3

### Architecture Overview

#### 3.1 Confidant

One of the most well-known reputation mechanisms for ad hoc networks is the CONFIDANT protocol. It is a reputation based dynamic and weighted transitive trust management system based on DSR protocol

##### 3.1.1 Architecture

CONFIDANT consists of four major components:

- **Monitor:** Observes behavior of the neighboring nodes by observing transmission and identifies misbehavior.
- **Reputation manager:** This component maintains a table that has rating for nodes which is updated as per nodes own experience and reported experience.
- **Path manager:** it deals with path re-ranking, path detection, action on malicious node request and action on request for a route containing malicious node.
- **Trust manager:** It deals with trust table management, trust level calculations alarms generated by nodes on observation of a malicious activity.

##### 3.1.2 Description

. Trust in this mechanism is established through direct and indirect observations. It uses first hand and second hand reputation information to detect malicious nodes and isolate them. This is the closest work to ours. Nodes deploys monitors that observe their neighbors and detect any possible misbehavior. Like in the watchdog approach, a path manager helps nodes to choose paths depending on the reputation of nodes in these paths. Moreover, two other components, the trust manager and the reputation system are added to manage reputation of nodes. The first one is in charge of the alarm messages that are used by a node to signal misbehaving nodes to its “friends”, to decide the validity of such messages and to manage the list of friends. The former is

used to rate nodes depending on their behavior. In fact this protocol starts with the watchdog approach and enriches it with reputation information to avoid misbehaving nodes in network functionalities. Simulations show a throughput of good nodes equivalent to the one of a well-behaving DSR network even in the presence of one third of malicious nodes in the population. The protocol assumes that nodes can be identified (i.e. has a unique identity) and that it is possible to manage groups of “friends” in an ad hoc network. The difficulty when using these kinds of solutions is the propagation of reputation and the impact of false reputation. In [1], authors propose to use Bayesian statistics for exclusion of liars to improve the CONFIDANT protocol.

When a malicious activity is observed by any node, this suspicious event is detected by monitor and reputation system is called in turn. Reputation system checks the significance of the event and number of occurrences of events and update rating of nodes accordingly. In case of intolerable rating, path manager is called for deletion of all routes entries containing this malicious node and alarm is sent to monitor. Monitor passes this alarm to trust manager and it evaluates trust of the node due to which the alarm has been generated. If the source of the alarm is trusted one, the alarm table is updated. In case, the source of alarm is malicious, the reputation system is called which again evaluate the alarm.

### **3.1.3 Performance Analysis**

- Throughput increase because of decrease in number of drop of packets.
- Overhead due to alarms increases if the number of malicious node increases.
- Malicious behavior is an exception and false praise attack is not possible because of sharing negative information.
- A malicious node when see negative information about itself can change its strategy and node of good reputation may stop sharing negative information because of the fear of revenge.
- Malicious node that is excluded from the network may reenter the network after timeout.
- CONFIDANT treats faulty and malicious node in same way.
- This scheme not only detects the misbehaving nodes but also refrain malicious nodes from getting benefits from other cooperating nodes.

## 3.2 Watchdog Pathrater

Watchdog Pathrater is a dynamic trust management scheme which is an extension of DSR protocol.

### 3.2.1 Architecture

Watchdog pathrater consists of two components:

- **Watchdog:** in promiscuous node, it listens and monitors that the next node forward packets.
- **Pathrater:** Pathrater is used to delete the misbehaving nodes, to create new paths, avoid uses of misbehaving nodes and select a reliable path for data delivery.

### 3.2.2 Description

WD runs on every node in the network are in the promiscuous mode i.e. such that they can hear the transmission from other nodes. When a node forwards a packet to neighboring node, WD monitors this forwarding. If neighboring node does not forward the packet to next node or fails to do so, it is detected as a mischievous node and gets reported to Pathrater. WD maintains a buffer for storing recently sent packets. The buffer packet is then matched with the observed packet. If the packet is matched, then no failure is detected and the buffered packet is removed. However, if a mismatch is detected or the packet is not overheard within timeout then failure is implemented for the node and when failure exceeds the threshold then the node is marked as misbehaving. The source of the packet is informed about this misbehaving node. The Watchdog/Pathrater is a solution to the problem of selfish (or “misbehaving”) nodes in MANET. The system introduces two extensions to the DSR algorithm to mitigate the effects of routing misbehavior: the Watchdog, to detect the misbehaving nodes and the Pathrater, to respond to the intrusion by isolating the selfish node from the network operation.

Watchdog runs on each node. When a node forwards a packet, the node’s watchdog module verifies that the next node in the path also forwards the packet. The Watchdog does this by listening in promiscuous mode to the next node’s transmissions. If the next node does not forward the packet, then it is considered to be misbehaving and is reported. This is done by

sending an alarm message to the other nodes on its friends list. When those nodes receive the alarm message, they evaluate it and change the reputation of the accused node only if the alarm source is fully trusted or the same node was accused by several partially trusted nodes. If the Watchdog module that detected the misbehaving node is not in the same node that is acting as source node for the packets, then it sends a message to the source identifying the misbehaving node.

- Watchdog weaknesses
  - Ambiguous collisions
  - Receiver collisions
  - Limited transmission power
  - Misbehavior falsely reported
  - False positives
  - Collusion
  - Partial dropping

The Pathrater module uses the information generated by Watchdog to select a better route to deliver the packets, avoiding the selfish nodes.

After the Watchdog module detects the malicious node, the Pathrater module then deletes the corresponding route from the route cache and tries to determine if there is another route available to the destination by looking in its cache table. If not, Pathrater will broadcast a Route Request to get a new route to the destination.

### **3.2.3 Performance Analysis**

- More than one node in collision can circumvent the WD. For example, a node B forwards a packet to node C but node B does not inform A, if C drops the packet.
- WD does not know the regarding collision occurs at the receiver of the packet.
- Malicious node can drop fewer packets that can be lower than the threshold of the WD.

### 3.3 CORE

CORE is a dynamic reputation based distributed scheme based DSR protocol enforces node cooperation based on collaborative monitoring technique.

#### 3.3.1 Description

CORE nodes operate in promiscuous mode and are required to contribute the in network activities in order to remain trusted and to maintain their reputation. If the node does not participate in network activities or remain idle for a specific time then its reputation degrades. If a provider does not cooperate in network activities, then it leads to exclusion. Requestor requires the provider for the execution of a particular function activates WD for the corresponding function and waits for outcome from WD. Reputation value for provider is updated accordingly as per outcome of WD scenarios when no misbehavior is detected, and a request by misbehaving entity is made.

#### 3.3.2 Architecture

CORE has three major components:

- **Network Entity:** The network entity corresponds to a mobile node. Entity that request the execution of a particular function is called a Requester and entity that executes that particular requested function is called a Provider.
- **Reputation Table:** The Reputation table has entries for nodes that corresponds to Subjective Reputation, Unique Identifier for network entity, Collection of Indirect Reputation and Reputation evaluated for each function.
- **Watchdog Mechanism:** The WD detects the misbehaving network entities, examines the correct execution of the requested function and updates the reputation value accordingly.

For the implementation of critical functions. Entity authentication in a large network, on the other hand, raises key management requirements. If tamper-proof hardware and strong authentication infrastructure are not available, the reliability of basic functions like routing can be endangered by any node of an ad hoc network. The correct operation of the network requires not only the correct execution of critical network functions by each participating node but it also

requires that each node performs a fair share of the functions. No classical security mechanism can help counter a misbehaving node in this context. Node misbehavior that affects network operations (routing, packet forwarding) may range from simple selfishness or lack of collaboration due to the need for power saving to active attacks aiming at denial of service (DoS) and subversion of traffic. Selfish nodes use the network but do not cooperate, saving battery life for their own communications: they do not intend to directly damage other nodes. Malicious nodes, on the other hand, aim at damaging other nodes by causing network outage by partitioning while saving battery life is not a priority. A basic requirement for keeping the network operational is to enforce ad hoc nodes' contribution to network operations despite the conflicting tendency of each node towards selfishness as motivated by the scarcity of node power. We propose a mechanism called CORE to enforce node cooperation based on a collaborative monitoring technique. CORE is suggested as a generic mechanism that can be integrated with any network function like packet forwarding, route discovery, network management, and location management. Each network entity in CORE keeps track of other entities' collaboration using a technique called reputation. The reputation metric is computed based on data monitored by the local entity and some information provided by other nodes involved in each operation. An interesting feature of the CORE mechanism is that denial of service attacks based on malicious broadcasting of negative ratings for legitimate nodes are prevented.

### **3.3.3 The CORE Scheme**

This section presents the CORE scheme in details, starting from the definition of the components that participate to the collaborative reputation mechanism and concluding with the description of the complete process in which the different parts are involved.

### **3.3.4 Components**

- Network entity :

The network entity corresponds to a mobile node. Each entity  $s_i$  is enriched with a set of Reputation Tables (RT) and a watchdog mechanism (WD). The RT and the WD together constitute the basis of the collaborative reputation mechanism presented in this paper.

These two components allow each entity to observe and classify each other entity that gets involved in a request/reply process, reflecting the cooperative behavior of the involved parts. The classification of the entities based on their behavior is then used to enforce the strong binding between the cooperative behavior of a subject and the utilization of the common resources made available by all the other entities of the network. We use the notation requestor when referring to a network entity asking for the execution of a function  $f$  and the notation provider when referring to any entity supposed to correctly execute  $f$ . We also use the notation trusted entity when referring to a network entity with a positive value of reputation

- **Reputation Table:**

The Reputation Table (RT) is defined as a data structure stored in each network entity. Each row of the table includes the reputation data pertaining to a node. Each row consists of four entries: the unique identifier of the entity, a collection of recent subjective observations made on that entity's behavior, a list of the recent indirect reputation values provided by other entities and the value of the reputation evaluated for a predefined function. Each network entity has one RT for each function that has to be monitored.

- **The Watchdog Mechanism in CORE:**

The watchdog (WD) mechanism implements the validation phase depicted in section 2.1 and it is used to detect misbehaving nodes. Every time a network entity ( $s_{i,m}$ , monitoring entity) needs to monitor the correct execution of a function implemented in a neighboring entity ( $s_{j,o}$ , observed entity), it triggers a WD specific to that function ( $f$ ). The WD stores the expected result  $e_r(f)$  in a temporary buffer in  $s_{i,m}$  and verifies if the observed result  $O_r(f)$  and  $e_r(f)$  match. If the monitored function is executed properly then the WD removes from the buffer the entry corresponding to the  $s_{j,o}, e_r(f)$  couple and enters in an idle status, waiting for the next function to observe. On the other hand, if the function is not correctly executed or if the couple  $s_{j,o}, e_r(f)$  remains in the buffer for more than a certain time out, a negative value to the observation rating factor  $\sigma_k$  is reported to the entry corresponding to  $s_{j,o}$  in the RT and a new reputation value for that entity is calculated. It should be noticed that the term expected result corresponds to the correct execution of the

function monitored by the WD, which is substantially different from the final result of the execution of the function.

### 3.3.5 Protocol

The CORE scheme involves two types of protocol entities, a requestor and one or more providers, that are within the wireless transmission range of the requestor. The nature of the protocol and the mechanisms on which it relies assure that if a provider refuses to cooperate (i.e. the request is not satisfied), then the CORE scheme will react by decreasing the reputation of the provider, leading to its exclusion if the non-cooperative behavior persists. For sake of simplicity, the following scenarios are related to the execution of the protocol between a requestor and one provider.

- Protocol execution when no misbehavior is detected

First, the requestor asks for the execution of a function  $f$  to the provider. It then activate the WD related to the provider for the required  $f$  and waits for the outcome of the WD within a predefined time out. Since the two parties correctly behave, the outcome of the WD assures that the requested function was correctly executed and the requestor disarms the WD. We suppose that the reply message corresponding to the result of the execution of function  $f$  includes a list of all the entities that correctly participated to the protocol: the requestor uses this indirect information to update its RT and enters in an idle mode.

- Protocol execution when misbehavior is detected

As described in the previous scenario, the requestor asks for the execution of a function  $f$  and arms the related WD, waiting for the outcome. Since we suppose that the provider does not cooperate, the outcome of the watchdog will be negative. The requestor will then update the entry in the RT corresponding to the misbehaving entity with a negative factor and will enter in an idle mode.



### **3.3.6 Request Made by a Misbehaving Entity**

We describe here the process that any entity receiving a request has to follow. Upon receiving the request for the execution of a function  $f$  the entity checks the reputation value evaluated for the requestor in its global RT. If the reputation value is negative then the entity will not execute the requested function. It has then the choice whether to notify or not the denial of service. A detailed analysis on the best practice will be presented in section 3.4.

### **3.3.7 RT Updates and Distribution**

We focus now on the mechanism used to update and distribute reputation information. RTs are updated in two different situations: during the request phase of the protocol and during the reply phase corresponding to the result of the execution of  $f$ . In the first case, it is possible to notice that only the subjective reputation value is updated. If the outcome of the WD shows that the provider did not cooperate, a negative rating factor will be assigned to the observation and consequently the reputation related to the misbehaving entity will decrease. If no misbehavior is detected, the RTs are not updated.

In the second case, only the indirect reputation value is updated. We suppose that the reply message contains a list of all the entities that correctly behaved: the indirect reputation will be positive and consequently the reputation related to the cooperating entities will increase. The reason why only positive rating factors can be distributed among the entities while the negative rating factors are evaluated locally derives from a possible attack to the protocol. If negative factors could be spread around, it would be simple for a misbehaving entity to distribute false information about other entities in order to initiate a denial of service (DoS) attack. The protocol presented in this paper allows only the distribution of positive rating factors: if we suppose a scenario where collusion between misbehaving entities is impossible, then there would be no advantage for a misbehaving entity to distribute positive rating factors to other unknown entities. Furthermore, reputation information is distributed and updated only during the reply phase avoiding a indiscriminate broadcast of bogus information. Reputation values calculated for each entry of the RT are not constant: if the reputation value is positive then it is decremented along time. The reason why we decided to decrement positive reputation values comes from a possible attack to the CORE scheme: if a network entity enters in an idle status for most of the time

except when it has to communicate, its reputation has to be decreased, even if during the active time it cooperates to the network operation.

### **3.4 Cooperation Enforcement**

This section describes how reputation information is used to enforce cooperation between entities. Reputation is directly related to the cooperative behavior of an entity: if the reputation value is negative then the entity is classified as a misbehaving entity while if the reputation value is positive then the entity is tagged as a trusted entity. The execution of a function requested by any requestor is conditioned by the corresponding reputation value stored in the global RT of the provider: when this reputation value is negative then the provider will deny the execution of the requested operation. There is no advantage for an entity to misbehave because any resource utilization will be forbidden. Reputation is hard to build because positive rating factors are acquired only in the reply message which contains the list of all the network entities that cooperated to obtain of the final result of the requested function. On the other hand, negative rating factors are attributed every time the outcome of the WD is negative. Even if reputation is not linearly decreased for every negative rating factor in order to avoid false evaluations (e.g. apparent misbehavior due to link breaks), a persistent non-cooperative behavior compromises normal resource utilization leading to the exclusion of the misbehaving entity from the network.

### 3.4.1 The CORE Node Operation

Collaborative Reputation (CORE) trust scheme was founded in 2002. CORE scheme differentiate the selfish node and malicious node. The nodes which not cooperate with other nodes in the MANET, for saving battery for its own communication is called “selfish node” while these nodes does not damage other node. The malicious node in MANET behaves abnormally and can damage other nodes by doing any suspicious activity. CORE purposed three different type of reputation:

**1. Subjective Reputation:** Reputation value evaluated by giving priority on Ad-Hoc Networking Systems of mobile node, rather than current one. If malicious node is found out then node’s subjective reputation value is changed by using WD (watchdog) mechanism.

**2. Indirect Reputation:** This value is calculated by providing reputation by one node to other node. Reputation value can be updated through reply message that contains the list of nodes which behaved normally in context of every function. If any node having negative reputation value all requested by that node will be rejected and this node works only as service provider not as requester. For long period of time if this node will provide correct services to all other nodes in MANET, node can achieved their reputation value again. When reputation value is above then the threshold reputation value, that node will again works as service provider as well as service requester.

**3. Functional Reputation:** This reputation is the combination of indirect and subjective reputation value. The weight combine formula is used for calculation of functional reputation value.

The mechanism of CORE is used to impose the cooperation of the nodes. In CORE each entity of the network encourages the collaboration of other entities by using metric cooperation called reputation. This metric is calculated while being based on the local data for each node and can be based optionally on the information provided by other nodes of the network implicated in the interchange messages with the supervised nodes. This reputation is based on the analysis of the behavior (Watchdog) associated each node. A Boolean vector represents a good (with one 1) or a bad (with one 0) behavior. A punishment mechanism is adopted as solution to prevent a selfish behavior for gradually refusing the communication services to the entities which have bad

behavior. This punishment is applied if the metric of reputation (Pathrater) reached a threshold and in this case we declare that the selfish nodes constitute a denial of service and they will be put in the blacklist. Thus the legitimate nodes (which cooperate) reach to save energy.

### **3.4.2. Vulnerabilities of CORE**

CORE suffers unfortunately from important defects. First, it doesn't really resolve the problem of selfish [1]. Immediately, all the selfish nodes see their packets rejected systematically and in this, the protocol is effective. But on the other hand, a quantity of data remains lost, reducing significantly the efficiency of the network. The protocol is based on assumptions (secure routing, single and nonservable addresses) which still remain to make a reality. It's a common disadvantage to all the reputation protocols. Indeed, this one is based on the information observed for the nodes and consequently requires an authentication mechanism in order to affect the marks to the legitimate which could store nonexistent links thus causing the Overflow attack . In addition, it's difficult to avoid the problem of fictitious denunciation (Blackmail) in which a malicious node generates false messages to put up the legitimate nodes on the blacklist. The mechanism of the reputation is potentially vulnerable face up to the cooperative nodes (Black Hole Cooperative) which agree between them to assign good marks and to allocate in the other hand, bad marks the legitimate nodes. Moreover, in that case the nodes couldn't make the distinction between the useful and the useless messages, and will be obliged to forward all the messages which come through them for having their good reputation. This could generate a waste of energy (sleep deprivation) and moreover the constant monitoring nodes would engender a network overload causing a reduction in the bandwidth. In our algorithm we try to fend off the four vulnerabilities cited for endowing CORE with a mechanism called DRI table

### **3.4.3 Performance Analysis**

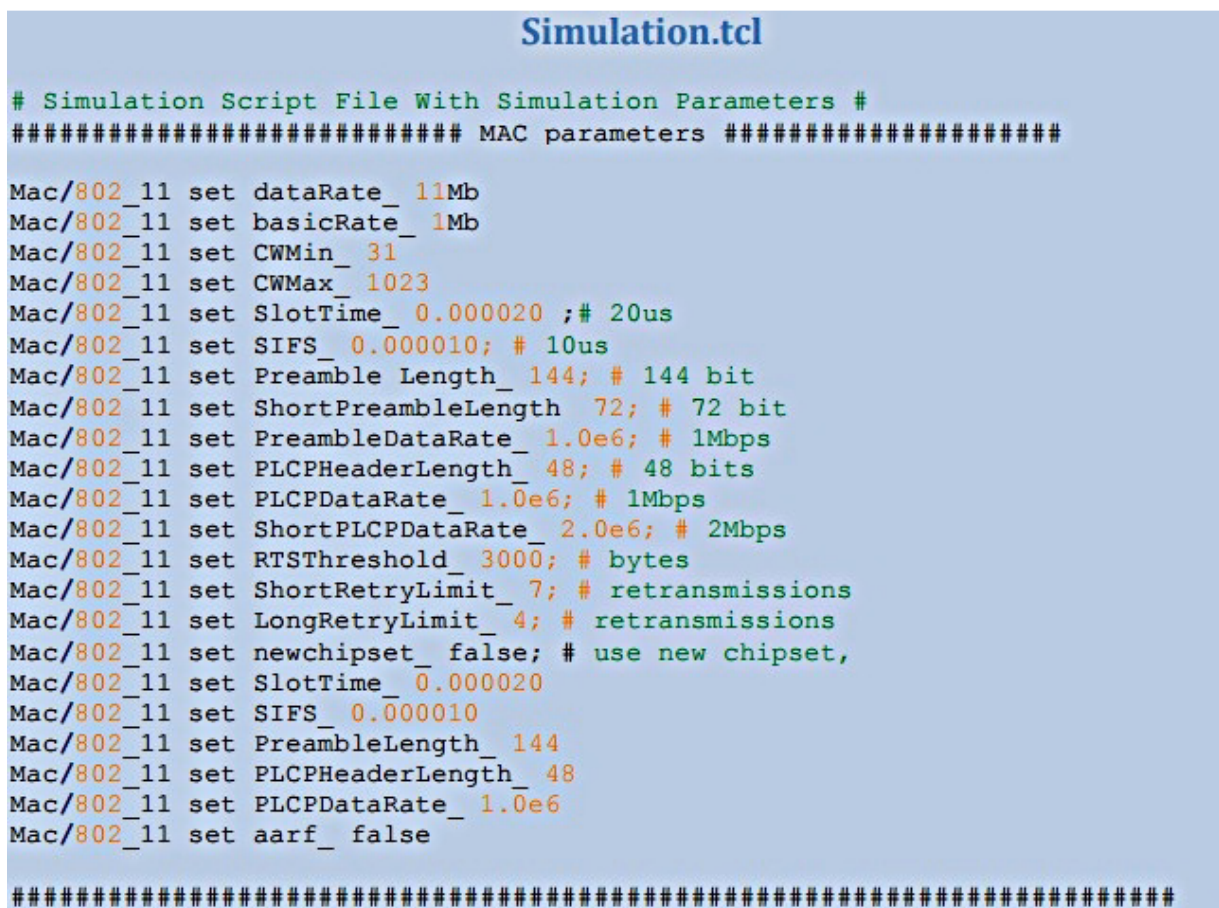
- CORE handles misbehaving nodes, DOS attacks and propagation of fake/negative information.
- Only positive information is shared with other nodes.
- There is no fear of revenge by sharing positive information instead of negative information.
- CORE uses functional reputation. A network work entity is considered for execution of a particular function if its operation value is above a certain threshold else it is ignored.
- CORE does not exclude malicious node from network if the node is well reputed in a function.
- CORE is generic mechanism that can be integrated with network and application layer function.

## Chapter 4

### Implementation Details

#### 4.1 Implementation

For implementation purposes, a number of files were used. The most important files are as follows: x Simulation.tcl: Used to specify the simulation settings and scenarios e.g. wireless model, radios, antenna types, traffic types, topology and simulation duration x ShellScript.sh: This file contains Shell Scripting code and is used to feed dynamic parameters to the Simulation.tcl file. The main aim is to automate the functioning of the simulation by automatically running the simulations with varying parameters (e.g. Traffic, Mobility, Network Size)



```

Simulation.tcl

# Simulation Script File With Simulation Parameters #
##### MAC parameters #####

Mac/802_11 set dataRate_ 11Mb
Mac/802_11 set basicRate_ 1Mb
Mac/802_11 set CWMin_ 31
Mac/802_11 set CWMax_ 1023
Mac/802_11 set SlotTime_ 0.000020 ;# 20us
Mac/802_11 set SIFS_ 0.000010; # 10us
Mac/802_11 set Preamble Length_ 144; # 144 bit
Mac/802_11 set ShortPreambleLength 72; # 72 bit
Mac/802_11 set PreambleDataRate_ 1.0e6; # 1Mbps
Mac/802_11 set PLCPHeaderLength_ 48; # 48 bits
Mac/802_11 set PLCPDataRate_ 1.0e6; # 1Mbps
Mac/802_11 set ShortPLCPDataRate_ 2.0e6; # 2Mbps
Mac/802_11 set RTSThreshold_ 3000; # bytes
Mac/802_11 set ShortRetryLimit_ 7; # retransmissions
Mac/802_11 set LongRetryLimit_ 4; # retransmissions
Mac/802_11 set newchipset_ false; # use new chipset,
Mac/802_11 set SlotTime_ 0.000020
Mac/802_11 set SIFS_ 0.000010
Mac/802_11 set PreambleLength_ 144
Mac/802_11 set PLCPHeaderLength_ 48
Mac/802_11 set PLCPDataRate_ 1.0e6
Mac/802_11 set aarf_ false

#####

```

Figure 4.1 : Screenshot of Simulation.tcl 1

```
#####
set val(chan) Channel/WirelessChannel ;# Channel type
set val(netif) Phy/WirelessPhy ;# network interface
set val(ifq) Queue/DropTail/PriQueue ;# Queue type
set val(ll) LL ;# Link layer
set val(ant) Antenna/OmniAntenna ;# Antenna type
set val(ifqlen) 500 ;# Interface Q len
set val(mac) Mac/802_11 ;# MAC
set val(x) 700
set val(y) 700
set val(prop) Propagation/TwoRayGround

#####
###                               Update Run Time Parameters                               ###
set val(nn) [lindex $argv 0] ;# No. of Nodes
set val(rp) [lindex $argv 1] ;# Routing Protocol
set val(stop) [lindex $argv 2] ;# Stop Time
set sc [lindex $argv 3] ;# Topology File
set val(rate) [lindex $argv 4] ;# Data Rate
puts "nn is $val(nn)"
```

**Figure 4.2 :Screenshot of Simulation.tcl 2**

The purpose of each variable is commented next to it. After defining variables, initialize some global variables to start the simulation. These variable initialization codes are presented in the following code.

```
#####
set ns [new Simulator]
set tracefd [open traceFile.tr w]
set namtrace [open simwrls.nam w]
$ns_ use-newtrace
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
set god_ [create-god $val(nn)]

#####
##                               Configuring the nodes                               ##
$ns_ node-config -adhocRouting $val(rp) \
-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 \
```

**Figure 4.3: Screenshot of Simulation.tcl 3**

In the above code, we have created ns\_simulation object that performs simulation. The tracefd and namtrace variables are file handlers that contain simulation results. In the next lines a topology and general attributes of each host in the network are defined. Most of these attributes use variables that are defined in the start of program. The above code is a common part of all simulations in ns2. After setting common parameters there is a need to create a random topology of nodes. The following code creates 100 wireless hosts in a 2D flat field and set their location in random. In the above code as a first phase a source node with ID=0 is created. The position of source node is set to the left-down corner of the field. After that, set the destination node ID to 99. Then in the for loop body 98 nodes created with random positions. And finally after the loop body destination node in the top-right corner of the field is created.



```
#####
##                               set up connections                               ##
#####
for {set j 0} {$j < 5} {incr j} {
set udp($j) [new Agent/UDP]
$udp($j) set fid $j
set udpsink($j) [new Agent/Null]
}
for {set j 0} {$j < 5} {incr j} {
$ns_ attach-agent $node_($j) $udp($j)
}
set index [expr {$val(nn)-1}]
for {set j 0} {$j < 5} {incr j} {
$ns_ attach-agent $node_($index) $udpsink($j)
set index [expr {$index-1}]
}
for {set j 0} {$j < 5} {incr j} {
$ns_ connect $udp($j) $udpsink($j)
}
for {set j 0} {$j < 5} {incr j} {
```

**Figure 4.4: Screenshot of Simulation.tcl 4**

After creating nodes there is a need to create traffic between source and destination. The CBR packet generator with a UDP agent is used to create and transmit traffic between nodes. In the following code first UDP agent is created to transmit packets using UDP protocol over the network. Also at the destination NULL agent is used to remove the incoming packets to sink. After that, establish a connection between two agents. And finally CBR packet generator is created and attached it to source agent to provide packets for UDP agent.

```
set cbr($j) [new Application/Traffic/CBR]
$cbr($j) attach-agent $udp($j)
$cbr($j) set packet_size_ 1500B
$cbr($j) set rate_ $val(rate)
}
```

**Figure 4.5: Screenshot of Simulation.tcl 5**

The simulations start and finish time need to be determined after creating Agents and traffic. The following code shows these times and start simulation.

```
#####
##### Start and Stop Timings of the connections #####

$ns_ at 20 "$cbr(0) start"
$ns_ at 20.1 "$cbr(1) start"
$ns_ at 20.2 "$cbr(2) start"
$ns_ at 20.3 "$cbr(3) start"
$ns_ at 20.4 "$cbr(4) start"
$ns_ at $val(stop) "$cbr(0) stop"
$ns_ at $val(stop) "$cbr(1) stop"
$ns_ at $val(stop) "$cbr(2) stop"
$ns_ at $val(stop) "$cbr(3) stop"
$ns_ at $val(stop) "$cbr(4) stop"

#####

puts "Loading mobility pattern....."
source $sc

#####
##                               Defining initial node positions for NAM                               ##

for {set i 0} {$i < $val(nn)} {incr i} {
$ns_ initial_node_pos $node_($i) 40
}

#####
##                               Telling nodes when the simulation ends                               ##
```

Figure 4.6: Screenshot of Simulation.tcl 6

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

#####
##                               Ending NAM and the simulation                               ##

$ns_ at $val(stop) "stop"
$ns_ at [expr $val(stop)+0.001] "puts \"Simulation ends \"; $ns_ halt"

#####

$ns_ run
```

Figure 4.7: Screenshot of Simulation.tcl 7

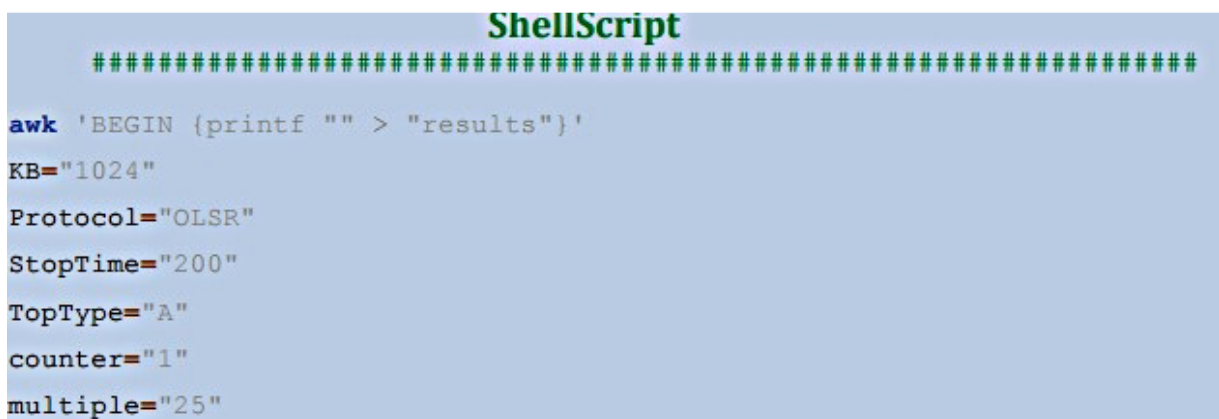
In the above code for loop body sets the finishing time in each node. After that, a simulation finish time is set and starts the simulation. The stop producer is called at the end of simulation to flush and close the result files. The source code of OLSR protocol is very similar to AODV protocol. The first change is routing protocol name in the start of program.

```
Val (rp) OLSR ; #routing protocol
```

The second and last change is setting Hello and TC message broadcast times in OLSR protocol.

## 4.2 Simulation Setting File

This is the main simulation file, which defines all simulation parameters. Most of the Physical, MAC and Network layer parameters are defined in this file, while some parameters, such as number of nodes, total simulation time, data rate of sources and the topology are passed as arguments to this file. The rest of the file is the standard ns tcl file, which involves the creation of sources, sinks, linking them together, and defining other related parameters. There are two main types of traces generated through this file, namely the simulation trace and the NAM trace. The simulation trace file (traceFile.tr) contains the detailed trace of all the events that occurred during the simulation while the NAM trace file simwrls.nam contains the trace of the Network Animation i.e. visualization and movement of nodes.



```
#####  
awk 'BEGIN {printf "" > "results"}'  
KB="1024"  
Protocol="OLSR"  
StopTime="200"  
TopType="A"  
counter="1"  
multiple="25"
```

Figure 4.8: Screenshot of Simulation.tcl 8

```
while [ $counter -le 4 ]
do
Nodes=$((counter*multiple) # total nodes
Topology="/home/AODV-OLSR/Scenarios/"$Nodes"Node-pause5-speed1-700-700-
"$TopType
DataRate="500" # specify the data rate of one source in Kbps
DataRate=$((DataRate * $KB) # x 8 to convert into bits
# Generate the five sources and destinations
src1="0" src2="1" src3="2" src4="3" src5="4"
dst1=$((Nodes-1) dst2=$((Nodes-2) dst3=$((Nodes-3)
dst4=$((Nodes-4) dst5=$((Nodes-5)
initialTime="20"
TimeDuration=$((StopTime-$initialTime) # subtract starting time

ns GenericScript.tcl $Nodes $Protocol $StopTime $Topology $DataRate

printf "$Protocol-$Nodes-$TopType" >> "results"
printf "\n" >> "results"
awk -f PDR.awk duration=$TimeDuration traceFile.tr
awk -f PacketLoss.awk duration=$TimeDuration traceFile.tr
awk -f Throughput.awk duration=$TimeDuration traceFile.tr
awk -f Delay.awk s1=$src1 s2=$src2 s3=$src3 s4=$src4 s5=$src5 d1=$dst1
d2=$dst2 d3=$dst3 d4=
$dst4 d5=$dst5 traceFile.tr
awk -f RtOverhead.awk traceFile.tr
counter=$((counter+1)
done
cp results results-$Protocol-$TopType
```

Figure 4.9: Screenshot of Simulation.tcl 9

### **4.3 Shell Script File**

The ShellScript.sh is the main driving script of the simulation, which calls the network simulator to simulate a given network. This script automates the simulation process by handling multiple simulations in same time. Mainly the script provides the capability to vary the parameters such as network size, data rate, topology, and passes them dynamically to the network simulation executable.

ShellScript.sh file is a Linux Shell script file. After the Network Simulator has completed a simulation, it writes the results in a trace file. The trace file contains all the details of the simulation at the packet level and bind with time. The ShellScript File is used to pass parameters to the Simulation.sh file at run time, by varying the input parameters so that the simulation process is automated. At the start, the shell script initializes some parameters, such as Stop Time, Topology Type and then run a loop. Inside that loop the author calculated number of nodes (this varies from 25 to 100 for successive iterations), specified the data rate, the source and destination nodes and then pass all these variables to the Simulation.tcl file as input parameters so that the Simulation.tcl file can execute the simulation based on these parameters. The script at the end calls various AWK files to calculate the evaluation metrics such as throughput, PDR, delay and writes these values to a result file for later reference.

## 4.4 Simulation Settings

In order to carry out the performance comparison of AODV and OLSR in a MANET topology, we have used the open source Network Simulator (ns-2) for this purpose. Following are the details of the simulation parameters and settings.

### . Simulation Parameters

The following table lists the detailed simulation parameters used in our experiments

<b>Simulation Parameters</b>	<b>Value</b>
Network Area	700m × 700m
No. of Nodes	25,50,75,100
Traffic Type	CBR/UDP
No. of Flows	5
Packet Size	1500 Bytes
MAC Protocol	IEEE 802.11b
Data Rate	11 Mbps
Frequency	2.5 GHz
Propagation Model	Two-Ray Ground
Transmission Power	281mW
Antenna	Omni-Directional
Simulation Time	2000s

**Table 1: Simulation Parameters**

## 4.5 Simulation Topologies

The following figures show the simulation topologies for 25, 50, 75 and 100 nodes.

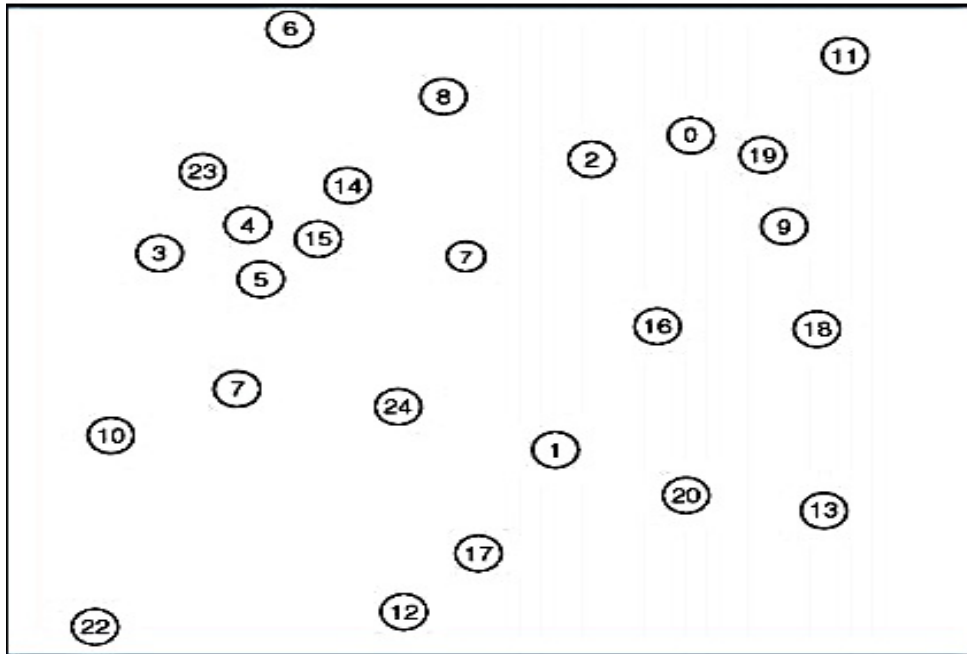


Figure 4.10: 25 Node Topology

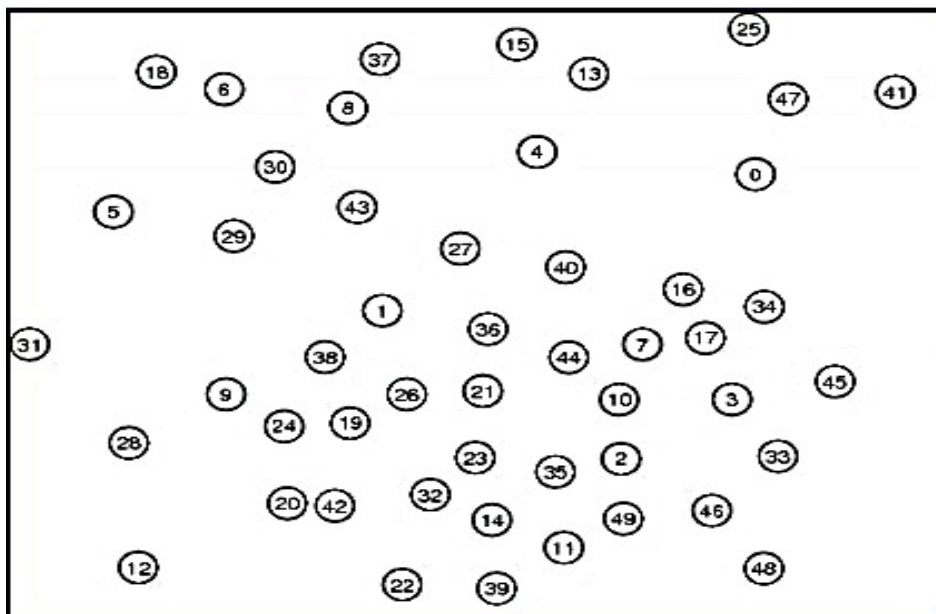


Figure 4.11: 50 Node Topology

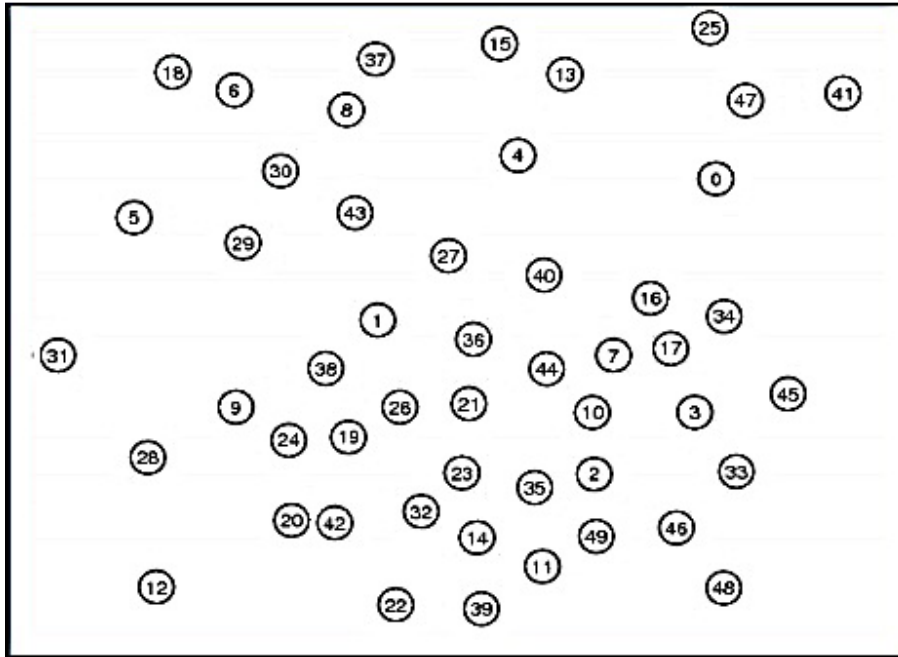


Figure 4.12: 75 Node Topology

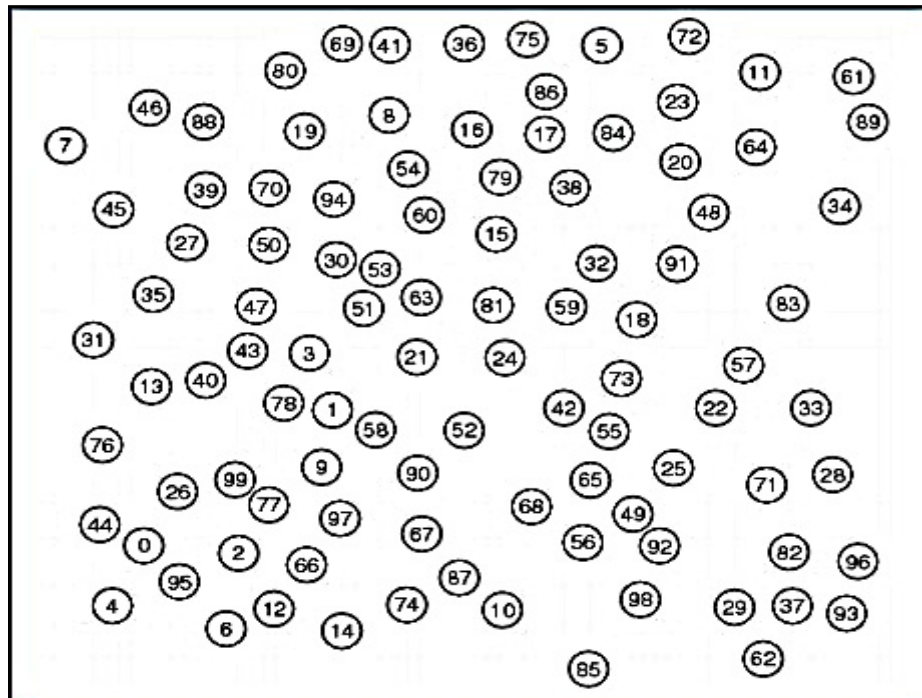


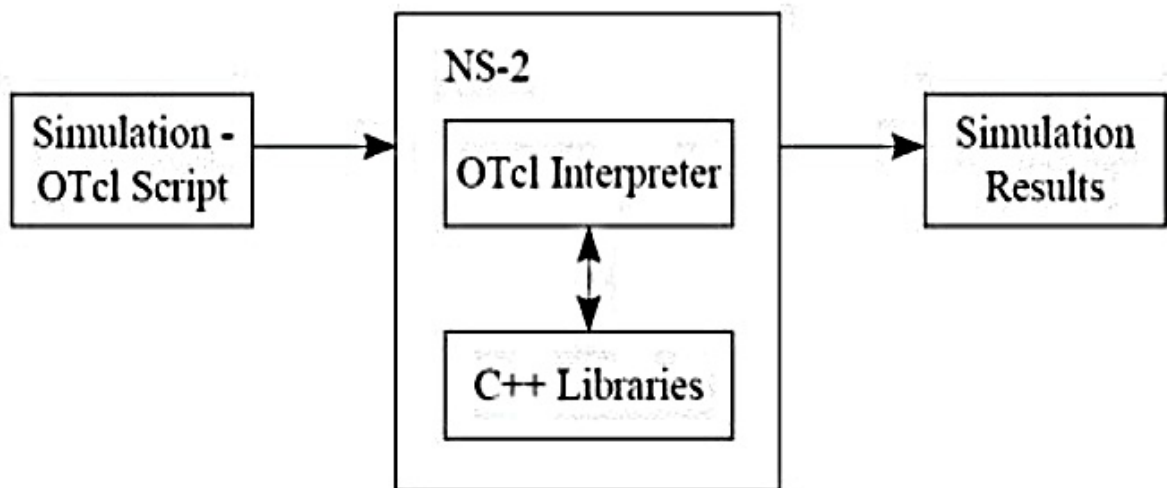
Figure 4.13: 100 Node Topology



## 4.6 Simulation Modeling

The aim of this research is to analyze the performance of routing protocol by varying the number of nodes from 25 nodes to 100 nodes. The performance is evaluated by means of multiple simulations with respect to metrics such as: packet delivery ratio, packet loss ratio, aggregate throughput, end-to-end delay, routing overhead. The experiment is performed by the use of Network Simulator ns (licensed for use under version 2 of the GNU General Public License) to compare the group of chosen routing protocols representing specific approaches and algorithms. NS2 (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkeley written in C++ and OTcl. The choice of this simulator is motivated by many advantages, among which:

- It is open source software
- Large amount of implemented protocols and contribution code.
- It consists of different topology and traffic generators which helps users to create different scenarios.
- Reliability confirmed by common usage for research purposes.
- It provides an interface to users to configure different network protocols to each network layer.



**Figure 4.14: Data Flow For a Single Simulation**

The simulation and analysis process are specified in the Business Process Model and Notation (BPMN) diagram. Parameters such as dimensions of topology (width and length), number of nodes in the scenario, name of simulated routing protocol and number of simulations repeated in the sequence are required by OTcl script. Two external source files are used for simulation and these files contains OTcl code for nodes movement and positioning in one file, and traffic pattern in the other file.

## 4.7 Model Design And Implementation

In this project, the author has designed a wireless Ad-Hoc network with the simulation area to be 700\*700 sq. units.

```
Phy/WirelessPhy set bandwidth_ 11Mb;
Phy/WirelessPhy set freq_ 9.14e+08
Phy/WirelessPhy set Pt_ 0.281838
#Phy/WirelessPhy set RXThresh_ 6.0908e-10
Phy/WirelessPhy set RXThresh_ 4.65262e-10

#####

set val(chan) Channel/WirelessChannel ;# Channel type
set val(netif) Phy/WirelessPhy ;# network interface
set val(ifq) Queue/DropTail/PriQueue ;# Queue type
set val(ll) LL ;# Link layer
set val(ant) Antenna/OmniAntenna ;# Antenna type
set val(ifqlen) 500 ;# Interface Q len
set val(mac) Mac/802_11 ;# MAC
set val(x) 700
set val(y) 700
set val(prop) Propagation/TwoRayGround
```

**Figure 4.15: Screenshot of Simulation.tcl 10**

To understand the above parameters the simulation should be conducted and concentrated on

- Traffic patterns
- Mobility models o Interface queues
- Parameters affecting radio propagation

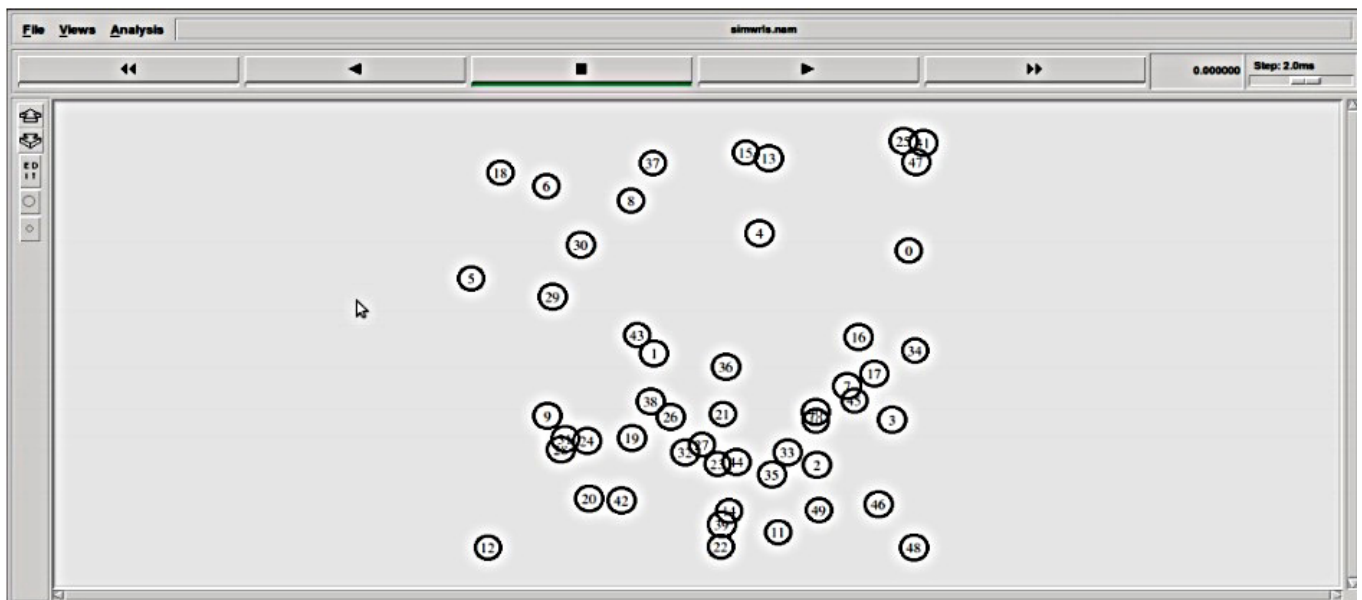
### 4.7.1 Node Movement

There is a significant distinction made between mobile and router nodes in simulation topologies in order to illustrate real conditions [1], [4]. The main difference is the lack of movement for router nodes. A tool is created to generate movement animation of nodes this tool takes ns2 file to generate network animator (NAM). It is important to note that this generation of animation file is not a post simulation trace. The network animator (NAM) file that is produced by network simulator (NS2) is playable animation file.

### 4.7.2 Node Transmission range

Apart from mobility, the router and mobile node's properties differ in the matter of receiving threshold and transmitting power[1],[3]. The value of receiving threshold (represented by variable RXThresh\_assigned to network interface type Phy/Wireless Phy in OTcl simulation code) is assigned to a wireless node and determines the minimum value of packet's signal power required to succeed with its delivery. If the packet's signal power at the destination node doesn't reach the threshold value, it is marked as error and dropped by the MAC layer.

The following are the figures for nodes movement before and after simulation for 50 and 100 nodes.



**Figure 4.16: 50 Nodes Before Simulation**

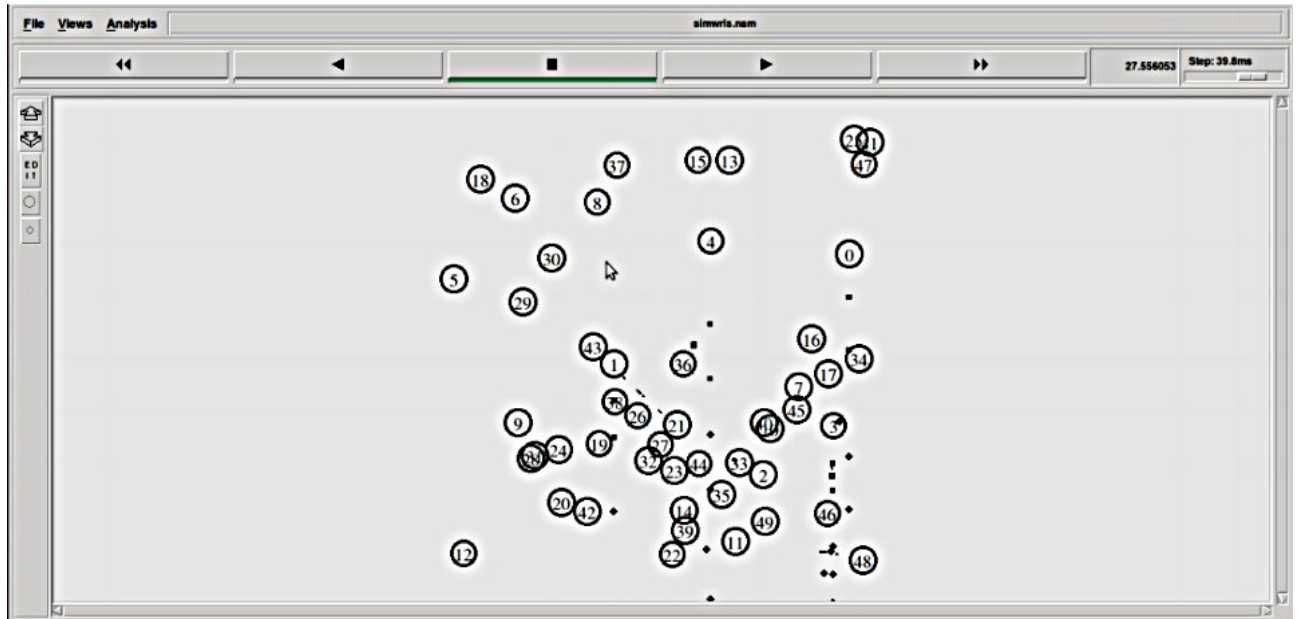


Figure 4.17: 50 Nodes After Simulation

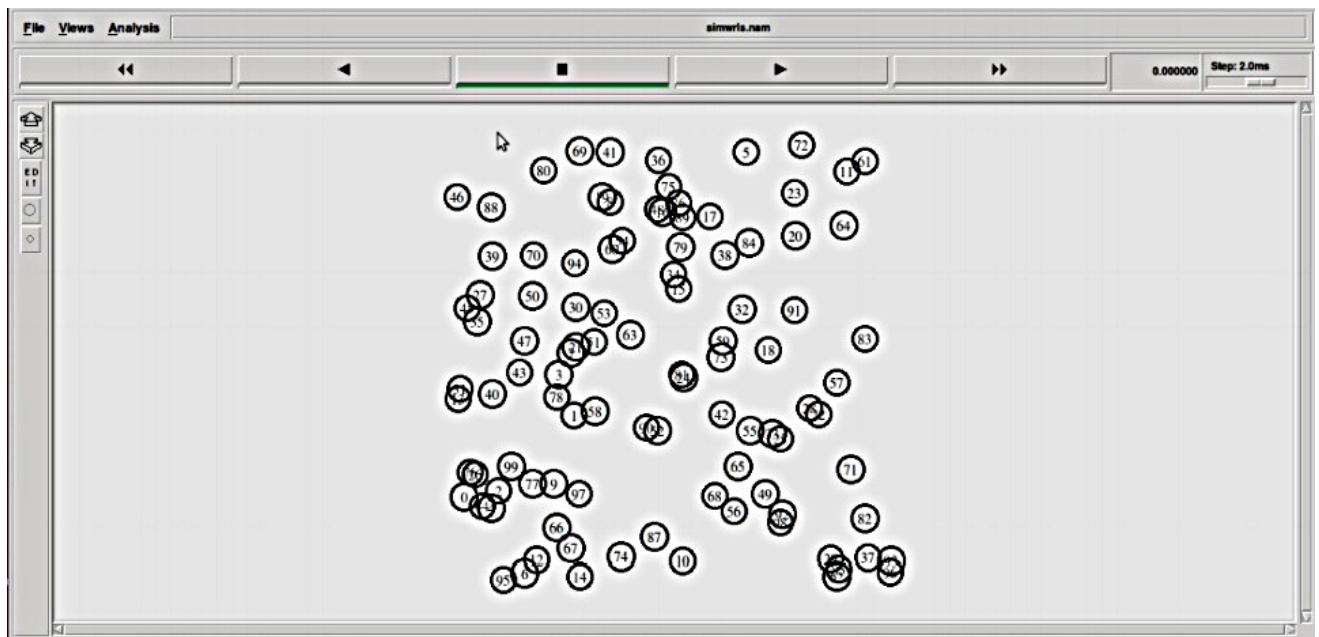
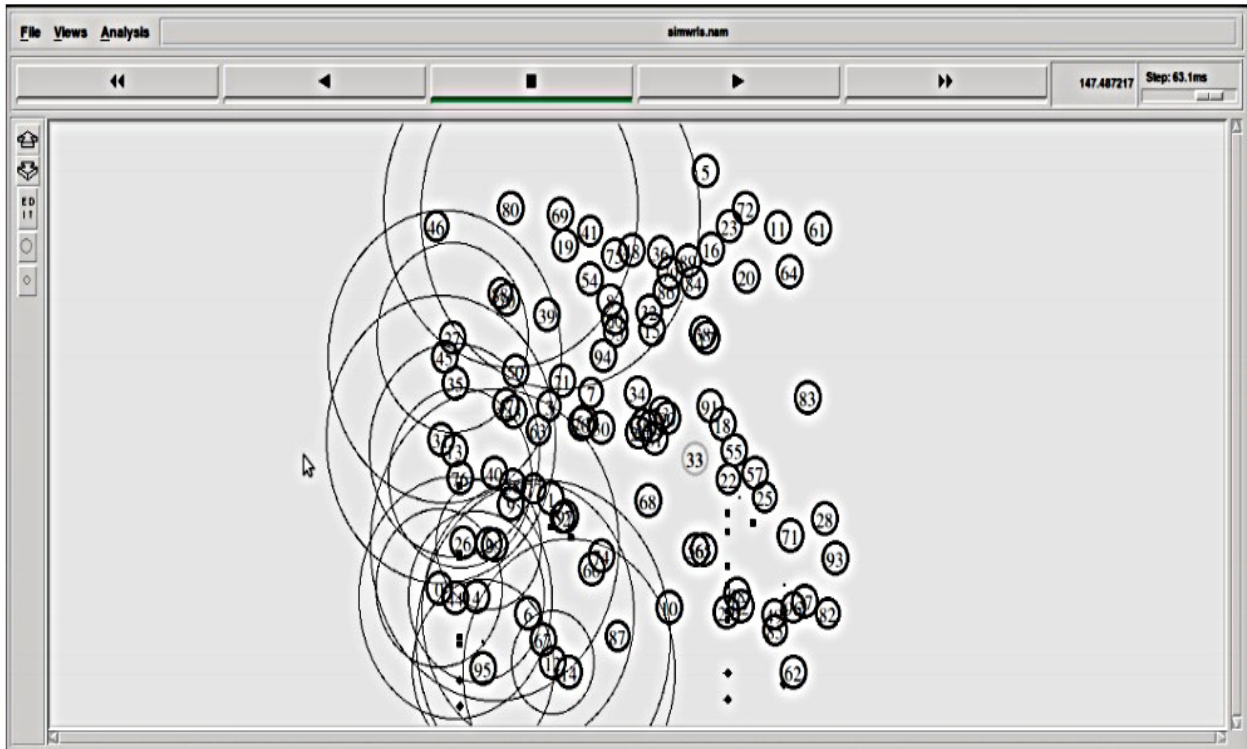


Figure 4.18: 100 Nodes Before Simulation



**Figure 4.19: 100 Nodes After Simulation**

```
Loading mobility pattern.....
SORTING LISTS ...DONE!
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
Packet Delivery Ratio: 80.975781%
Pkt Loss Ratio: 19.024219%
Throughput: 1990 Kbps
avg e-to-e delay: 636 ms
Routing Overhead: 75 KB
nn is 100
num_nodes is set 100
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
INITIALIZE THE LIST xListHead
Loading mobility pattern.....
SORTING LISTS ...DONE!
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
Packet Delivery Ratio: 74.008707%
Pkt Loss Ratio: 25.991293%
Throughput: 1820 Kbps
avg e-to-e delay: 939 ms
Routing Overhead: 290 KB
```

Figure 4.20: Screenshot of Evaluation Program

### 4.7.3 Physical and MAC layers

Wireless mesh routers in the simulation are equipped with IEEE 802.11b compliant wireless cards and the Physical and MAC layers of IEEE 802.11b are used.

### 4.7.4 Radio Propagation Model

The popular Two-Ray Ground radio propagation model is used to model the wireless communication. The two-ray Ground Model is a radio propagation model that predicts path loss when the signal received consists of the line of sight component and multi path component formed predominately by a single ground reflected wave. In practice, a single line-of-sight path between two mobile nodes is seldom the only means of propagation. The two-ray ground reflection model considers both the direct path and a ground reflection path. In general, this model gives more accurate prediction at a long distance than the free space model.

### **4.7.5 Omni-directional antenna**

An Omni-directional antenna transmits and receives signals equally, in all directions. That is, an Omni directional antenna transmits signals in a 360° angle. The advantage of such an antenna is that it covers all directions and provides connectivity in all directions, but the disadvantage is that since the energy is scattered in all directions, the wireless range is somewhat limited. This is in contrast to directional antennas which perform beam-forming in a particular direction only, giving a higher range but limited degree of coverage.

### **4.7.6 Topology and Traffic Settings**

The network size is varied from 25 nodes to 100 nodes with every topology comprising of the selected number of nodes randomly distributed in an area of 700m x 700m. Five randomly selected nodes act as the sources of five different flows and other five randomly selected nodes act as the destinations of these flows.

### **4.7.7 Routing and Transport Protocols**

At the network layer, two protocols AODV and OLSR are compared. At the transport layer, UDP protocol was used.

## 4.8 Evaluation Matrices

The performance of routing protocols is measured through performance metrics including the throughput, end-to-end delay and the packet delivery ratio. In general, as the traffic load increases, the routing protocol needs to transport more data across the network, which causes more transmissions on the wireless medium, resulting in more collisions and packet losses. Similarly, high mobility also strains the performance of the routing protocol by involving constantly changing routes. The end-to-end delay is also higher for high traffic, mobile topologies since there are a large number of collisions, which requires more frequent retransmissions at the link layer, resulting in long delays. In particular, the end-to-end delay is also tightly coupled with the network size since a large network has longer routes on average, requiring more hops and consequently, more delay.

- **Packet Delivery Ratio:** The packet delivery percentage represents the percentage of total sent packets from source nodes, which are successfully received at the destination nodes.
- **Packet Loss Ratio:** The Packet Loss Percentage (or Ratio) represents the total number of packets lost in the network between source and destination nodes.
- **Aggregate Throughput:** The aggregate throughput is the total number of bytes received at the destination divided by the total time duration. This aggregates all the flows in the network.
- **End-to-End Delay:** The end-to-end delay is the averaged results of how long it takes a packet to go from the source to the destination.
- **Routing Overhead:** The measure of routing packets (non-data) generated by the protocol.



## CHAPTER 5

### Simulation Scenarios

#### 5.1 Description and Motivation about Scenarios

To carry out the performance evaluation, the three parameters are varied and the impact of those parameters on the performance of the two protocols is observed. In general, as the network size increases, the average route length increases, and the routing protocol has to carry the data through a larger number of wireless hops which introduces more delays and more probability of collisions, therefore the performance degrades. Moreover, an increase in load also overloads the network since the wireless medium is a shared medium and the Carrier-Sense-Multiple-Access (CSMA) mechanism of IEEE 802.11 radios is prone to collisions especially in high traffic conditions. Therefore, the routing protocol performance worsens in the face of increasing load. As the mobility is increased, it causes rapid changes in the network topology whereby old links break and new links and routes are created. This requires the routing protocol to constantly adapt to the changing topology and this typically degrades the performance of the routing protocol since it needs to update the routing tables and creates additional routing packets which cause further strain on the wireless medium.

**6.1 Network Size** The network is varied from 25 nodes to 100 nodes in order to study the scalability of the routing protocol. It is extremely important for a routing protocol to perform well for large networks as well as for small networks. By varying the size, the aim is to study the scalability of the routing protocol in terms of how well it addresses the maintenance of a large number of nodes and routes. The network size is varied from 25 nodes to 100 nodes in increments of 25 nodes. The selected area of simulation is 700mx700m, which provides sufficient space for nodes to be mobile and sufficiently placed apart to observe the impact of multihop routing. The network size is varied so that the behavior of the two protocols scales with the network size. More importantly, as the network size increases, the link (and route breakage) probability increases.

**6.2 Traffic Load** To study the impact of traffic load on the performance of the protocols, the input traffic load is varied from 1 Mbps to 4 Mbps in increments of 1 Mbps while keeping other parameters such as Network Size and Mobility constant. The traffic load strains the network and creates additional load on the wireless network and hence it gives a good idea of the performance of the protocol under heavy load conditions.

The input load is varied because as the network load increases, the collisions on the wireless medium also increase along with packet losses. Thus, it is interesting to see the behavior of the two protocols as the network load increases. 6.3 Mobility has a significant impact on the performance of routing protocols because mobility causes changes in the topology of the network. More precisely, mobility causes route breakages and creation of new routes, which forces the routing protocol to converge again. This enables us to study how well the protocol performs in terms of dynamically evolving network conditions. Also vary the mobility of the network by varying the pause time from 5s to 15s. The mobility is an important criterion in the performance evaluation of ad-hoc routing protocols. High mobility creates stress on the network in terms of higher route breakages, high packet loss probability. Therefore, it is interesting to see the performance of the two protocols under varying mobility scenarios.

Packet lost may have various reasons:

1. **Packet collision:** two nodes send packet in same time.
2. **High packet rate at source node:** if the packet generation rate is higher than link bandwidth some packets are lost.
3. **Incoming packet queue size:** if the incoming queue size is low during the routing some packets may lost because of full queue.
4. **Routing delay:** if the node cannot find a route to destination in a reasonable time it drops the packet.

## CHAPTER 6

### Results and Discussion

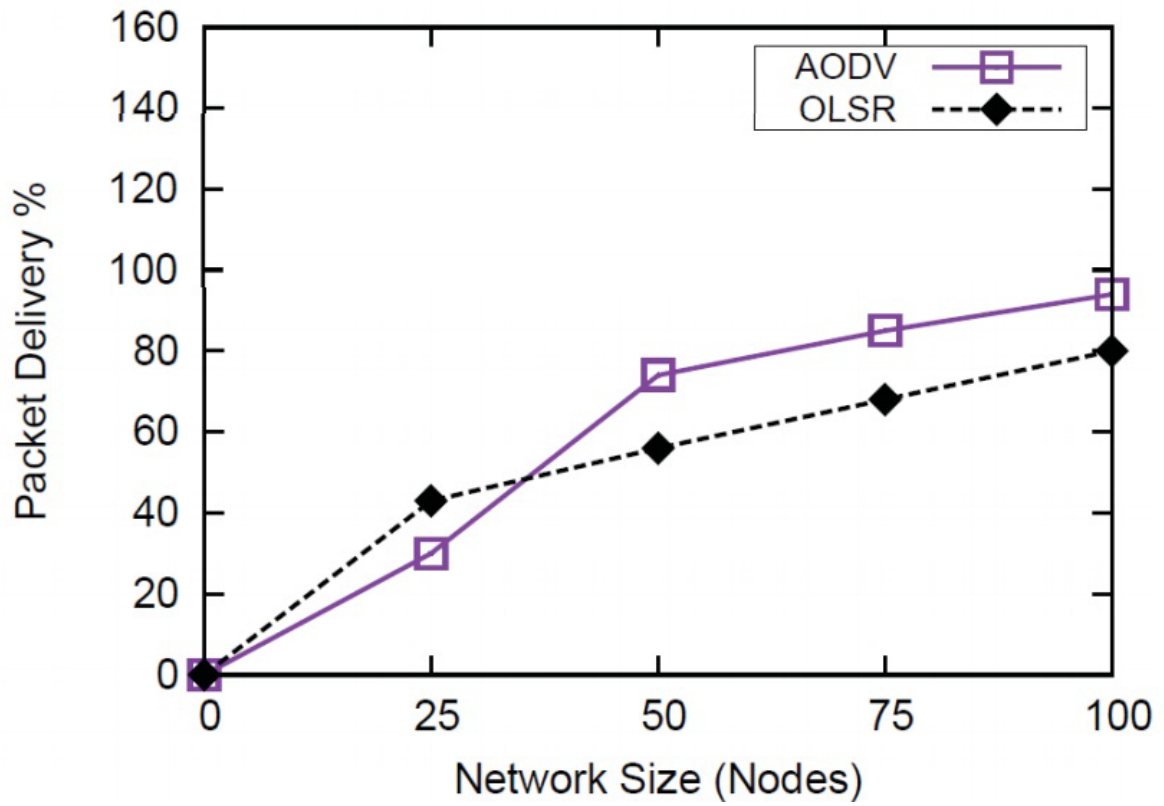
#### 6.1 Performance Evaluation Results

In this section, a performance comparison of AODV and OLSR protocols is carried out by varying network size, varying traffic and varying mobility and gives their comparison in terms of the selected evaluation metrics. To prove the observations, 95% confidence interval for the sample difference between two routing protocols is calculated. If the confidence interval shows zero then one can conclude that the routing protocols have almost same performance. For example, the calculation of 9% confidence interval for AODV and OLSR shows similar results. After calculating mean ( $\bar{x}$ ) for a pair wise difference of the two samples of two protocols, standard deviation ( $\sigma$ ) of the sample difference was determined. Since the number of samples is 40, the 95% confidence interval of the two protocols will be as follows:

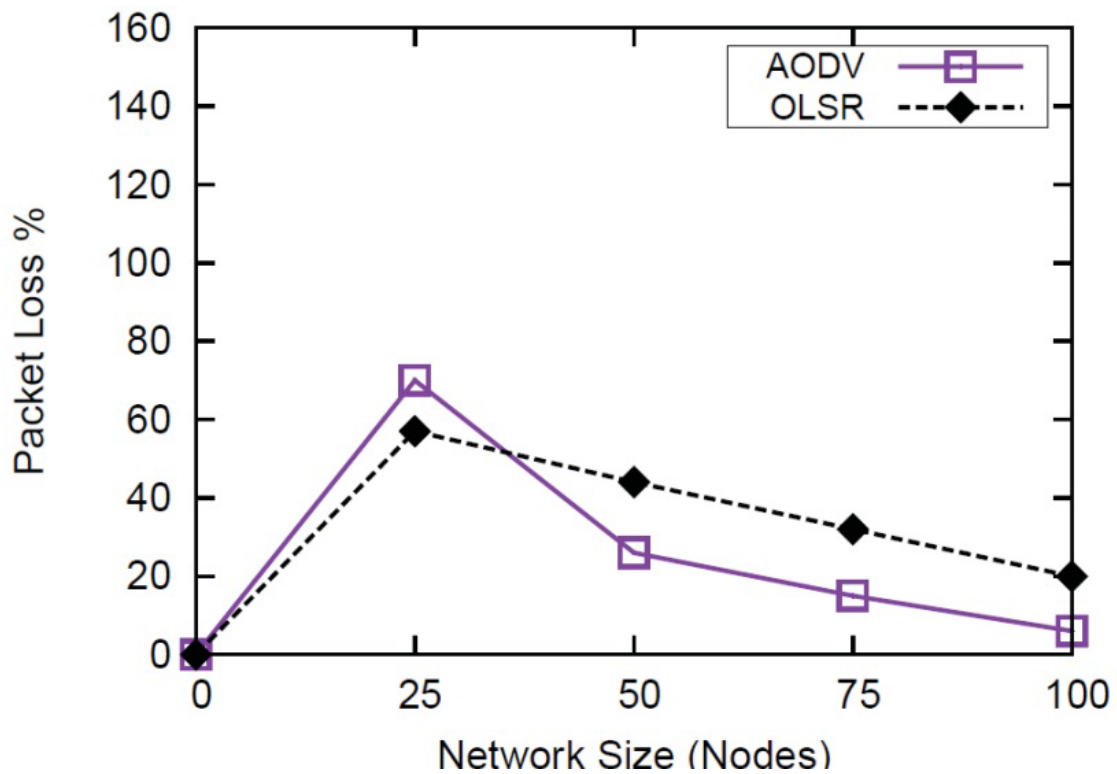
This interval does not include zero, we can conclude with 95% confidence interval that AODV is significantly better than OLSR. The confidence interval is also presented in tables.

##### 6.1.1 Performance Comparison of AODV and OLSR with Varying Network Size

The following figure 6.1 shows the comparison between AODV and OLSR with regard to packet delivery performance for varying network sizes i.e. 25-100 nodes. Initially (25 nodes), OLSR outperforms AODV because it is proactive in nature and creates routes in advance, whereas AODV wastes sometimes in creating routes. The overhead of OLSR is small for smaller topologies, however, for larger topologies i.e. 50, 75 and 100 nodes, the significantly large routing overhead of OLSR degrades performance, creating interference in the network and causing loss of packets (figure 6.2). On the other hand, AODV creates significantly smaller overhead and hence causes fewer collisions even for larger topologies, thereby achieving a better PDR.

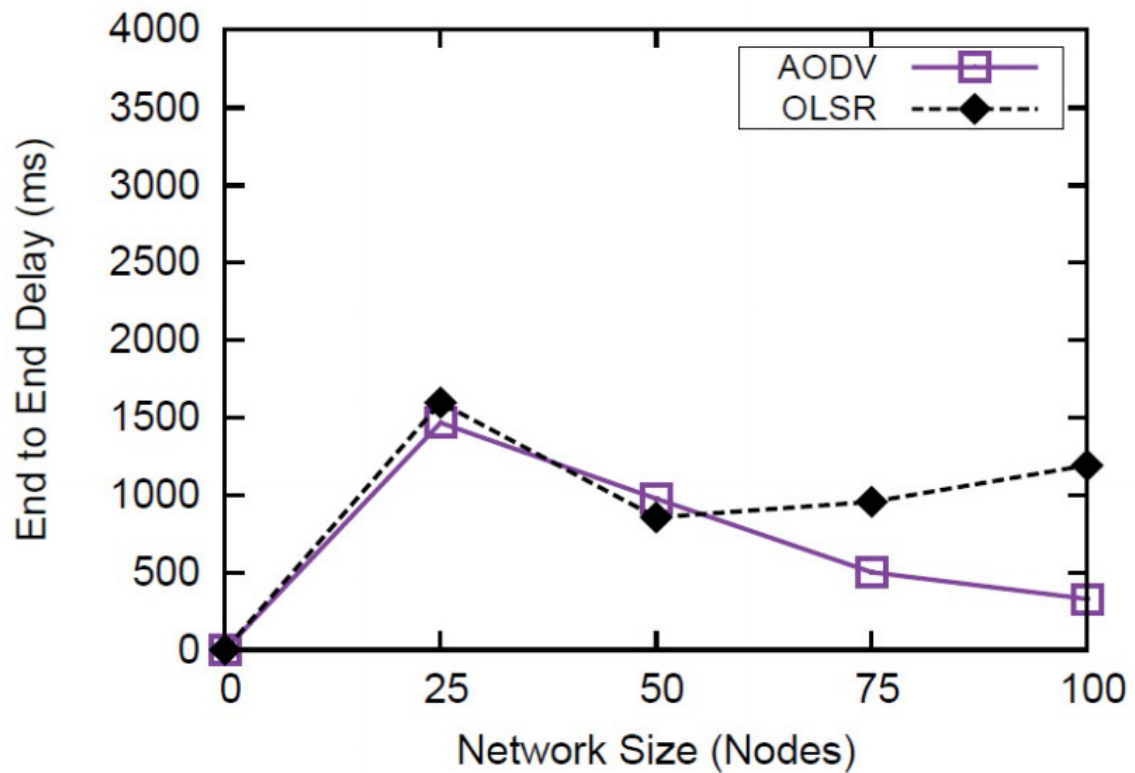


**Figure 6.1: Packet Delivery Percentage for AODV and OLSR**



**Figure 6.2: Packet Loss Percentage for AODV and OLSR**

Figure 6.3 shows the comparison of end-to-end delay of the two protocols. The overall end-to-end delay for the two protocols is comparable but OLSR has a slightly higher delay compared to AODV. The primary reason is that, for larger topologies, OLSR creates more routing packets due to its proactive nature, which causes collisions and results in larger delays compared to AODV, creating so a similar routing overhead for all topologies.



**Figure 6.3: End-to-End Delay for AODV and OLSR**

Figure 6.4 shows comparison of routing overhead generated by the two protocols. OLSR being a proactive protocol creates a significantly larger routing overhead especially for larger topologies. OLSR generates a lot of HELLO and Topology Control messages, which results in larger overhead while AODV relies on infrequent Route Discoveries which generate less traffic.

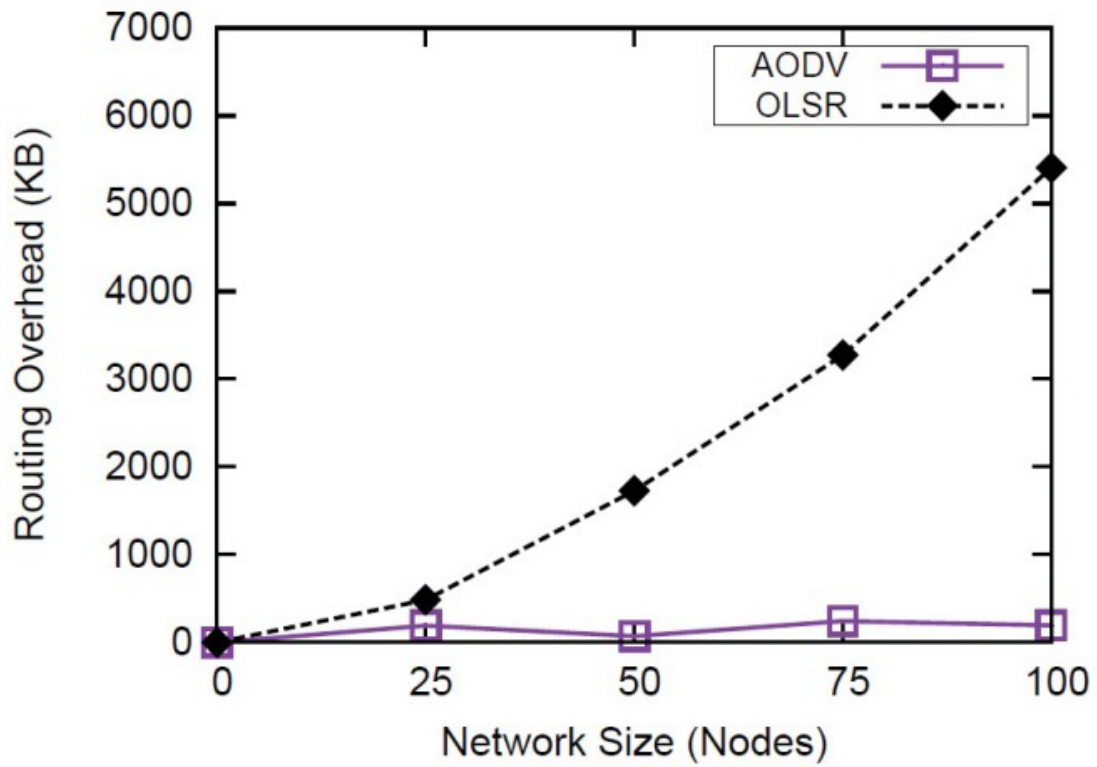


Figure 6.4: Routing Overhead for AODV and OLSR

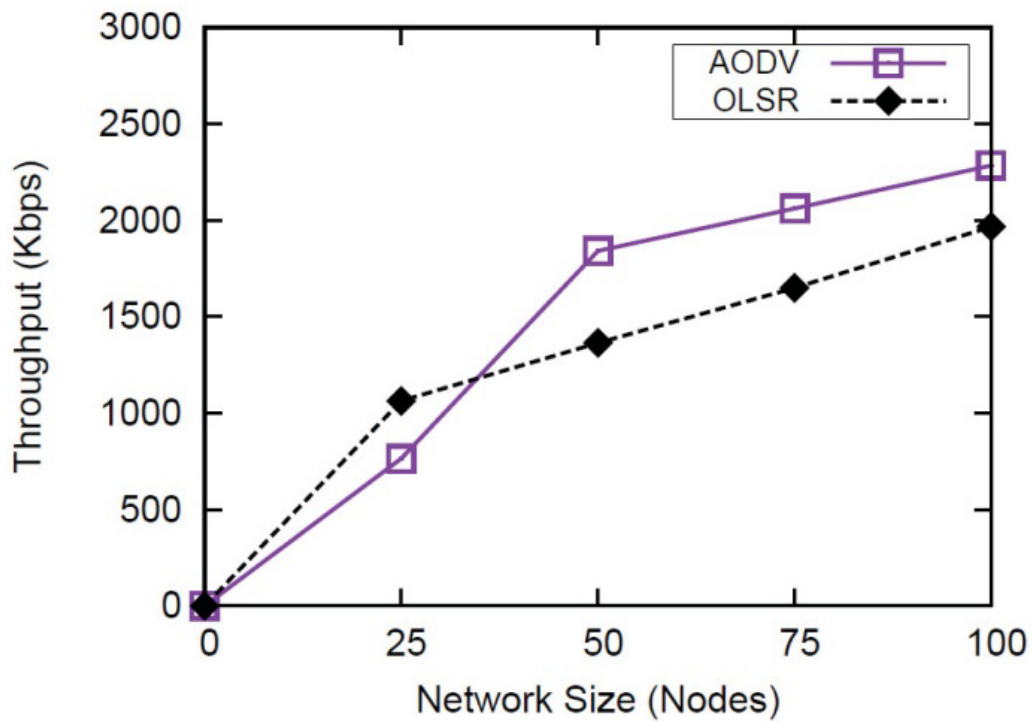


Figure 6.5: Throughput for AODV and OLSR

The throughput is another representation of the Packet Delivery Ratio (figure 6.5). AODV provides a higher throughput for larger topologies because it has a smaller routing overhead compared to OLSR which creates a lot of overhead for larger topologies.

## 6.2 Performance Comparison of AODV and OLSR with Varying Traffic

As the traffic load is varied, AODV performs relatively better than OLSR, because AODV being a reactive protocol launches the route discovery process relatively infrequently whereas OLSR generates periodic routing traffic (figure 6.6). Moreover, mobility causes significantly more changes for OLSR (neighbor detection, Topology Control) compared to AODV. Excessive packets worsen the network conditions as the load increases and hence OLSR performs worse than AODV. Overall, the performance of both protocols deteriorates as the load increases (figure 6.7). Hence, we see decreased packet delivery rates and increasing packet loss rates for both protocols.

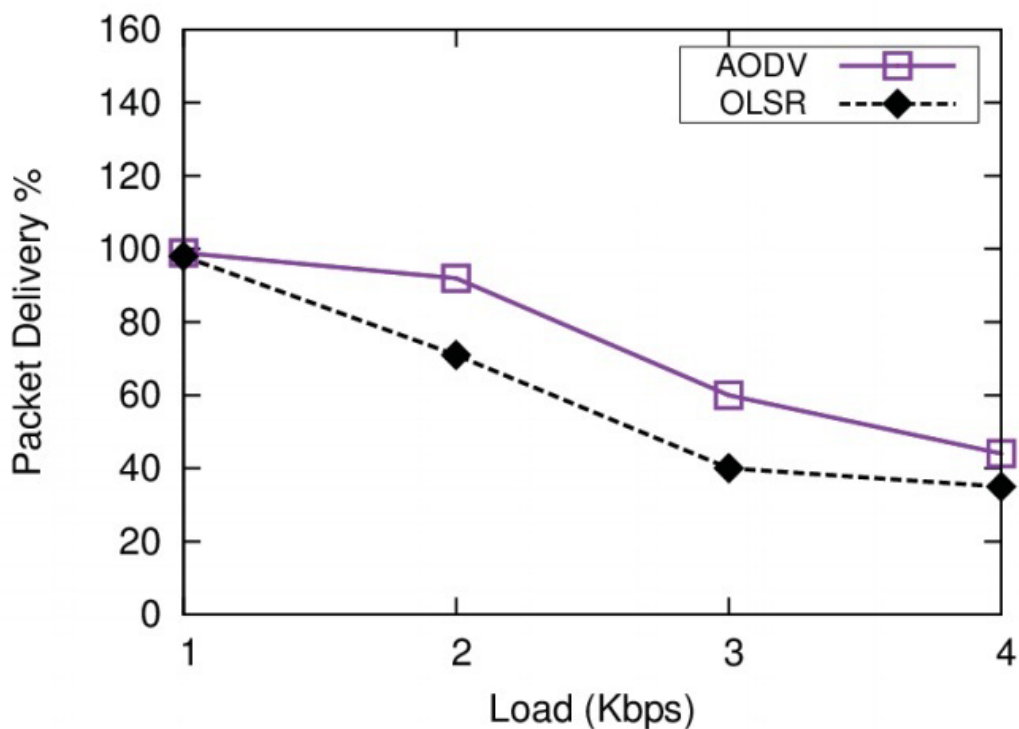
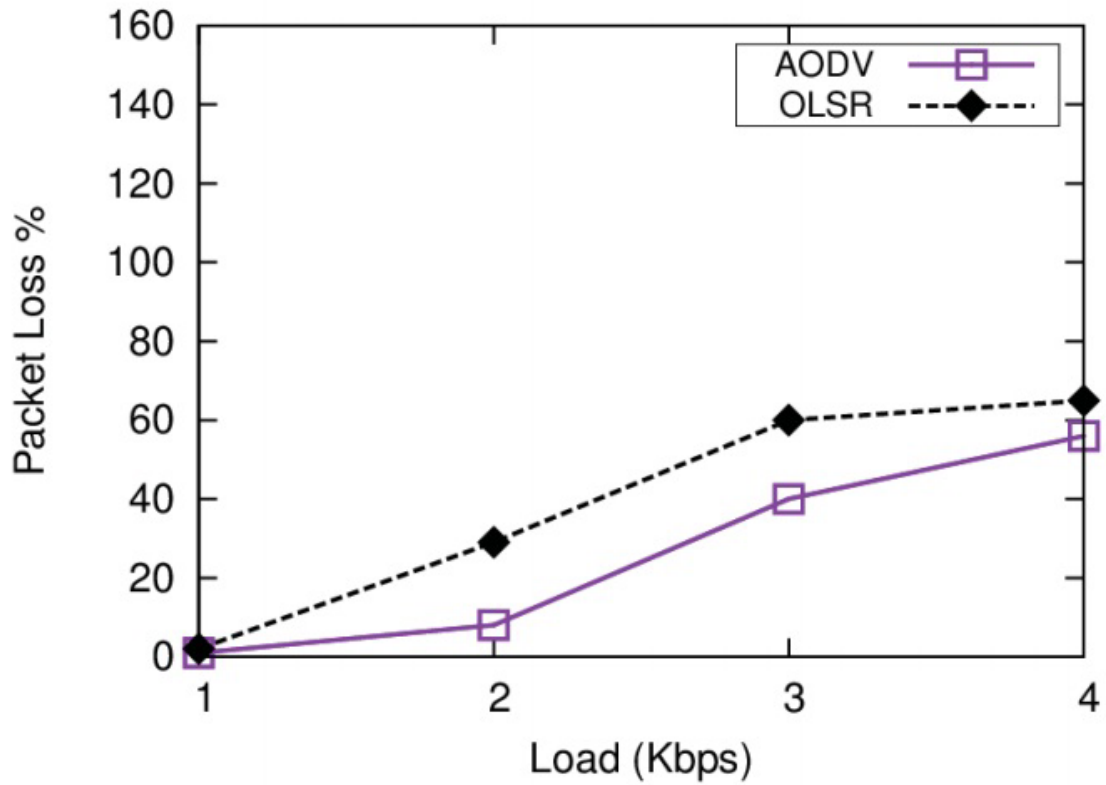


Figure 6.6: Performance Comparison of AODV and OLSR with Varying Traffic



Both protocols show comparable performance in terms of end-to-end delay, as the traffic load is increased on the network (figure 6.8). Overall, we see that both the protocols have increasing delays as the traffic load is increased because increased traffic on the wireless medium causes collisions which in turn necessitate retransmissions at MAC layer, resulting in larger end-to-end delays.



**Figure 6.7: Packet Loss Percentage for AODV and OLSR**

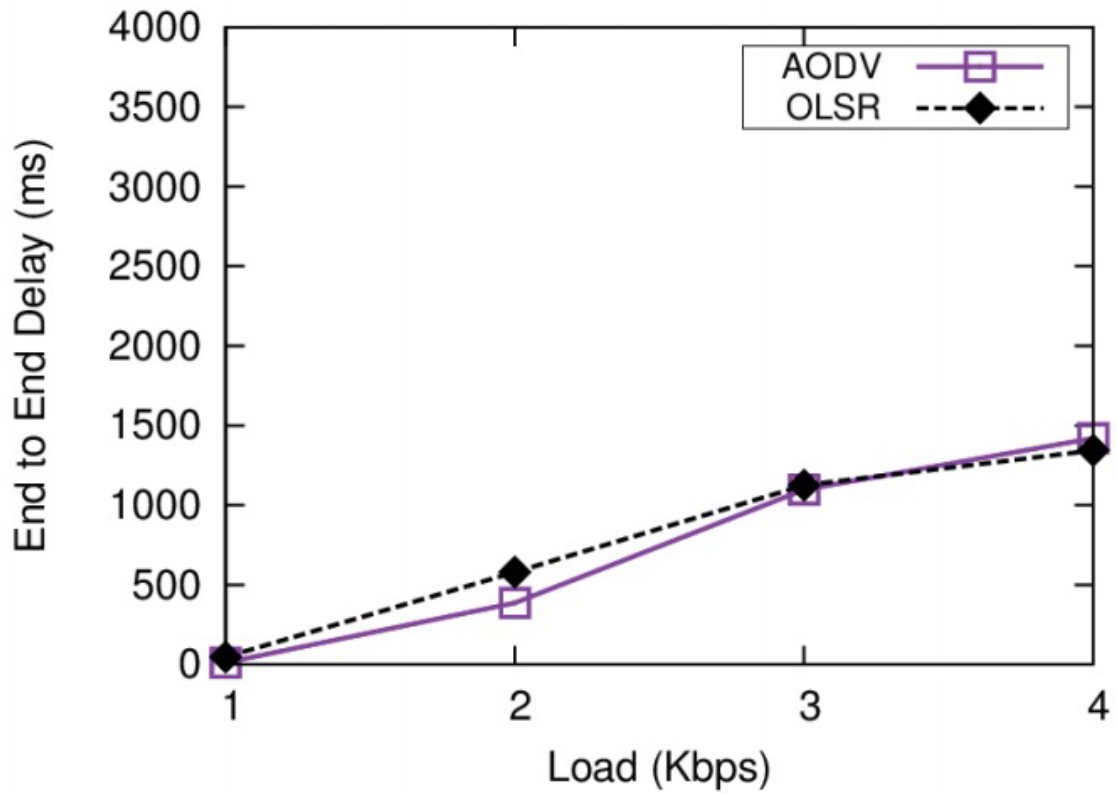


Figure 6.8: End-to-End Delay AODV and OLSR

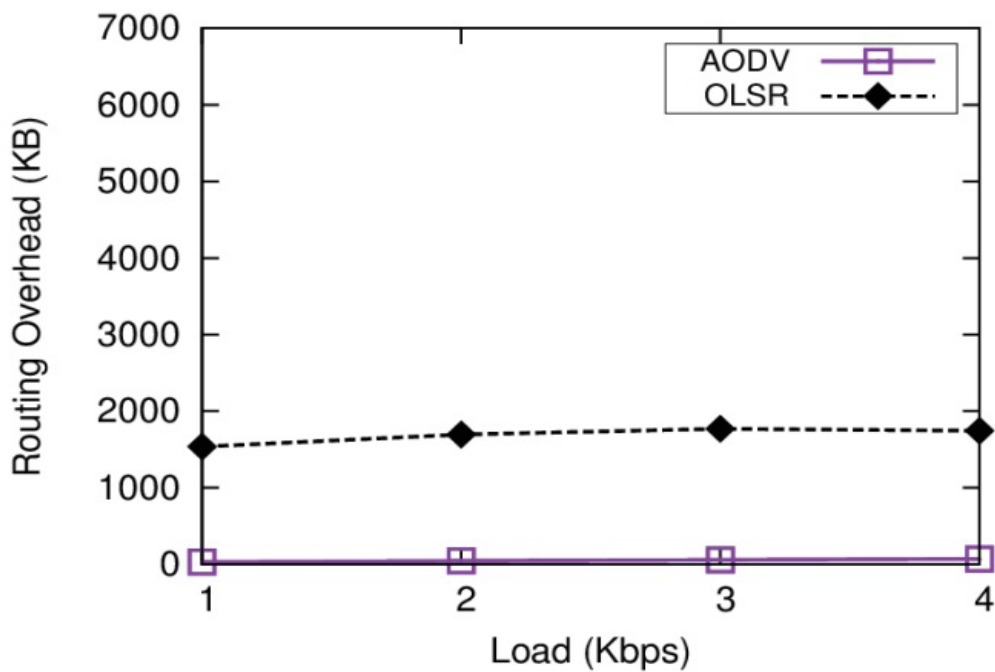
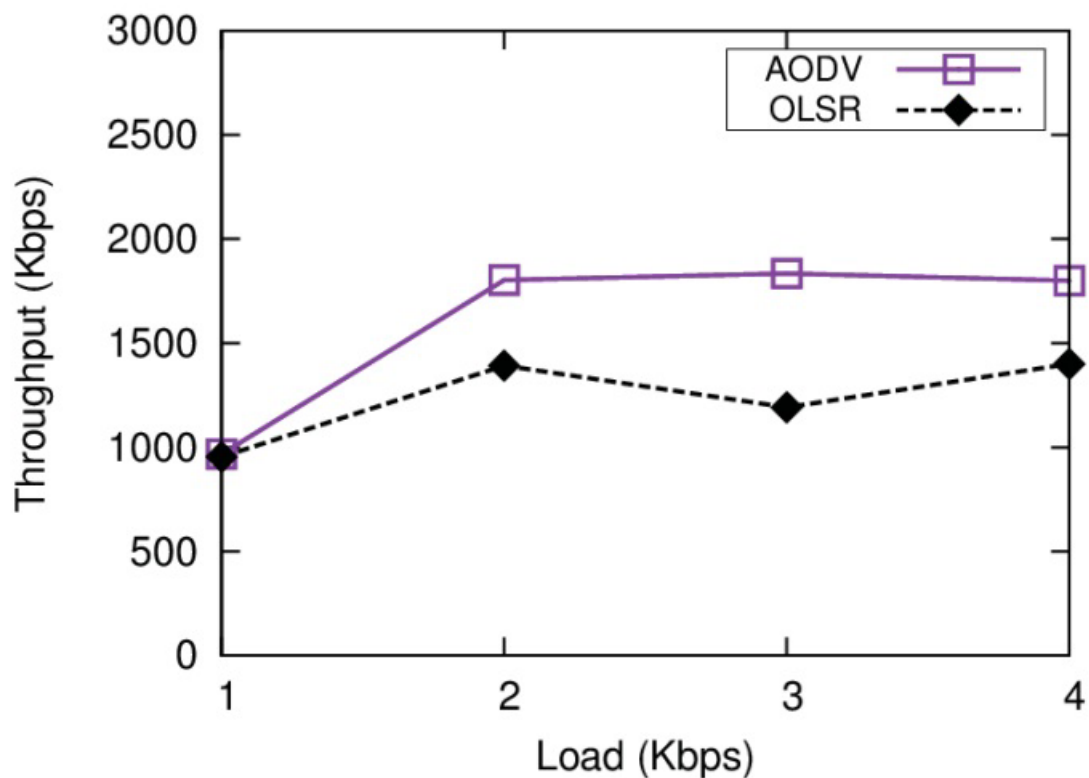


Figure 6.9: Routing Overhead Comparison for AODV and OLSR

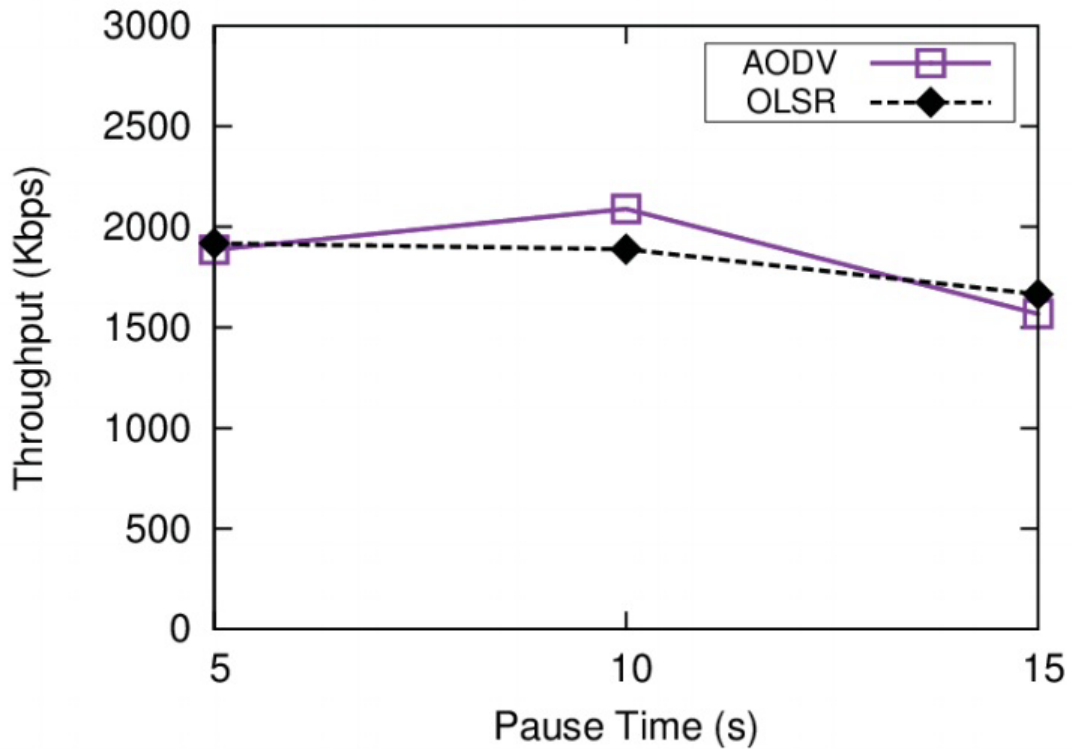
The biggest difference in terms of performance of the two protocols stems from the large difference in the routing overhead of the two protocols (figure 8-9). OLSR in general generates a larger overhead being a proactive protocol while AODV generates a smaller overhead as it creates routes only when required. It is also interesting to note that increasing the traffic has almost no impact on the routing overhead because the routing overhead is mainly dependent on the network size, which for this simulation remains constant.



**Figure 6.10: Throughput Comparison for AODV and OLSR**

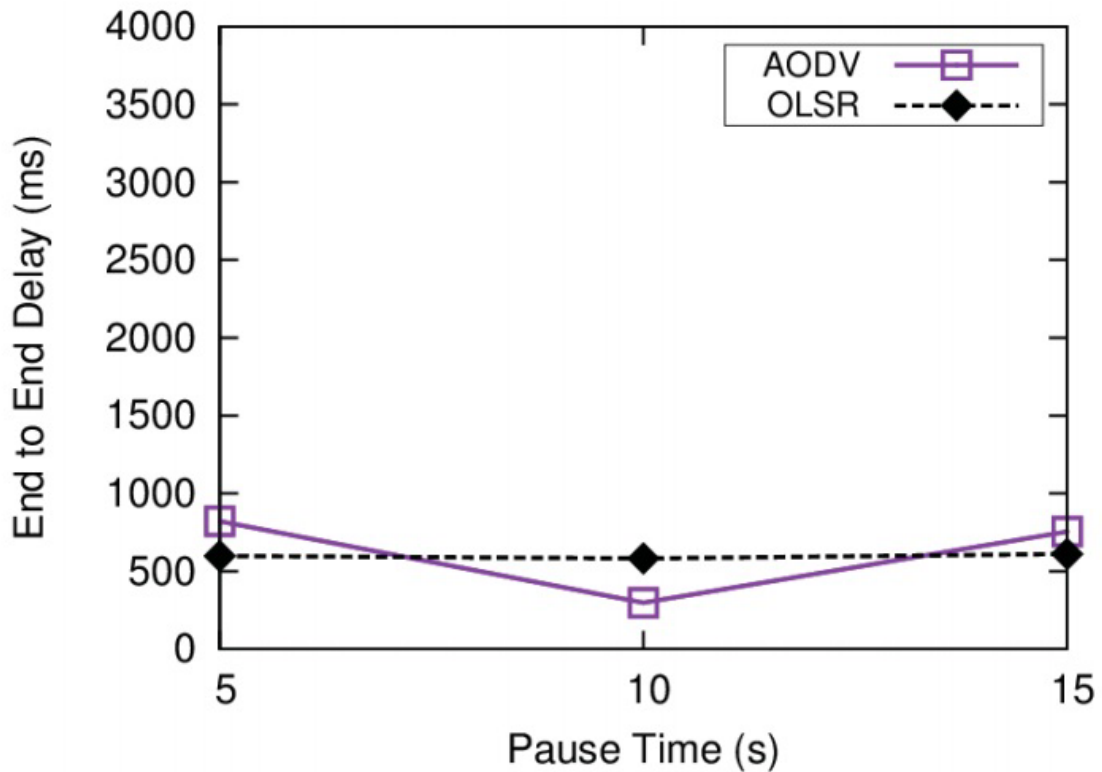
Similar to the results for Packet Delivery Rate, the throughput obtained with AODV is higher than that of OLSR mainly because of the problem of routing overhead and a higher collision rate in OLSR as the load increases (figure 6.10). Overall, for both protocols, the throughput increases as the amount of traffic injected in the network increases (figure 6.10).

### 6.3 Performance Comparison of AODV and OLSR for Varying Mobility



**Figure 6.11: Throughput Comparison for AODV and OLSR**

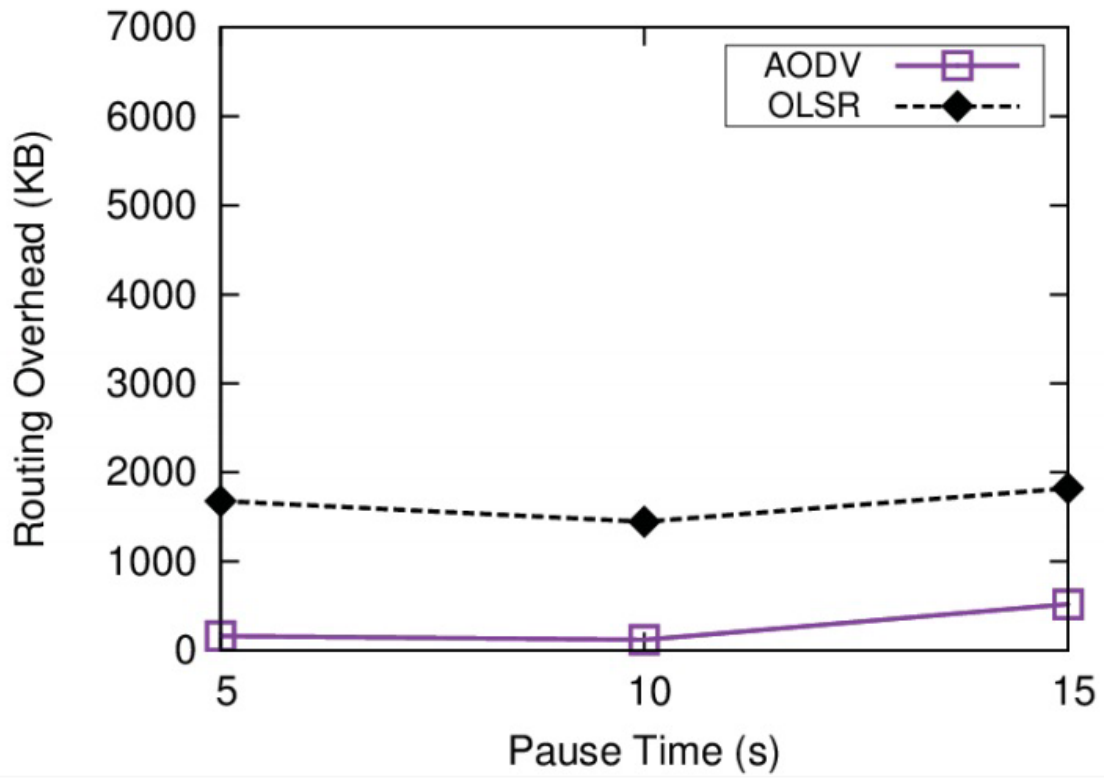
In terms of throughput, the two protocols show similar performance as the mobility rate is varied (pause time 5s to 15s) (figure 6.11). This is primarily because the two protocols differ significantly when the topology size changes, but for the case of mobility, the topology size is constant.



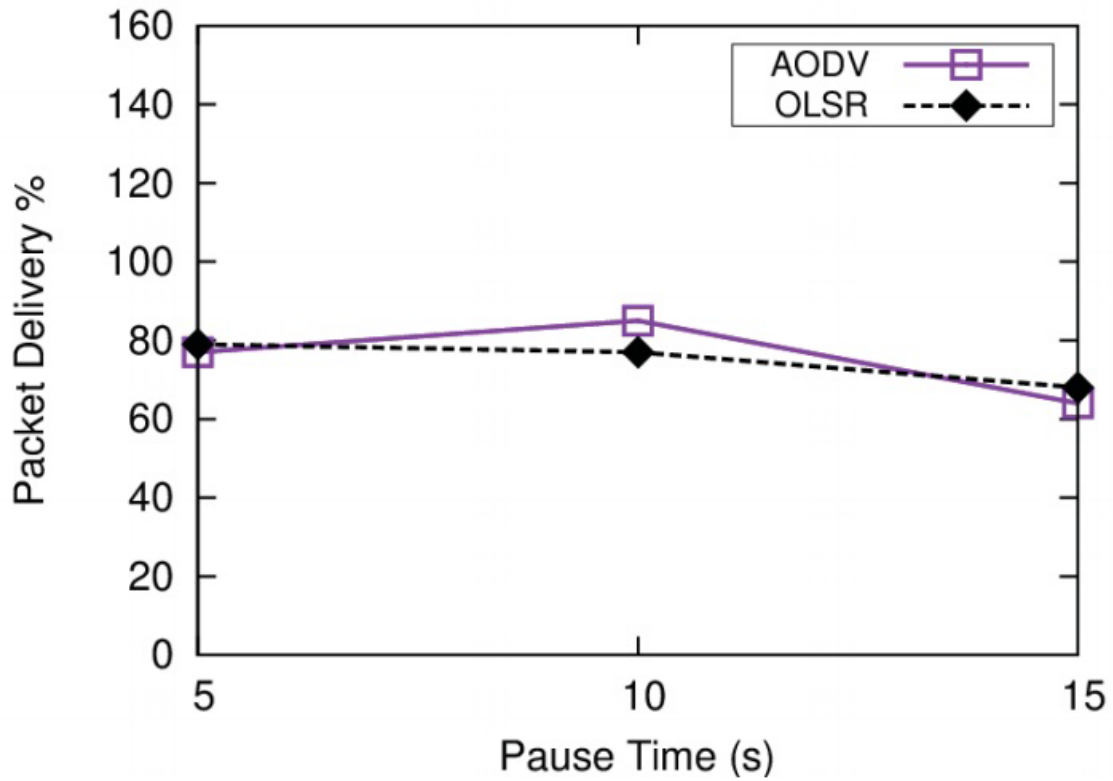
**Figure 6.12: End-to-End Delay Comparison for AODV and OLSR**

In terms of end-to-end delay, the delay remains more or less constant as the mobility is varied (figure 6.12). Both protocols are well equipped to handle mobility scenarios and therefore give acceptable performance.

In terms of routing overhead, the important point to note is that the routing overhead remains more or less constant for both the protocols with AODV giving a smaller routing overhead due to its reactive nature (figure 6.13). The overhead remains constant because it is mainly dependent on the network size and not on the mobility.

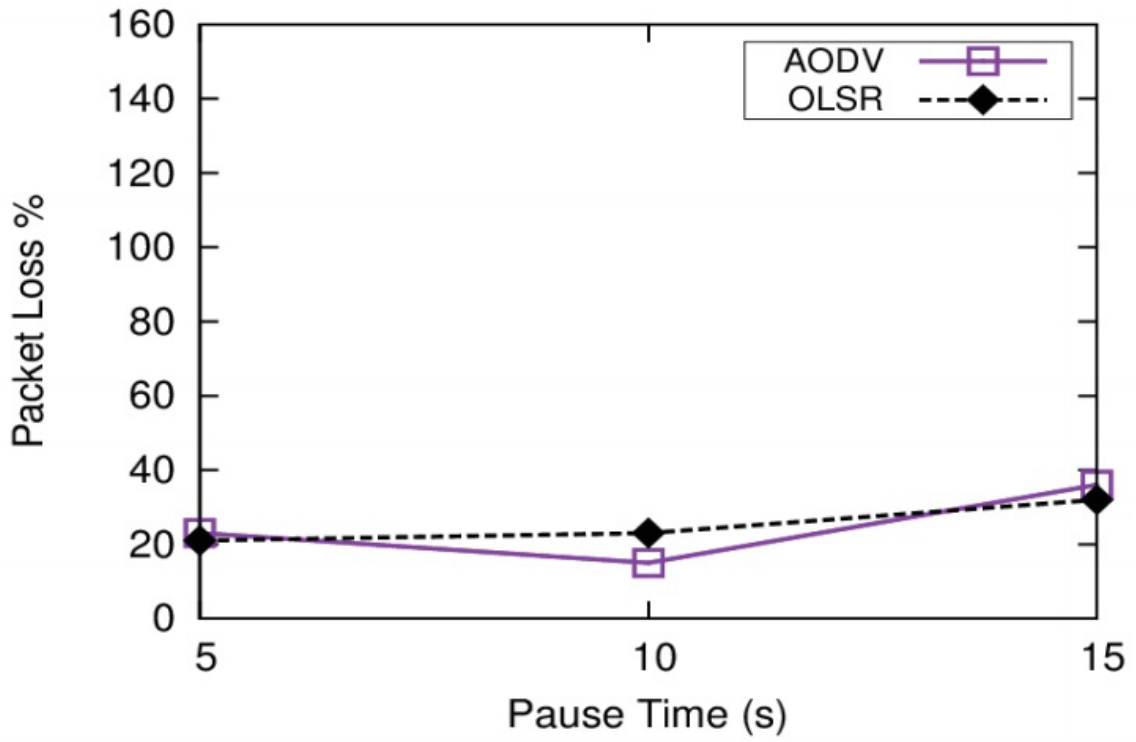


**Figure 6.13: Routing Overhead Comparison for AODV and OLSR**



**Figure 6.14: Packet Delivery Comparison for AODV and OLSR**

In terms of packet delivery and loss, again, both protocols perform more or less similarly because the topology size remains constant and hence, the number of routing packets remains more or less constant giving a constant and somewhat stable performance for both protocols (figure 6.14 and 6.15).



**Figure 6.15: Packet Loss Comparison for AODV and OLSR.**



## CHAPTER 7

### Conclusion and Future work

The aim of this work was to evaluate the performance of routing protocols AODV and OLSR. In this thesis, based on the results of simulation a comparative analysis was done between selected routing protocols AODV and OLSR and the results were documented. The performance has been evaluated based on parameters that aim to figure out the effects of routing protocols. By comparing these protocol performances, this work justifies that the AODV routing protocol performs better compared to OLSR in terms of: 1) End-to-end delay 2) Throughput 3) Packet loss 4) Packet delivery ratio 5) Routing overhead AODV is a reactive protocol and creates a very low routing overhead due to discovering routes only when needed, OLSR is proactive in nature. From the comparative analysis of routing protocols, the AODV outperforms the OLSR. The AODV has low load than OLSR respectively. From the above results 4-3, 4-4, 4-5, 4-6 and 4-7 the behavior of all the routing protocols in different number of mobile nodes, it can be seen that which routing protocol perform well. In terms of network size, mobility and traffic load AODV shows better results than OLSR. From the simulated results the behaviors of all routing protocols for different numbers of mobile nodes was observed and we came to the conclusion that AODV routing protocol performs well. The study of these routing protocols shows that the AODV is better in wireless ad-hoc network according to the simulation results but it is not necessary that AODV perform always better in all the networks. Its performance may vary by varying the network. At the end we came to the point that the performance of routing protocols vary with network size and selection of accurate routing protocols according to the network that ultimately influence the efficiency of that network in efficient way. Future work is about the development of modified version of the selected routing protocols, which should consider different aspects of routing protocols such as rate of higher route establishment with less route breakage and the weakness of the protocols mentioned should be improvised.

## References

- [1] S.Marti, T. Guiuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proc. 6th Annual ACM/IEEE Mobile Computing and Networking, Boston, MA, Aug.2000, pp.255-265
- [2] S. buchegger and J. -Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes-Fairness In Dynamic Ad-hoc Networks," Proc. 3<sup>rd</sup> IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, CH, 9-11 June 2002, pp.226-236
- [3] P. Michiardi and R. Molva, "CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," The 6th IFIP Conf. on Security Communications, and Multimedia, Porotoz, Slovenia, 2002.
- [4] K. Govindan and P. Mohapatra, "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey", IEEE Communications Surveys and Tutorials, 2012, pp.279-298
- [5] D.B.Jhonson and D.A.Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing, Kluwer Academic Publishers, vol.353, pp. 153-181, 1996
- [6] Y. Hu and A. Perrig, "A Survey of Secure Wireless Ad Hoc Routing", IEEE Security and Privacy, vol.2, no. 3, pp. 28-39, May 2004
- [7] Jin-Hee Cho, Anathram Swami, Ing-Ray Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks", IEEE Communications Survey & Tutorials, vol. 13, No. 4, FourthQuarter 2011
- [8] Patroklos G. Argyroudis and Donal O'Mahony, "Secure Routing for Mobile Adhoc Networks", IEEE Communications Surveys, vol.7, No. 4