

# **SIMULATION AND ANALYSIS OF AODV ROUTING PROTOCOL IN VANETS**

*Thesis submitted in partial fulfillment of the requirements for the award of  
degree of*

**Master of Engineering  
in  
Computer Science & Engineering**

*Submitted By*  
**Tajinder Kaur**  
**(Roll No. 801032029)**

Under the supervision of:

**Ms. Karamjit Cheema**  
Lecturer, CSED, Thapar University

**Dr. A. K. Verma**  
Assistant Professor, CSED, Thapar University



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT  
THAPAR UNIVERSITY  
PATIALA – 147004

**MAY 2012**

## Certificate

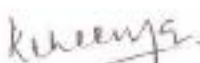
---

I hereby certify that the matter which is being presented in the thesis entitled, "SIMULATION AND ANALYSIS OF AODV ROUTING PROTOCOL IN VANETS", in partial fulfillment of the requirements for the award of degree of *Master of Engineering in Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. A. K. Verma* and *Ms. Karamjit Cheema* and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

  
(Tajinder Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

  
(Ms. Karamjit Cheema)  
Lecturer  
CSED  
Thapar University, Patiala

  
(Dr. A. K. Verma)  
Assistant Professor  
CSED  
Thapar University, Patiala

Countersigned by:

  
(Dr. Maninder Singh)  
Associate Professor & Head  
CSED  
Thapar University, Patiala.

  
(Dr. S. K. Mohapatra)  
Dean (Academic Affairs)  
Thapar University,  
Patiala.

## ***Acknowledgement***

---

*No volume of words is enough to express my gratitude towards my guide, **Dr. Anil Kumar Verma**, Assistant. Professor and **Ms. Karamjit Cheema**, Lecturer, Computer Science and Engineering Department, Thapar University, who have been very concerned and have supervised the work presented in this thesis report. They have helped me to explore this vast field in an organized manner and provided me with all the ideas on how to work towards a research oriented venture.*

*I am also thankful to **Dr. Maninder Singh**, Head of Department, CSED and **Mr. Karun Verma**, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.*

*I would also like to thank the staff members and my colleagues who were always there in the need of the hour and provided with all the help and facilities, which I required, for the completion of my thesis.*

*Most importantly, I would like to thank my **parents, friends** and the **Almighty** for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.*

**Tajinder Kaur**  
**(801032029)**

## ABSTRACT

---

Recent advances in wireless networks have led to the introduction of a new type of networks called Vehicular Networks. Vehicular Ad Hoc Network (VANET) is a form of Mobile Ad Hoc Networks (MANET). VANETs provide us with the infrastructure for developing new systems to enhance drivers' and passengers' safety and comfort. VANETs are distributed self organizing networks formed between moving vehicles equipped with wireless communication devices. This type of networks is developed as part of the Intelligent Transportation Systems (ITS) to bring significant improvement to the transportation systems performance.

As the necessity of such network increases the implementation challenges are being taken into account. They are broadcasting, routing, priority scheduling and security and privacy. In this paper the routing is considered as the research factor. Many routing protocols have been proposed and assessed to improve the efficiency of VANET. Previous studies of VANET showed that the Ad hoc On-demand Distance Vector (AODV) has the highest efficiency in topology-based protocols.

In this thesis simulation of AODV is done on simulators which allow users to generate real world mobility models for VANET simulations. For this MOVE is used along with SUMO and NS2. MOVE tool is built on top of SUMO which is open source micro-traffic simulator. Output of MOVE is a real world mobility model and can be used by network simulator NS-2. In this thesis we analyze and evaluate the performance of AODV under different node densities. Three different sets of node density would be used to compare the performance of the said protocol. Simulation results obtained in the form of graphs are then compared under various parameters like throughput, packet size, packet drop, End to End delay etc.

Keywords: AODV, VANET, MOVE, SUMO, NS2.

# TABLE OF CONTENTS

---

---

<b>Certificate.....</b>	<b>i</b>
<b>Acknowledgement.....</b>	<b>ii</b>
<b>Abstract.....</b>	<b>iii</b>
<b>Table of Contents.....</b>	<b>iv</b>
<b>List of Abbreviations.....</b>	<b>vii</b>
<b>List of Figures.....</b>	<b>ix</b>
<b>List of Tables .....</b>	<b>xi</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Networks.....	1
1.1.1 Wired Networks.....	1
1.1.2 Wireless Networks.....	1
1.2 TYPES OF WIRELESS NETWORKS.....	2
1.2.1 WPAN.....	2
1.2.2 Wireless Local Area Network (WLAN).....	3
1.2.3 Wireless Wide Area Network (WWAN).....	3
1.3 MANET.....	3
1.4 VANET– An Introduction.....	4
<b>CHAPTER 2: LITERATURE SURVEY.....</b>	<b>5</b>
2.1 Overview of VANET.....	5
2.1.1 Intelligent Transportation Systems (ITSs).....	5
2.1.2 Inter-Vehicle Communication .....	6
2.1.3 Vehicle-to-roadside Communication .....	6
2.1.4 Routing-based Communication .....	7
2.2 Standards for Wireless Access in VANETs .....	7
2.2.1 Dedicated Short Range Communication (DSRC).....	8

2.2.2 IEEE 1609 – Standards for WAVE (IEEE 802.11p).....	9
2.3 NETWORK ARCHITECTURE AND CHARACTERISTICS.....	9
2.4 Routing, QoS, Broadcasting, and Security in VANET.....	11
2.5 ROUTING PROTOCOLS .....	11
2.5.1 Topology-based Routing Protocols.....	12
2.5.2 Geographic Routing.....	15
2.6 The Mobility Model.....	16
2.6.1 Survey Models.....	17
2.6.2 Event driven models or Trace models.....	17
2.6.3 Software Oriented Models.....	17
2.6.4 Synthetic Model.....	17
2.6.5 Classification of Synthetic Mobility Model.....	18
2.7 AODV in Detail.....	20
2.7.1 Message Formats.....	20
2.7.2 Various Mechanisms.....	20
<b>CHAPTER 3: PROBLEM STATEMENT &amp; OBJECTIVE.....</b>	<b>26</b>
3.1 Problem Statement.....	26
3.2 Objective and Sub-Tasks.....	27
<b>CHAPTER 4: INSTALLATION, SIMULATION &amp; DESIGN.....</b>	<b>28</b>
4.1 NS2.....	29
4.1.1 Overview.....	29
4.1.2 Basic Architecture.....	30
4.1.3 Implementing languages.....	30
4.1.4 Tool Command Language (Tcl).....	30
4.1.5 The Network Animator (NAM).....	31
4.1.6 The Trace File.....	31
4.1.7 Installation Steps.....	32
4.2 SUMO.....	33
4.2.2 Features.....	33

4.2.3 SUMO GUI.....	34
4.2.1 Installation steps on Ubuntu.....	35
4.3 MOVE Simulator.....	35
4.3.1 Mobiliy Model Generation.....	36
4.3.2 Traffic Model Generation.....	44
4.4 Tracegraph.....	47
4.4.1 Installation Steps.....	47
<b>CHAPTER 5: RESULTS &amp; EVALUATION.....</b>	<b>49</b>
5.1 CASE 1: Throughput of sending packets.....	50
5.2 CASE 2: Throughput of receiving packets.....	52
5.3 CASE 3: Throughput of dropping packets.....	54
5.4 CASE 4: Packet Size vs Average throughput of sending packets.....	56
5.5 CASE 5: Packet Size vs Average throughput of receiving packets.....	57
5.6 CASE 6: Packet Size vs Average throughput of dropping packets.....	59
5.7 CASE 7: Throughput of sending bits vs Average simulation End2End delay....	61
5.8 CASE 8: Throughput of receiving bits vs Average simulation End2End delay...	63
<b>CHAPTER 6: CONCLUSION &amp; FUTURE SCOPE.....</b>	<b>65</b>
<b>ANNEXURES</b>	
<b>I. REFERENCES.....</b>	<b>67</b>
<b>II. LIST OF PUBLICATIONS .....</b>	<b>71</b>

## LIST OF ABBREVIATIONS

---

AODV	: Ad hoc On-demand Distance Vector
ARIB	: Association of Radio Industries and Businesses
ASK	: Amplitude Shift Keying
ASTM	: American Society for Testing and Materials
CEN	: European Committee for Standardization
DGPS	: Differential Global Positioning System
DSDV	: Destination-Sequenced Distance Vector
DSR	: Dynamic Source Routing
DSRC	: Dedicated Short Range Communication
DTN	: Delay Tolerant Network
FCC	: Federal Communications Commission
GPS	: Global Positioning System
MAC	: Media Access Control
MANET	: Mobile Ad Hoc Network
MOVE	: MObility model generator for VEhicular networks
NAM	: Network Animator
NS	: Network Simulator
OBU	: On-Board Unit
OFDM	: Orthogonal Frequency Division Multiplexing
OTcl	: Object-oriented Tool Command Language
PGB	: Preferred Group Broadcasting
PSK	: Phase Shift Keying
RERR	: Route Error
RREP	: Route Reply
RREQ	: Route Request
RSU	: Road Side Unit
SUMO	: Simulation of Urban MObility
Tcl	: Tool Command Language
TCP	: Transmission Control Protocol

TORA : Temporally Ordered Routing Algorithm  
UDP : User Datagram Protoco  
VANET : Vehicular Ad Hoc Network  
WAVE : Wireless Access in Vehicular Environments  
WLAN : Wireless Local Area Network  
WMAN : Wireless Metropolitan Area Network  
WPAN : Wireless Personal Area Network  
WWAN : Wireless Wide Area Network

## LIST OF FIGURES

---

Fig. 1.1: Types of Wireless Networks.....	2
Fig 1.2: VANET Scenario.....	4
Fig. 2.1: Inter-vehicle communication.....	6
Fig. 2.2: Vehicle-to-roadside communication.....	6
Fig. 2.3: Routing-based communication.....	7
Fig. 2.4: Three categories of VANET network architecture.....	9
Fig. 2.5: Taxonomy of Various Routing Protocols in VANET.....	12
Fig. 2.6: AODV route discovery.....	13
Fig. 2.7: Topologies.....	18
Fig. 2.8: RREQ Message Format.....	20
Fig.2.11 RREP Message Format.....	21
Fig.2.12 RERR Message Format.....	22
Fig2.13. Route discovery Cycle.....	23
Fig. 4.1: Basic architecture of NS2.....	30
Fig.4.2: NAM Visualization.....	31
Fig. 4.3: Fields of trace file.....	32
Fig.4.4: SUMO Visualization.....	34
Fig. 4.5: MOVE main menu.....	35
Fig. 4.6: This is the roadmap in the VANET manual.....	36
Fig. 4.7: Mobiliy Model Generation Main menu.....	37
Fig. 4.8: Map Nodes Editor.....	38
Fig. 4.9: Roads Editor.....	39
Fig. 4.10: Map Configuration Editor.....	39
Fig. 4.11: Vehicle Flow Definition Editor.....	40
Fig. 4.12: Junction Turning Probability Editor.....	41
Fig. 4.13: Automatic Vehicle Route Generator.....	42
Fig. 4.14: SUMO Visualization.....	43
Fig. 4.15: Traffic Model.....	43
Fig. 4.16: Static Traffic Model Generator for NS-2.....	44

Fig. 4.17 NAM Visualization.....	45
Fig. 4.18: Tracegraph.....	47
Fig. 4.19: Graph plotted by Tracegraph.....	48
Fig. 5.1: Throughput of sending packets for 4 nodes.....	50
Fig. 5.2: Throughput of sending packets for 10 nodes.....	51
Fig. 5.3: Throughput of sending packets for 25 nodes.....	52
Fig. 5.4: Throughput of receiving packets for 4 nodes.....	52
Fig. 5.5: Throughput of receiving packets for 10 nodes.....	53
Fig. 5.6: Throughput of receiving packets for 25 nodes.....	54
Fig. 5.7: Throughput of dropping packets for 4 nodes.....	54
Fig. 5.8: Throughput of dropping packets for 10 nodes.....	55
Fig. 5.9: Throughput of dropping packets for 25 nodes.....	55
Fig. 5.10: Packet Size vs Average throughput of sending packets for 4 nodes.....	56
Fig. 5.11: Packet Size vs Average throughput of sending packets for 10 nodes.....	56
Fig. 5.12: Packet Size vs Average throughput of sending packets for 25 nodes.....	57
Fig. 5.13: Packet Size vs Average throughput of receiving packets for 4 nodes.....	58
Fig. 5.14: Packet Size vs Average throughput of receiving packets for 10 nodes.....	58
Fig. 5.15: Packet Size vs Average throughput of receiving packets for 25 nodes.....	59
Fig. 5.16: Packet Size vs Average throughput of dropping packets for 4 nodes.....	59
Fig. 5.17: Packet Size vs Average throughput of dropping packets for 10 nodes.....	60
Fig. 5.18: Packet Size vs Average throughput of dropping packets for 25 nodes.....	60
Fig. 5.19: Throughput of sending bits vs Avg. simulation End2End delay for 4 nodes.....	61
Fig. 5.20: Throughput of sending bits vs Avg. simulation End2End delay for 10 nodes.....	61
Fig. 5.21: Throughput of sending bits vs Avg. simulation End2End delay for 25 nodes.....	62
Fig. 5.22: Throughput of receiving bits vs Avg. simulation End2End delay for 4 nodes.....	63
Fig. 5.23: Throughput of receiving bits vs Avg. simulation End2End delay for 10 nodes...	63
Fig. 5.24: Throughput of receiving bits vs Avg. simulation End2End delay for 25 nodes...	64

## LIST OF TABLES

---

Table 2.1: DSRC standards in Japan, Europe, and US.....	8
Table 2.2: RREQ Field Description.....	20
Table 2.3: RREP Field Description.....	21
Table 2.4: RERR Field Description.....	22
Table 4.1 System Configuration.....	28
Table 4.2: Simulation Configuration.....	45

### 1.1 Networks

Networks are widely used in both the business and consumer landscapes. A network involves a number of devices linked together to form a communications system for information and device sharing. Local Area Networks (LANs) are small, limited to about 500 meters, and are commonly deployed in corporate offices to facilitate low-cost, high-bandwidth information transfer within a company. Cities and other metropolitan regions can be connected via Metropolitan Area Networks or (MANs), and Wide Area Networks (WANs) involve systems communicating across large geographic regions such as states or countries. Globally, computers in networks interlink to form what is referred to as "the Internet."

Networks can be classified on many different bases like network layer (OSI, TCP/IP), scale (PAN, LAN, MAN, WAN), functional relationship (Client-Server, Peer to peer), topology (Bus, Star, Ring, Mesh) etc. On the basis of connection method, networks can be classified according to the technology used to connect the individual devices in the network that gives rise to **Wired or Wireless** networks.

#### 1.1.1 Wired Networks

Wired networks are the most commonly used networks all over the world. Wired LANs use Ethernet cables and network adapters. Although two computers can be directly wired to each other using a crossover cable, wired LANs generally also need central devices like hubs, switches, or routers to accommodate more computers.

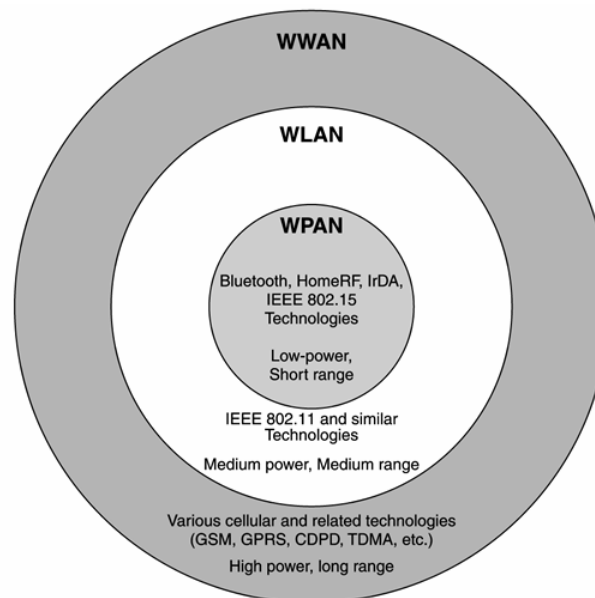
#### 1.1.2 Wireless Networks

Over the past few years, an evolution has taken place toward using networks without wires. As a result, the world has become increasingly mobile. In fact, today wireless interfaces are available to utilize network services that allow the people to use e-mail and access applications, and browse the Internet from just about anywhere. Wireless networks provide a great deal of convenience and flexibility, and are relatively easy to set up.

Wireless communication involves: Radio frequency communication, Microwave communication, for example long-range line-of-sight via highly directional antennas, or short-range communication, or Infrared (IR) short-range communication, for example from remote controls or via IRDA

## 1.2 Types of Wireless Networks

Wireless networks operate on the same hierarchy as their wired counterparts; small networks of three or more devices are referred to as Wireless LANs (WLANs), while the global wireless network is referred to as the wireless Internet. Other basic types of wireless networks include the Wireless Personal Area Network (WPAN), the Wireless Metropolitan Area Network (WMAN), and the Wireless Wide Area Network (WWAN) [1].



**Fig. 1.1: Types of Wireless Networks [2]**

### 1.2.1 WPAN

A wireless PAN, for example, might involve some user wirelessly synchronizing his PDA to a laptop or desktop computer. Likewise, a wireless PAN can provide wireless connectivity to a printer. The benefit of eliminating the tangle of wires when using computer peripherals in this fashion is extremely useful, and the initial installation and movement of peripherals is easy.

Example of WPAN: Bluetooth

### **1.2.2 Wireless Local Area Network (WLAN)**

Wireless LANs supply high performance within and around office buildings, factories, and homes. Users in these areas typically have laptops, PCs, and PDAs with large screens and processors that support higher-end applications. Wireless LANs efficiently satisfy connectivity requirements for these types of computer devices.

WLANs use electromagnetic waves (typically radio or infrared), to enable communication between devices in a limited area. Spread spectrum technology, based on radio transmission is most commonly used to deploy WLANs today.

### **1.2.3 Wireless Wide Area Network (WWAN)**

Wireless WANs offer mobile applications covering a large area, such as a country or continent. Because of economies of scale, a telecommunications operator can feasibly deploy the relatively expensive wireless WAN infrastructure to provide long-range connectivity for a large customer base. The costs such as deployment can be spread across many users, resulting in low subscriber fees.

## **1.3 MANET**

The increasing demand of wireless communication and the needs of new wireless devices have tend to research on self organizing, self healing networks without the interference of centralized or pre-established infrastructure/authority. The networks with the absence of any centralized or pre-established infrastructure are called Ad hoc networks. Ad hoc Networks are collection of self-governing mobile nodes [3]

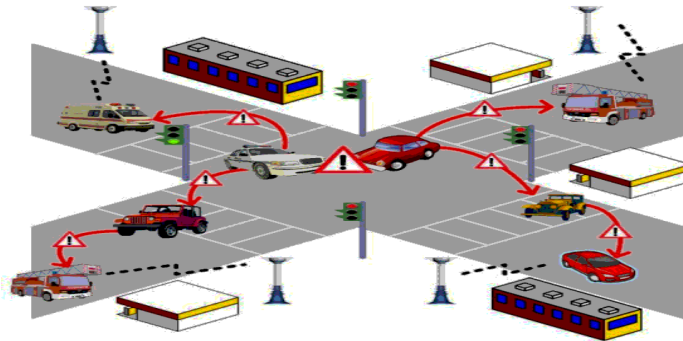
Wireless communication is ubiquitous because of its flexibility to adapt to different scenarios. Mobile Ad Hoc Networks (MANETs) is a term coined for the continuously varying network topology handheld mobiles devices. A Mobile Ad Hoc Network (MANET) is a network consisting of a collection of nodes capable of communicating with each other without help from a network infrastructure. Applications of MANETs include the battlefield applications, rescue work, as well as civilian applications like an outdoor meeting, or an ad-hoc classroom

## 1.4 VANET– An Introduction

Traffic congestion on the roads has become a large problem in big cities. The congestion and related vehicle accommodation problem is accompanied by a constant threat of accidents as well. Absence of road traffic safety takes a toll of precious human lives and poses a dire threat to our environment as well. Other negative consequences are related to energy waste and environmental pollution. According to National Highway Traffic Safety Administration, the following figures indicate some of the consequences of recent car accidents [4].

- 6.3 million Police reported traffic accidents
- 43,000 people were killed
- Millions of people were injured
- The economy effects caused due to these accidents were more than \$230 billion

Various precautions like seat belts and airbags are used but they cannot eliminate problems due to driver's inability to foresee the situation ahead of time .On a highway a vehicle cannot currently predict the speed of other vehicles. However, with use of sensor, computer and wireless communication equipment, speed could be predicted and a warning message sent every 0.5 seconds could limit the risk of potential accidents [5].



**Fig. 1.2: VANET scenario [6]**

VANET is one of the influencing areas for the improvement of Intelligent Transportation System (ITS) in order to provide safety and comfort to the road users. Vehicular Ad Hoc Networks (VANETs) is the subclass of MANET. It deploys the concept of continuously varying vehicular motion. These are self organized and distributed network, where fast moving vehicles have fixed movement along some traffic path. Fig. 1.2 shows a VANET scenario [6].

### 2.1 Overview of VANET

VANET assists vehicle drivers to communicate and to coordinate among themselves in order to avoid any critical situation through Vehicle to Vehicle communication e.g. road side accidents, traffic jams, speed control, free passage of emergency vehicles and unseen obstacles etc. Besides safety applications VANET also provide comfort applications to the road users e.g. weather information, mobile e-commerce, internet access and other multimedia applications. The most well known applications include, “*Advance Driver Assistance Systems (ADASE2)*, *Crash Avoidance Matrices Partnership (CAMP)*, *CARTALK2000* and *Fleet Net*” that were developed under collaboration of various governments and major car manufacturers [7].

#### 2.1.1 Intelligent Transportation Systems (ITSs)

Every vehicle in ITSs takes on the role of sender, receiver, and router [8] to broadcast information to the vehicular network or transportation agency, which then uses the information to ensure safe, free-flow of traffic. Vehicles must be equipped with some sort of radio interface or OBU that enables short-range wireless ad hoc networks to be formed for communication to occur between vehicles and RSUs. Vehicles must also be fitted with hardware that permits detailed position information such as GPS or a DGPS receiver. Fixed RSUs must be in place to facilitate communication. The number and distribution of roadside units is dependent on the communication protocol is to be used.

The possible communication configurations in ITSs include inter-vehicle, vehicle-to-roadside, and routing-based communications. It is worth mentioning that Intra-vehicle communication uses technologies such as IEEE 802.15.1 (Bluetooth), IEEE 802.15.3 (Ultra-wide Band) and IEEE 802.15.4 (Zigbee) that can be used to support wireless communication inside a vehicle but this is outside the scope of this paper and will not be discussed further.

### 2.1.2 Inter-Vehicle Communication

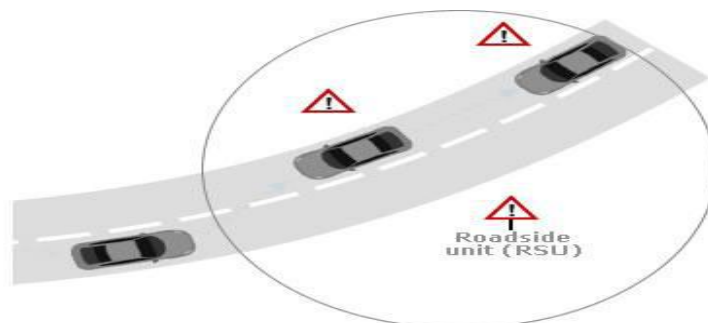
The inter-vehicle communication configuration (Fig. 2.1) uses multi-hop multicast/broadcast to transmit traffic related information over multiple hops to a group of receivers. In ITSs, vehicles need only be concerned with activity on the road ahead and not behind. There are two types of message forwarding in inter-vehicle communications: naïve broadcasting and intelligent broadcasting. In naïve broadcasting, vehicles send broadcast messages periodically and at regular intervals. Intelligent broadcasting with implicit acknowledgement addresses the problems inherent in naïve broadcasting by limiting the number of messages broadcast for a given emergency event.



**Fig. 2.1: Inter-vehicle communication [9]**

### 2.1.3 Vehicle-to-roadside Communication

The vehicle-to-roadside communication configuration (Fig. 2.2) represents a single hop broadcast where the roadside unit sends a broadcast message to all equipped vehicles in the vicinity.



**Fig. 2.2: Vehicle-to-roadside communication [9]**

This communication configuration provides a high bandwidth link between vehicles and roadside units. The roadside units may be placed every kilometer or less, enabling high data rates to be maintained in heavy traffic. The roadside unit will periodically broadcast a message containing the speed limit and will compare any geographic or directional limits with vehicle data to determine if a speed limit warning applies to any of the vehicles in the vicinity. If a vehicle violates the desired speed limit, a broadcast will be delivered to the vehicle in the form of an auditory or visual warning, requesting that the driver reduce his speed.

#### 2.1.4 Routing-based Communication

The routing-based communication configuration (Fig. 2.3) is a multi-hop unicast where a message is propagated in a multi-hop fashion until the vehicle carrying the desired data is reached. When the query is received by a vehicle owning the desired piece of information, the application at that vehicle immediately sends a unicast message containing the information to the vehicle it received the request from, which is then charged with the task of forwarding it towards the query source

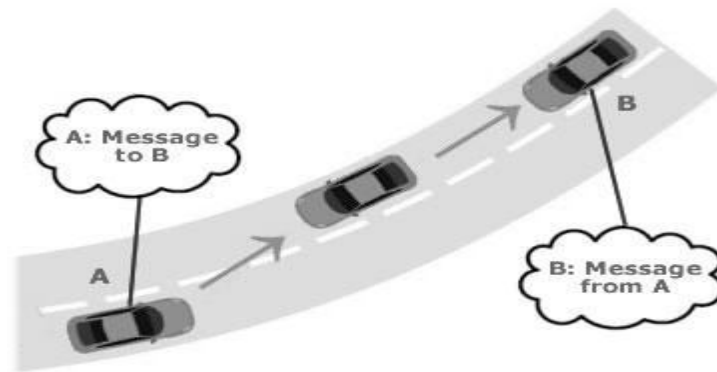


Fig. 2.3: Routing-based communication [9]

## 2.2 Standards for Wireless Access in VANETs

Standards simplify product development, help reduce costs, and enable users to compare competing products. There are many standards that relate to wireless access in vehicular environments.

### 2.2.1 Dedicated Short Range Communication (DSRC)

DSRC is a short to medium range communications service that was developed to support vehicle-to-vehicle and vehicle-to-roadside communications. Such communications cover a wide range of applications, including vehicle-to-vehicle safety messages, traffic information, toll collection, drive-through payment, and several others. DSRC is aimed at providing high data transfers and low communication latency in small communication zones. In 1999, the United States FCC allocated 75 MHz of spectrum at 5.9 GHz to be used by DSRC. In 2003, The ASTM approved the ASTM-DSRC standard which was based on the IEEE 802.11a physical layer and 802.11 MAC layer [10]. This standard was later published as ASTM E2213-03. In February 2004, the report issued by the FCC established service and licensing rules that govern the use of the DSRC band. DSRC is a free but licensed spectrum. The DSRC standards in Japan, Europe, and US are shown in Table 2.1[11].

**Table 2.1: Dedicated Short Range Communication standards in Japan, Europe & US[11]**

<b>Features</b>	<b>JAPAN (ARIB)</b>	<b>EUROPE (CEN)</b>	<b>USA (ASTM)</b>
<i>Communication</i>	Half-duplex (OBU)/Full duplex (RSU)	Half-duplex	Half-duplex
<i>Radio Frequency Band</i>	5.8 GHz band 80 MHz bandwidth	5.8 GHz band 20 MHz bandwidth	5.9 GHz band 75 MHz bandwidth
<i>Channels</i>	Downlink: 7 Uplink: 7	4	7 S
<i>Channel Separation</i>	5 MHz	5 MHz	10 MHz
<i>Data Transmission rate</i>	Down/Up-link 1 or 4 Mbits/s	Down-link/500 Kbits/s Up-link/250 Kbits/s	Down/Up-link 3-27 Mbits/s
<i>Coverage</i>	30 meters	15-20 meters	1000 meters (max)
<i>Modulation</i>	2- ASK, 4-PSK	RSU: 2-ASK OBU: 2-PSK	OFDM

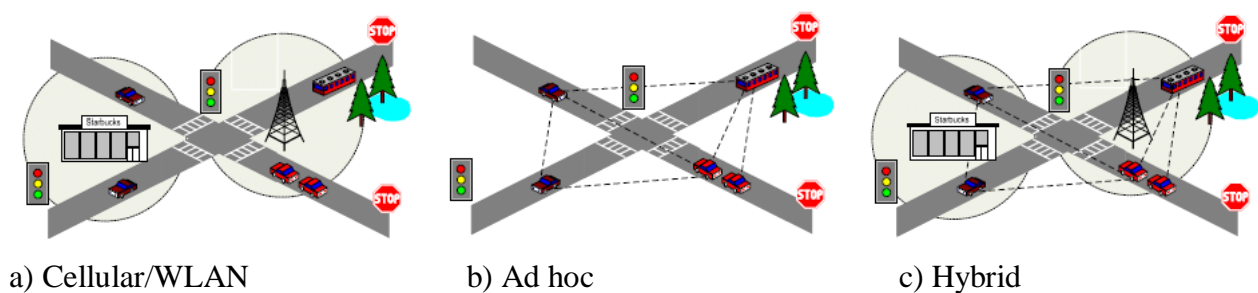
### 2.2.2 IEEE 1609 – Standards for Wireless Access in Vehicular Environments (WAVE) (IEEE 802.11p)

Wireless connectivity between moving vehicles can be provided by existing 802.11a compliant devices with data rates of up to 54 Mbps being achieved with 802.11a hardware [12]. However, vehicular traffic scenarios have greater challenges than fixed wireless networks, caused by varying driving speeds, traffic patterns, and driving environments. Traditional IEEE 802.11 Media Access Control (MAC) operations suffer from significant overheads when used in vehicular scenarios. For instance, to ensure timely vehicular safety communications, fast data exchanges are required. In these circumstances the scanning of channels for beacons from an access point along with multiple handshakes required to establish communication are associated with too much complexity and high overheads. To address these challenging requirements of IEEE MAC operations, the DSRC effort of the ASTM 2313 working group migrated to the IEEE 802.11 standard group which renamed the DSRC to IEEE 802.11p WAVE [13].

In contrast to the regional standards of DSRC, by incorporating DSRC into IEEE 802.11, WAVE will become a standard that can be universally adopted across the world.

### 2.3 VANET Architecture and Characteristics

According to Fig. 2.4 [14], the architecture of VANETs falls within three categories: pure cellular/WLAN, pure ad hoc, and hybrid.



**Fig. 2.4: Three categories of VANET network architecture [14]**

In pure cellular/WLAN architecture, the network uses cellular gateways and WLAN access points to connect to the Internet and facilitate vehicular applications. Vehicles communicate with the Internet by driving by either a cellular tower or a wireless access point.

Since the infrastructure of cellular towers and wireless access points are not necessarily widely deployed due to costs or geographic limitations, nodes may only engage in communication with each other. Information collected from sensors on a vehicle can become valuable in notifying other vehicles about traffic condition and helping the police solve crimes [15]. The infrastructure-less network architecture is in the pure ad hoc category where nodes perform vehicle-to-vehicle communication with each other.

When there are roadside communication units such as a cellular tower and an access point and vehicles are equipped with wireless networking devices, vehicles can take advantage of the infrastructure in communicating with each other. Various applications in areas of urban monitoring, safety, driving assistance, and entertainment [15] have used infrastructure communicating units to access dynamic and rich information outside their network context and share this information in a peer-to-peer fashion through ad hoc, infrastructureless communication. The hybrid architecture of cellular/WLAN and ad hoc approaches provides richer contents and greater flexibility in content sharing. VANETs possess a few distinguishing characteristics from MANETs. These are:

- a) **Highly dynamic topology.** Since vehicles are moving at high speed, the topology formed by VANETs is always changing. On highways, vehicles are moving at the speed of 60 mph (25 m/sec). The highly dynamic topology results in frequently disconnected network since the link between two vehicles can quickly disappear while the two nodes are transmitting information. The problem is further exacerbated by heterogeneous node density where frequently traveled roads have more cars than non-frequently traveled roads.
- b) **Patterned Mobility.** Vehicles follow a certain mobility pattern that is a function of the underlying roads, the traffic lights, the speed limit, traffic condition, and drivers' driving behaviors. Because of the particular mobility pattern, evaluation of VANET routing protocols only makes sense from traces obtained from the pattern.
- c) **Propagation Model.** In VANETs, the propagation model is usually not assumed to be free space because of the presence of buildings, trees, and other vehicles. A VANET propagation model should well consider the effects of free standing objects

as well as potential interference of wireless communication from other vehicles or widely deployed personal access points.

- d) Unlimited Battery Power and Storage.** Nodes in VANETs are not subject to power and storage limitation as in sensor networks, another class of ad hoc networks where nodes are mostly static. Nodes are assumed to have ample energy and computing power. Therefore, optimizing duty cycle is not as relevant as it is in sensor networks.
- e) On-board Sensors.** Nodes are assumed to be equipped with sensors to provide information useful for routing purposes. Many VANET routing protocols have assumed the availability of GPS unit from on-board Navigation system. Location information from GPS unit and speed from speedometer provides good examples for plethora of information to be utilized to enhance routing decisions.

## **2.4 Routing, QoS, Broadcasting, and Security in VANETs**

The field of VANETs started gaining attention after 1980s and has now been an active field of research and development. In fact, VANET research has attracted a lot of attention from researchers working in various fields including electronics, networking, security, software engineering, automotive, transportation, and so on. Recent results covering VANET-related issues include areas such as routing, Quality Service (QoS), broadcasting, security attacks and threats, capacity, collision and interference, the effects of transmission power on protocol performance and power control algorithms, congestion control, and service discovery. It is beyond the scope of this work to review each of these topics, so we concentrate only routing issue.

## **2.5 Routing Protocols**

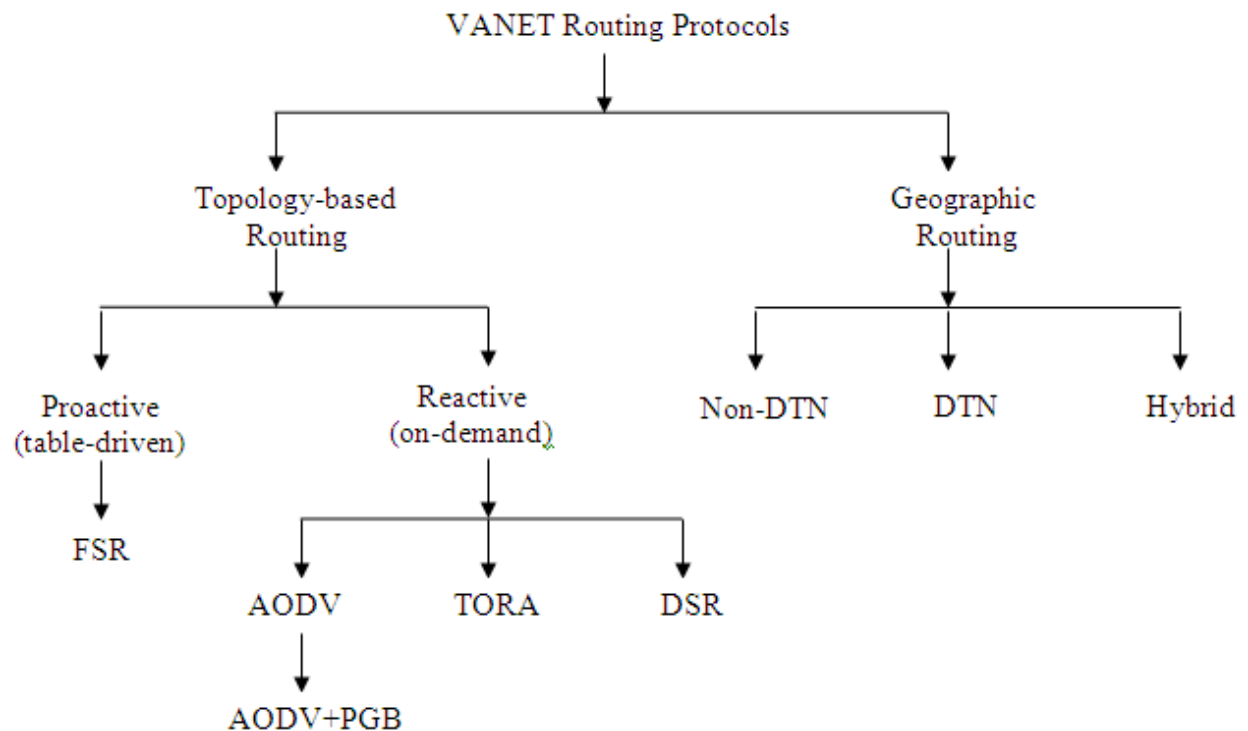
A routing protocol governs the way that two communication entities exchange information; it includes the procedure in establishing a route, decision in forwarding, and action in maintaining the route or recovering from routing failure.

There are number of MANET routing protocols but we will restrict our attention only to those MANET routing protocols which can be used in the VANET context. Fig. 2.5

illustrates the taxonomy of these VANET routing protocols which can be classified as topology-based and geographic (position-based).

### 2.5.1 Topology-based Routing Protocols:

These routing protocols use links' information that exists in the network to perform packet forwarding. They can further be divided into proactive (table-driven) and reactive (on-demand) routing.



**Fig. 2.5: Taxonomy of Various Routing Protocols in VANET**

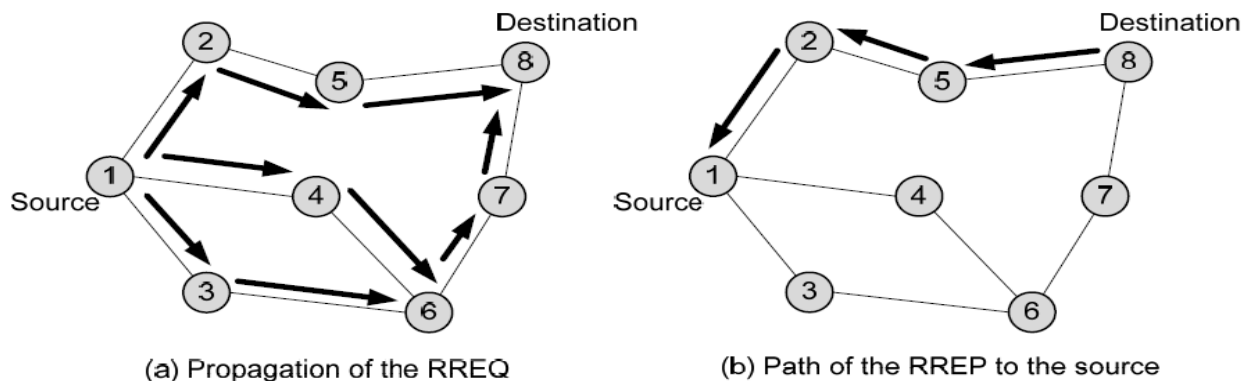
**a) Proactive (table-driven):** Proactive routing carries the distinct feature that the routing information such as the next forwarding hop is maintained in the background regardless of communication requests. Control packets are constantly broadcast and flooded among nodes to maintain the paths or the link states between any pair of nodes even though some of paths are never used. A table is then constructed within a node such that each entry in the table indicates the next hop node toward a certain destination. The advantage of the proactive routing protocols is that there is no route discovery since route to the destination is maintained in the background and is always available upon lookup. Despite its good property

of providing low latency for real-time applications, the maintenance of unused paths occupies a significant part of the available bandwidth, especially in highly mobile VANETs.

- **Fisheye State Routing (FSR) [16]** is an efficient link state routing that maintains a topology map at each node and propagates link state updates with only immediate neighbors not the entire network. Furthermore, the link state information is broadcast in different frequencies for different entries depending on their hop distance to the current node. Entries that are further away are broadcast with lower frequency than ones that are closer. The reduction in broadcast overhead is traded for the imprecision in routing. However, the imprecision gets corrected as packets approach progressively closer to the destination.

**b) Reactive (On Demand):** Reactive routing opens a route only when it is necessary for a node to communicate with another node. It maintains only the routes that are currently in use, thereby reducing the burden on the network. Reactive routings typically have a route discovery phase where query packets are flooded into the network in search of a path. The phase completes when a route is found.

- **AODV – In Ad Hoc On Demand Distance Vector (AODV) [17]** routing, upon receipt of a broadcast query (RREQ), nodes record the address of the node sending the query in their routing table (Fig. 2.6a). This procedure of recording its previous hop is called *backward learning*. Upon arriving at the destination, a reply packet (RREP) is then sent through the complete path obtained from backward learning to the source (Fig. 2.6b).



**Fig. 2.6: AODV route discovery [18]**

At each stop of the path, the node would record its previous hop, thus establishing the *forward path* from the source. The flooding of query and sending of reply establish a full duplex path. After the path has been established, it is maintained as long as the source uses it. A link failure will be reported recursively to the source and will in turn trigger another query-response procedure to find a new route. A detailed description of AODV appears in section 2.7, later in this thesis.

- **AODV+PGB** – Preferred Group Broadcasting (PGB) [19] is a broadcasting mechanism that aims to reduce broadcast overhead associated with AODV's route discovery and to provide route stability especially important in VANETs where fast moving vehicles are used as wireless hosts. Based on the received signal of the broadcast, receivers can determine whether they are in the preferred group and which one in the group to broadcast. Since only one node is allowed to broadcast and since the preferred group is not necessarily the one that makes the most progress towards the destination, route discovery might take longer than before. Another drawback is that broadcast can discontinue if the group is found to be empty (possibly because of sparse networks). Packet duplication can happen as two nodes in the preferred group can broadcast at the *same* time.
  
- **DSR** – Dynamic Source Routing (DSR) [20] uses *source routing*, that is, the source indicates in a data packet's the sequence of intermediate nodes on the routing path. In DSR, the query packet copies in its header the IDs of the intermediate nodes that it has traversed. The destination then retrieves the entire path from the query packet, and uses it to respond to the source. As a result, the source can establish a path to the destination. If the destination is allowed to send multiple route replies, the source node may receive and store multiple routes from the destination. An alternative route can be used when some link in the current route breaks. In a network with low mobility, this is advantageous over AODV since the alternative route can be tried before DSR initiates another flood for route discovery. There are two major differences between AODV and DSR. The first is that in AODV data packets carry

the destination address, whereas in DSR, data packets carry the full routing information. This means that DSR has potentially more routing overheads than AODV. Furthermore, as the network diameter increases, the amount of overhead in the data packet will continue to increase. The second difference is that in AODV, route reply packets carry the destination address and the sequence number, whereas, in DSR, route reply packets carry the address of each node along the route.

- **TORA** – Temporally Ordered Routing Algorithm (TORA) [21] routing belongs to a family of link reversal routing algorithms where a directed acyclic graph (DAG) toward the destination is built based on the height of the tree rooted at the source. The directed acyclic graph directs the flow of packets and ensures reachability to all nodes. When a node has a packet to send, it broadcasts the packet. Its neighbor only broadcasts the packet if it is the sending node's downward link based on the DAG. A node would construct the directed graph by broadcasting a query packet. Upon receiving a query packet, if a node has a downward link to the destination, it will broadcast a reply packet; otherwise, it simply drops the packet. A node, upon receiving a reply packet, will update its height only if the height from the reply packet gives the minimum of all the heights from reply packets it has received so far. It then rebroadcasts the reply packet.

The advantages of TORA are that the execution of the algorithm gives a route to *all* the nodes in the network and that it has reduced far-reaching control messages to a set of neighboring nodes. However, because it provides a route to all the nodes in the network, maintenance of these routes can be overwhelmingly heavy, especially in highly dynamic VANETs.

**2.5.2 Geographic (Position-based) Routing:** In geographic (position-based) routing, the forwarding decision by a node is primarily made based on the position of a packet's destination and the position of the node's one-hop neighbors. The position of the destination is stored in the header of the packet by the source. The position of the node's one-hop neighbors is obtained by the beacons sent periodically with random jitter (to prevent

collision). Nodes that are within a node's radio range will become neighbors of the node. Geographic routing assumes each node knows its location, and the sending node knows the receiving node's location by the increasing popularity of Global Position System (GPS) unit from an onboard Navigation System and the recent research on location services [22], respectively. Since geographic routing protocols do not exchange link state information and do not maintain established routes like proactive and reactive topology-based routings do, they are more robust and promising to the highly dynamic environments like VANETs.

Fig. 2.5 sub-classifies Geographic routing into three categories of non-Delay Tolerant Network (non-DTN), Delay Tolerant Network (DTN), and hybrid. The non-DTN types of geographic routing protocols do not consider intermittent connectivity and are only practical in densely populated VANETs whereas DTN types of geographic routing protocols do consider disconnectivity. However, they are designed from the perspective that networks are disconnected by default. Hybrid types of geographic routing protocols combine the non-DTN and DTN routing protocols to exploit partial network connectivity. But it's difficult to explain all these protocols in this thesis, so we will not go into details of these.

## **2.6 The Mobility Model**

To attain good results from VANET simulation, it is important to generate a realistic mobility model that is as realistic as real ad-hoc network communication. The usage of mobility model signifies the movement of mobile node that will consume the protocol. Here we will further explore the mobility model, their different types and classify them according to the level of details they generate.

The Mobility Model governs the set of rules that define movement pattern of nodes in ad-hoc network. Network simulators can then, by using this information, create random topologies based on nodes position and perform some tasks between the nodes.

While it is hard to present real world traffic scenarios in a single simulation model, ways can be adopted to develop a protocol suite which can support the implementation. The mobility patterns can be generated from various models. These models [23] are described below.

### **2.6.1 Survey Models**

Survey models represent realistic human behaviour in urban mesh environments. The model relies on data collected through surveys performed on human activities. The survey was recorded for the human performance, tasks, and activities. E.g. UDel mobility model is a tool for simulating urban mesh networks.

### **2.6.2 Event driven models or Trace models**

These models can be used monitor the movement of human beings and vehicles, analysing them and generating traces based on their locomotion. The problems in such models are correlation of different traces that were developed for specific purpose and also that there are limited simulators available for such purposes e.g. ELDA (Event Driven Light-weight Distilled State Charts-based Agents) model is an event driven mobility model based on MAO models (Mobile Active Object) and MAS models (Multi Agent Systems) deriving characteristics of behavioural, interaction and mobility models.

### **2.6.3 Software Oriented Models**

Various simulators like VISIM , CORSIM and TRANSIM are able to generate the traces of urban microscopic traffic. VANETMobiSim uses TIGER database and Voronoi graphs to extract road topologies, maps, streets etc for the network simulators. The problems with such simulators are that they can only operate at traffic level and cannot generate realistic levels of details. Moreover the inter-operability with network simulators and the generated level of details seems insufficient for network simulators

### **2.6.4 Synthetic Model**

The majority of the work has been carried out in the area of synthetic modelling. All models in this category use mathematical equations to develop realistic mobility models. The strength of the mathematical models is validated by comparing them with real mobility models.

Synthetic models can be further divided into 5 main categories.

1. Stochastic model: deals with totally random motion.

2. Traffic Stream model: examines the mechanical properties of mobility model.
3. Car Following model: monitors the behaviour of car-to-car interaction.
4. Queue model: considers cars as standing in queues and roads as queue buffers.
5. Behavioural models: examines how movement is influenced by social interaction.

For example, if we consider a mobile node in an area and observe its movement, it can either move in a fixed line or could follow a random path. The WWP (Weighted way point: Destination is chosen on the basis of current location and time) and RWP (Random way point: Destination is chosen randomly) mobility algorithm calculate mobility pattern of a node by defining certain mathematical equations. The synthetic model imposes certain limitations such as excluding a real human behaviour model hence restricting its abilities to create random topologies.

**2.6.5 Classification of Synthetic Mobility Model:** In this section, we will classify current synthetic mobility models according to the level of information they generate.

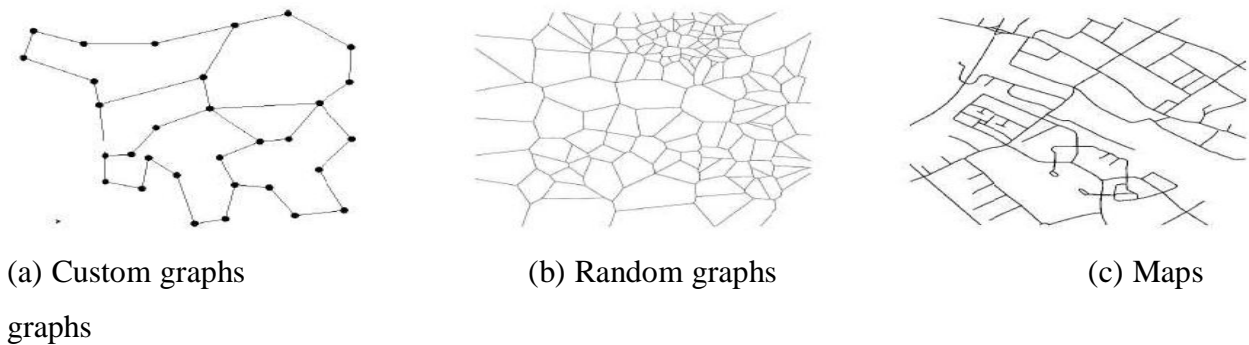
The Synthetic mobility model is classified as below:

- a) Traffic Level Criteria
- b) Motion Level Criteria
- c) Other Criteria

**a) Traffic Level Criteria**

The traffic level presents level of details that are concerned with streets, obstruction in communication paths, lights and vehicular densities. For the simulation to capture details at traffic level, it must include the following traces:

**1. Movement topologies:** Movement topologies are key features for simulation and are used to calculate some important factors like speed and distances etc.



**Fig. 2.7: Topologies [23]**

The topologies are represented with the help of graphs and are classified into the following three types: Custom graphs: Edges are connected by vertex. (Fig.2.7a). Random graphs: Using algorithms. (Fig.2.7b). Topologies from maps: Graphs from GDF (Geographical Data Files) and TIGER database. (Fig. 2.7c)[23].

**2. Start and End position:** The time a node starts its movement marks its initial position is referred to as repelling state, as the node traverses a certain path until it reaches its final position which can be referred to as its attracting point. These two points outline the start and end point for the vehicle. After the graphs are generated, the node's source and destination points are defined for simulation.

**3. Trip Creation:** During simulation, the vehicle navigates through different points. These different points are called trip for vehicle.

**4. Selection of Track:** The algorithms define the track between paths.

**5. Speed of Vehicles:** The speed of the vehicle depends on the road conditions and can be either smooth or arbitrary.

#### **b) Motion Level Criteria**

After the details captured at the traffic level, the motion level comes into playing its part by creating topologies between the nodes and analyzing their behaviour based on the details gathered at traffic level e.g a car may change its lane and try to overtake. It also monitors the situation during heavy traffic flow or vehicles standing in queue following each other. Motion level feature also defines human behaviour patterns through their movement which aids in finding vehicular behaviour .Such models are commonly adopted from mathematical equations which produce all possible vehicular behaviour patterns. There are various models that fall under this category .The most widely used model is the “car following model”.

- **Car following model:** This model describes the process of vehicles following each other in the similar line. *Carfollowing model* is one of the widely used model that present details at motion level. It describes the process of car following each other in the same lane. This model been preferred over other traffic model like *Krauss Model (KM)*, *General Motors Model (GM)*, *Gipps Model (GP)*, *Intelligent Driver Model (IDM)*.

c) **Other criteria:** Besides that, a good simulator should have an easy to use feature. All these features have been compiled according to this criteria.

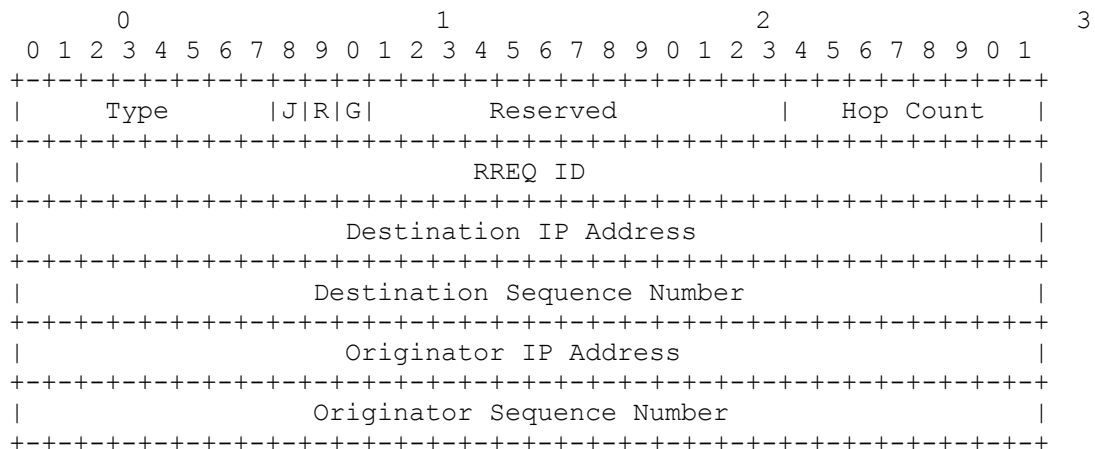
It would be appealing if the simulator was coupled with a Graphical User Interface (GUI). Furthermore, while simulating real world complex scenarios the simulator must also consider the approach to simulate radio obstacles in the wireless communication medium. Also, the simulator should also be able to generate trace files for other simulators such as NS-2 or QualNet.

## 2.7 AODV in detail

As in VANET, nodes (vehicles) have high mobility and moves with high speed. Proactive based routing is not suitable for it. Proactive based routing protocols may fail in VANET due to consumption of more bandwidth and large table information. AODV is a reactive routing protocol, which operates on hop-by-hop pattern. The three types of messages with their formats are as follows:

**2.7.1 Message Formats [24]:** The message formats of RREQ, RREP and RERR messages used while AODV operation is as follows:

### a) Route Request (RREQ) Message Format



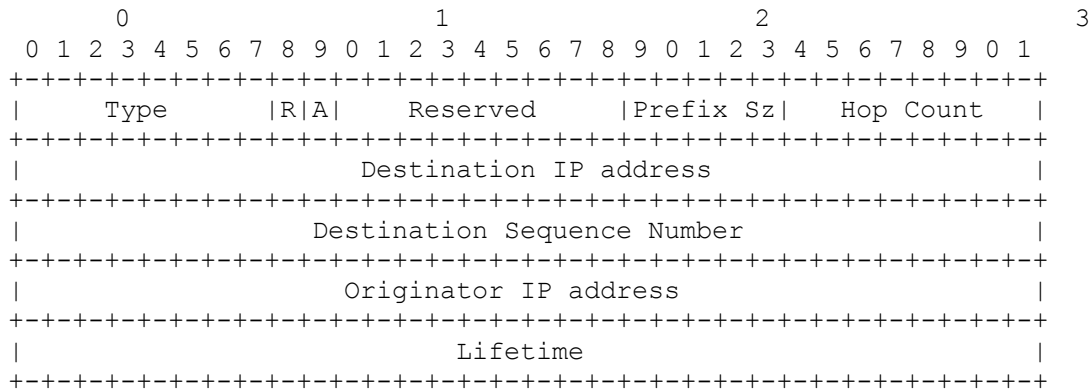
**Fig. 2.10: RREQ Message Format [24]**

The format of the Route Request message is illustrated above, and fields contained in it are shown in Table 2.2:

**Table 2.2: RREQ Field Description**

<i>Type</i>	1
<i>J</i>	Join flag; reserved for multicast.
<i>R</i>	Repair flag; reserved for multicast.
<i>G</i>	Gratuitous RREP flag; indicates whether a gratuitous RREP should be unicast to the node specified in the Destination IP Address field
<i>Reserved</i>	Sent as 0; ignored on reception.
<i>Hop Count</i>	The number of hops from the Originator IP Address to the node handling the request.
<i>RREQ ID</i>	A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originating node's IP address.
<i>Destination IP Address</i>	The IP address of the destination for which a route is desired.
<i>Destination Sequence Number</i>	The greatest sequence number received in the past by the originator for any route towards the destination.
<i>Originator IP Address</i>	The IP address of the node which originated the Route Request.
<i>Originator Sequence Number</i>	The current sequence number to be used for route entries pointing to (and generated by) the originator of the route request.

**b) Route Reply (RREP) Message Format**



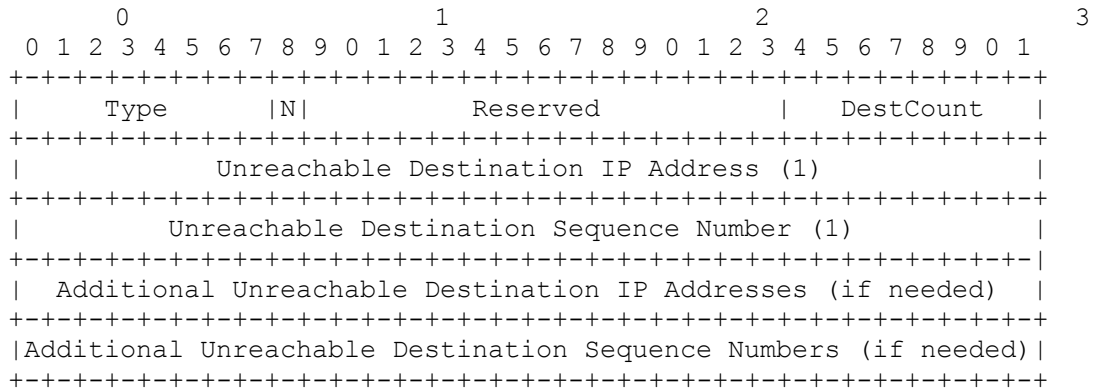
**Fig.2.11: RREP Message Format [24]**

The format of the RREP message (Fig. 2.11) contains the following fields (Table 2.3):

**Table 2.3: RREP Field Description**

<i>Type</i>	2
<i>R</i>	Repair flag; used for multicast.
<i>A</i>	Acknowledgment required; see sections 5 and 6.7.
<i>Reserved</i>	Sent as 0; ignored on reception.
<i>Prefix Size</i>	If nonzero, the 5-bit Prefix Size specifies that the indicated next hop may be used for any nodes with the same routing prefix as the requested destination.
<i>Hop Count</i>	The number of hops from the Originator IP Address to the Destination IP Address. For multicast route requests this indicates the number of hops to the multicast tree member sending the RREP.
<i>Destination IP Address</i>	The IP address of the destination for which a route is supplied.
<i>Destination Sequence Number</i>	The destination sequence number associated to the route.
<i>Originator IP Address</i>	The IP address of the node which originated the RREQ for which the route is supplied.
<i>Lifetime</i>	The time for which nodes receiving the RREP consider the route to be valid.

**c) Route Error (RERR) Message Format**



**Fig.2.12: RERR Message Format [24]**

The format of the RERR message (Fig.2.12) contains the following fields Table 2.4:

**Table 2.4: RERR Field Description**

<i>Type</i>	3
<i>N</i>	No delete flag; set when a node has performed a local repair of a link, and upstream nodes should not delete the route.
<i>Reserved</i>	Sent as 0; ignored on reception.
<i>DestCount</i>	The number of unreachable destinations included in the message; MUST be at least 1.
<i>Unreachable Destination IP Address</i>	The IP address of the destination that has become unreachable due to a link break.
<i>Unreachable Destination Sequence Number</i>	The sequence number in the route table entry for the destination listed in the previous Unreachable Destination IP Address field. The RERR message is sent whenever a link break causes one or more destinations to become unreachable from some of the node's neighbors.

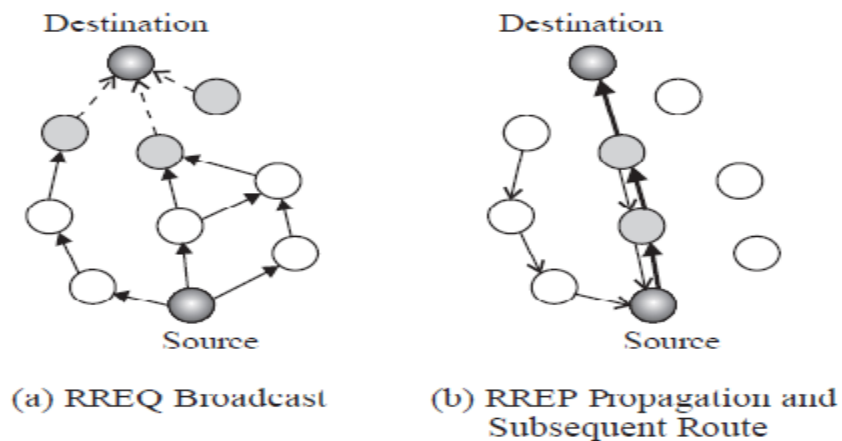
**2.7.2 Various mechanisms:** There are various mechanisms which are followed in AODV routing approach. These are as follows:

**a) AODV Route Discovery**

The need for basic route discovery arises when a source node wants to communicate with any particular destination node in order to forward data packet. AODV [24] uses route discovery by broadcasting RREQ to all its neighboring nodes. The broadcasted RREQ contains addresses of source and destination, their sequence numbers, broadcast ID and a counter, which counts how many times RREQ has been generated from a specific node. When a source node broadcast a RREQ to its neighbors it acquires RREP either from its neighbors or that neighbor(s) rebroadcasts RREQ to their neighbors by increment in the hop

counter. If node receives multiple route requests from same broadcast ID, it drops repeated route requests to make the communication loop free.

RREQ is generated from one source towards different destinations in order to reach at particular destination. If RREP is not received by the source node, it automatically setups reverse path to the source node. A reverse path is settled only when each node keeps the record of its neighbor from which it gets the RREQ (Fig.2.13). Reverse path is used to send a reply to source node, if any intermediate node does not satisfies the RREQ, moreover reverse path is settled for only the limited period of time.



**Fig. 2.13: Route discovery Cycle [25]**

All intermediate nodes stored the particular destination sequence number information and compare it with the RREQ destination sequence number. If RREQ sequence number is greater than or equal to stored sequence number of the intermediate node. Then the RREP is generated to source node following the same route from destination node to source node. This method is also known as the forward path discovery. And in this way a route is discovered for two nodes that need to communicate.

**b) AODV Route Table Management**

Routing table management in AODV is needed to avoid those entries of nodes that do not exist in the route from source to destination. Managing routing table information in AODV is handled with the destination sequence numbers. The need for routing table management is

important to make communication loop free. The following are characteristics to maintain the route table for each node .

- IP address of the particular destination.
- Total number of hops to the destination.
- Next hop: It contains information of those nodes that are used to forward data packets by using the current route.
- Destination sequence numbers.
- Active neighbors: Those nodes that currently using the active route.
- Expiration time: It contains information for the total time that route is being valid.

### **c) AODV Route Maintenance**

When nodes in the network detects that a route is not valid anymore for communication it delete all the related entries from the routing table for those invalid routes. And sends the RREP to current active neighboring nodes that route is not valid anymore for communication. AODV maintains only the loop free routes, when the source node receives the link failure notification it either start the process of rebroadcasting RREQ or the source node stop sending data through invalid route. Moreover, AODV uses the active neighbor's information to keep tracking of currently used route.

AODV reduces several problems that occurred in proactive routing protocols. AODV provide support by reacting at on demand needs for communication for such ad hoc network where large numbers of nodes. And this can help when the sudden change in topology happens.

So, AODV is an on demand algorithm, meaning that it maintains routes as long as they are needed by the sources. It uses sequence numbers to ensure the freshness of routes. These routes are loop-free, self-starting, and scale to large numbers of mobile nodes.

### **3.1 Problem Statement**

The field of VANETs started gaining attention after 1980s and has now been an active field of research and development. Various types of challenges in vehicular communications have been identified and addressed. A large number of routing protocols have been proposed for VANET. A routing protocol governs the way that two communication entities exchange information; it includes the procedure in establishing a route, decision in forwarding, and action in maintaining the route or recovering from routing failure.

VANET routing protocols can be classified as topology-based and geographic (position-based). Topology-based routing protocols can further be divided into proactive (table-driven) and reactive (on-demand) routing. Enough research has already been carried out which includes the comparison of various routing protocols and their performance evaluation based on different mobility models.

It will be interesting to evaluate the performance of one of the routing protocol by varying the number of mobile nodes. For this purpose Ad Hoc On Demand Distance Vector (AODV) routing protocol is simulated because it has been observed that AODV is a better approach as compared to both Destination-Sequenced Distance Vector (DSDV) [26] and Dynamic Source Routing (DSR) [27].

Ad Hoc On Demand Distance Vector (AODV) is a topology-based reactive routing protocol, which operates on hop-by-hop pattern. AODV maintains the established routing path in the given period and copes well with fast-changing network topologies and high relative vehicle speeds. The performance of the proposed protocol can be evaluated using simulation tools, mainly - Network Simulator (NS2) [28] and MOVE (MObility model generator for VEhicular networks) [29] over SUMO (Simulation of Urban MObility) [30].

### 3.2 Objective and Sub-Tasks

The primary objective of this thesis is the simulation and analysis of AODV routing protocol with realistic mobility model for VANETs. This objective can be divided into various sub-tasks,

- i) Firstly, simulation environment is to be setup. MOVE tool is used for rapid generation of realistic mobility model along with SUMO and NS2.
- ii) The performance comparison is made with different number of nodes. Three different sets of node density would be used to compare the performance of the said protocol. Tracegraph is used to generate graphs for evaluation.
- iii) Results are compared under various parameters like throughput, packet size, End to End delay, dropped packets etc.
- iv) Reporting and analysis of the results obtained.

## CHAPTER 4

# INSTALLATION, SIMULATION & DESIGN

---

Developing VANETs in the practical application is too costly, so simulation of VANETs assumes importance in all the research activities.

It is important to test and evaluate protocol implementations in a real world environment, simulations are commonly used as a first step in the protocol development for VANET research. Several communications network simulator already exist to provide a platform for testing and evaluating network protocols, such as NS-2, OPNET [31] and Qualnet [32]. These tools are not particularly tailored for specific applications in the transportation environment rather these are basically designed to provide generic simulation scenarios. Simulations play an important role in the area of transportation. There are number of simulation tools available such as PARAMICS [33], CORSIM [34] and VISIM [35], MOVE, SUMO, NS-2, VanetMobiSim [36] etc which have been developed to analyze transportation scenarios at the micro and macro-scale levels.

The most important parameter in simulating ad-hoc networks is the node mobility. It's important to use real world mobility model so that the results from the simulation correctly reflect the real-world performance of a VANET. E.g. a vehicle node is typically constrained to streets which are separated by buildings, trees or other objects. Such obstructions often increase the average distance between nodes as compared to an open-field environment [37, 38].

In this thesis, a tool MOVE (MObility model generator for VEhicular networks) has been used which provides the facility to generate real world mobility models for VANET simulations. MOVE is a java based tool with GUI and built on top of micro traffic simulator SUMO. It does the interpretation between NS2 and SUMO. MOVE tool is built on top of an open source. The output of MOVE is a mobility trace file that contains information of real-world vehicle movements which can be used by NS-2 or Qualnet. In addition, MOVE provides a set of GUI that allows the user to quickly generate real-world simulation scenarios without any need to manually write simulation scripts as well as to learn about the internal details of the simulator.

A brief introduction of the tools used and their installation steps is given in next three sections 4.1, 4.2, 4.3,4.4. Configuration of the system on which these tools were deployed is given in table 4.1:

**Table 4.1: System Configuration**

OS	Ubuntu 10.04 (Lucid Lynx)
CPU	Intel(R) Core 2 Duo 1.80 GHz
RAM	3 GB

## 4.1. NS-2

A computer network is a complex system that requires a careful treatment in design and implementation. Simulation, regarded as one of the most powerful performance analysis tools, is usually used in carrying out such a treatment to complement the analytical tools.

In this thesis focus is mainly on time-dependent simulation, which advances in a time domain. The time-dependent simulation can be divided into two categories. Time-driven simulation advances the simulation by fixed time intervals, while event-driven simulation proceeds from one event to another. NS2 is an event-driven simulation tool.

### 4.1.1 Overview

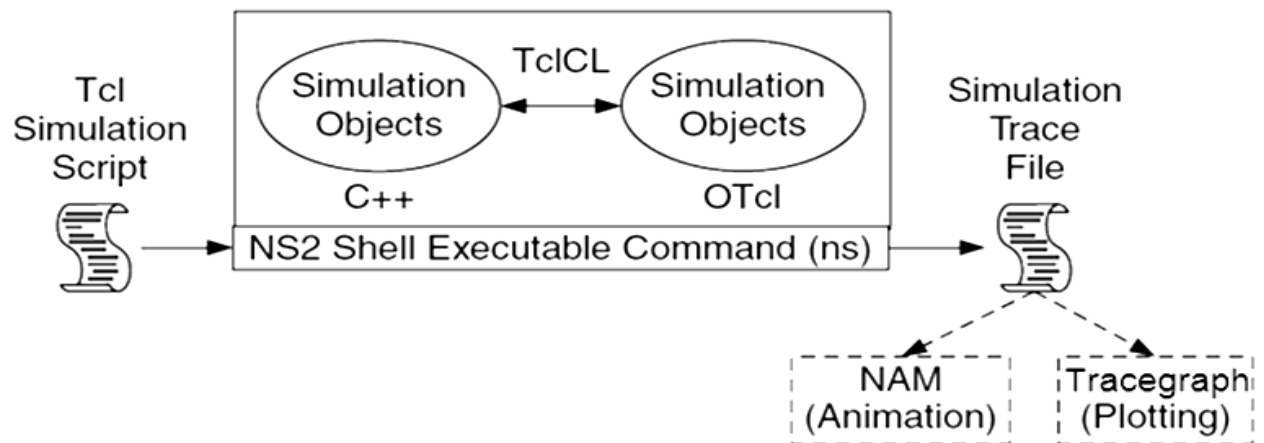
NS-2 is an open-source simulation tool running on Unix-like operating systems [28]. It is a discreet event simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP and SRM over wired, wireless and satellite networks. It has many advantages that make it a useful tool, such as support for multiple protocols and the capability of graphically detailing network traffic. Additionally, NS-2 supports several algorithms in routing and queuing. LAN routing and broadcasts are part of routing algorithms. Queuing algorithm includes fair queuing, deficit round robin and FIFO.

It is packaged with a bundle of rich libraries for simulating wireless networks. All the mobile nodes in NS-2 quickly assume that they are the part of Ad-hoc network and the simulation mobile nodes connected with infrastructure networks are not really possible. For simulating a

wireless node the physical layer, the link layer and MAC (media access control) protocol are all included at the same time.

#### 4.1.2 Basic Architecture

Fig. 4.1 shows the basic architecture of NS2. NS2 provides users with an executable command ns which takes on input argument, the name of a Tcl simulation scripting file. Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns. In most cases, a simulation trace file is created, and is used to plot graph and/or to create animation.



**Fig. 4.1: Basic architecture of NS2**

#### 4.1.3 Implementing languages

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL.

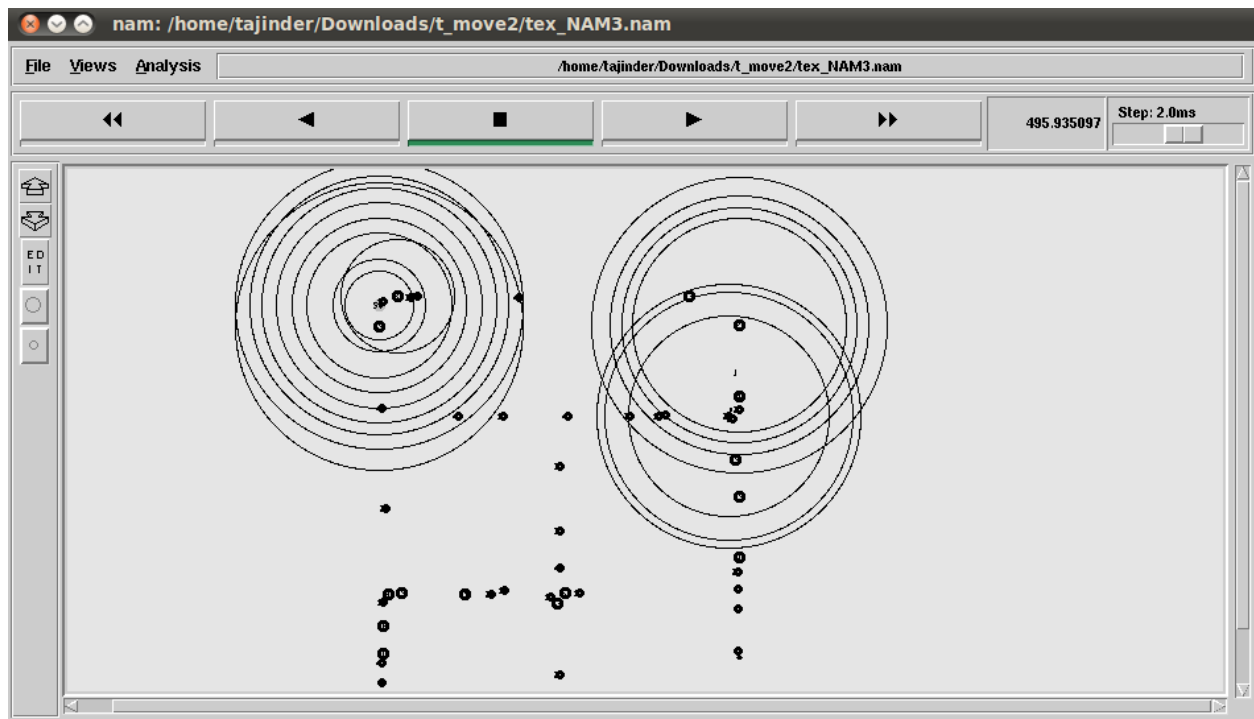
#### 4.1.4 Tool Command Language (Tcl)

Short for Tool Command Language, Tcl is a powerful interpreted programming language developed by John Ousterhout at the University of California, Berkeley. Tcl is a very powerful and dynamic programming language. It has a wide range of usage, including web

and desktop applications, networking, administration, testing etc. Tcl is a truly cross platform, easily deployed and highly extensible. The most significant advantage of Tcl language is that it is fully compatible with the C programming language and Tcl libraries can be interoperated directly into C programs.

#### 4.1.5 The Network Animator (NAM)

NAM [41][42] provides a visual interpretation of the network topology created. The application was developed as part of the VINT project. Fig. 4.2 displays the NAM visualization when .nam file has been loaded. This snapshot is taken for network of 10 nodes at an event time 495.935097.



**Fig.4.2: NAM Visualization**

#### 4.1.6 The Trace File

The trace file is an ASCII code files and the trace is organized in 12 fields as in Fig. 4.3.

Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id
-------	------	-----------	---------	----------	----------	-------	-----	----------	----------	---------	--------

**Fig. 4.3: Fields of trace file**

The trace file data explanation is given below starting from the first field as number 1.

1. Operation performed in the simulation, given by one of four available symbols r, +, - and d which correspond respectively to receive, enqueued, dequeued and dropped
2. Simulation time of event occurrence
3. Input node of link at which the events takes place.
4. Output node of link at which the events takes place.
5. Packet type like CBR or TCP.
6. Packet size
7. Flags
8. IP flow identifier
9. Packet source node address
10. Packet destination node address
11. Sequence number
12. Unique packet identifier

Depending on the user's purpose for an OTcl simulation script, simulation results are stored as trace files, which can be loaded for analysis by an external application:

1. A NAM trace file (file.nam) for use with the Network Animator Tool
2. A Trace file (file.tr) for use with XGraph or TraceGraph.

In this thesis NAM and Trace Graph were used to analyze the trace file.

#### **4.1.7 Installation steps**

The package could be easily installed in Ubuntu from PPA (Personal Package Archive). The ppa repository for ns2 created by Wouter Horré [39]. Following steps shows how it was installed on Ubuntu 10.04:

**Step 1:** Visited the PPA's overview page in Launchpad. There was a heading that read “*Adding this PPA to your system*” and clicked the Technical details about that PPA link.

**Step 2:** Used the Display sources.list entries drop-down box to select the version of Ubuntu.

**Step 3:** Two lines are there like these:

```
deb http://ppa.launchpad.net/wouterh/ppa/ubuntu lucid main
deb-src http://ppa.launchpad.net/wouterh/ppa/ubuntu lucid main
```

**Step 4:** These lines were copied and added to `/etc/sources.lst` by using the following command in terminal

```
$ sudo gedit /etc/apt/sources.list
```

This then opened a text editor with `sources.lst` contents. The copied lines were then added to the end portion. Saved the file and exited the editor.

**Step 5:** Back on the PPA's overview page, there was a Signing key heading. We get 1024R/B3F3334F. To add the key, following command is executed in terminal.

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys B3F3334F
```

**Step 6:** After this the packages were installed by running the following commands in terminal

```
$ sudo apt-get update
```

```
$ sudo apt-get install ns nam xgraph
```

After these steps NS2.33 got installed.

## 4.2. SUMO

"Simulation of Urban MObility", or "SUMO" [30] for short is a highly portable, microscopic road traffic simulation package designed to handle large road networks. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center

It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network.

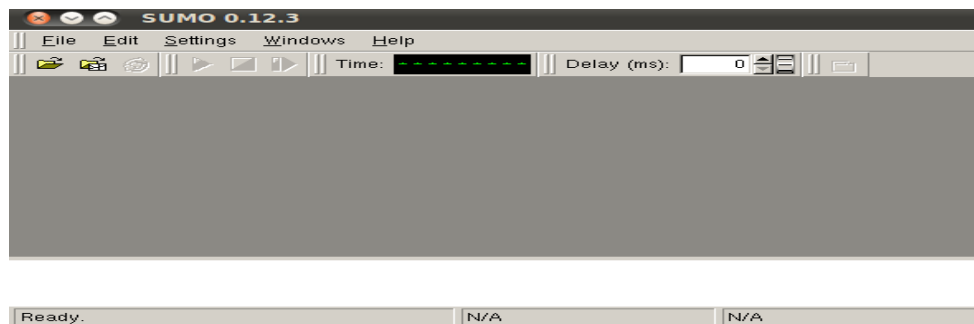
### 4.2.1. Features

- Complete workflow (network and routes import, DUA, simulation)
- Simulation
  - Collision free vehicle movement
  - Different vehicle types
  - Multi-lane streets with lane changing

- Junction-based right-of-way rules
- Hierarchy of junction types
- A fast openGL graphical user interface
- Manages networks with several 10.000 edges (streets)
- Fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine)
- Interoperability with other application on run time using TraCI
- Network-wide, edge-based, vehicle-based, and detector-based outputs
- Network
  - Many network formats (VISUM, Vissim, Shapefiles, OSM, Tiger, RoboCup, XML-Descriptions) may be imported
  - Missing values are determined via heuristics
- Routing
  - Microscopic routes - each vehicle has an own one
  - Dynamic User Assignment
- High portability
  - Only standard c++ and portable libraries are used
  - Packages for Windows main Linux distributions exist
- High interoperability through usage of XML-data only

#### 4.2.2 SUMO GUI

SUMO-GUI [43] is basically the same application as SUMO, just extended by a graphical user interface. Initial appearance of SUMO-GUI is shown below.



**Fig.4.4: SUMO Visualization**

**SUMO-GUI** uses the same configuration files as SUMO.

### 4.2.3 Installation steps on Ubuntu

Sumo 0.12.3 has been installed on our system. Its installation steps are as follows:

1. Downloaded sumo-src-0.12.3.tar.gz

2. Extracted it to home folder, here we extracted to /home/tajinder/Downloads

```
root@ubuntu:/home/tajinder/Downloads# tar xvzf sumo-src-0.12.3.tar.gz
```

3. Installed important libraries

```
root@ubuntu:/home/tajinder/Downloads# apt-get install gfortran libproj-dev  
libxerces-c-dev libfox-1.6-dev
```

Then installed sumo by following commands.

4. **root@ubuntu:/home/tajinder/Downloads# cd sumo-0.12.3**

5. **root@ubuntu:/home/tajinder/Downloads/sumo-0.12.3#./configure**

6. **root@ubuntu:/home/tajinder/Downloads/sumo-0.12.3# make**

7. **root@ubuntu:/home/tajinder/Downloads/sumo-0.12.3# make install**

With these steps sumo-0.12.3 got installed for use.

### 4.3 MOVE Simulator

MOVE [44] is implemented in Java and runs on top of an open-source micro-traffic simulator SUMO. We used following steps to implement MOVE.

First install java sdk 1.6 and NS-2 Version 2.33 on Ubuntu 10.04 "Lucid Lynx"

a) openjdk-6-jdk [45]

➤ SUMO version: 0.12.3

➤ Xerces (XML-parser)

➤ FOX-Toolkit (GUI Toolkit)

➤ PROJ (Cartographic Projections Library)

GDAL (Geospatial Data Abstraction Library)

b) NS2 version: 2.33

Here OpenJDK Development Kit (JDK) is a development environment for building applications, applets, and components using the Java programming language. The packages are built using the IcedTea build support and patches from the IcedTea project.

Following commands are needed to install openjdk-6-jdk on ubuntu:

```
root@ubuntu:/home/tajinder# apt-get update
```

```
root@ubuntu:/home/tajinder# apt-get install openjdk-6-jdk
```

Executed MOVE jar file with following command:

```
root@ubuntu:/home/tajinder# java -jar MOVE.jar
```

After this command MOVE main window appears as shown in Fig. 4.5.



**Fig. 4.5: MOVE main menu**

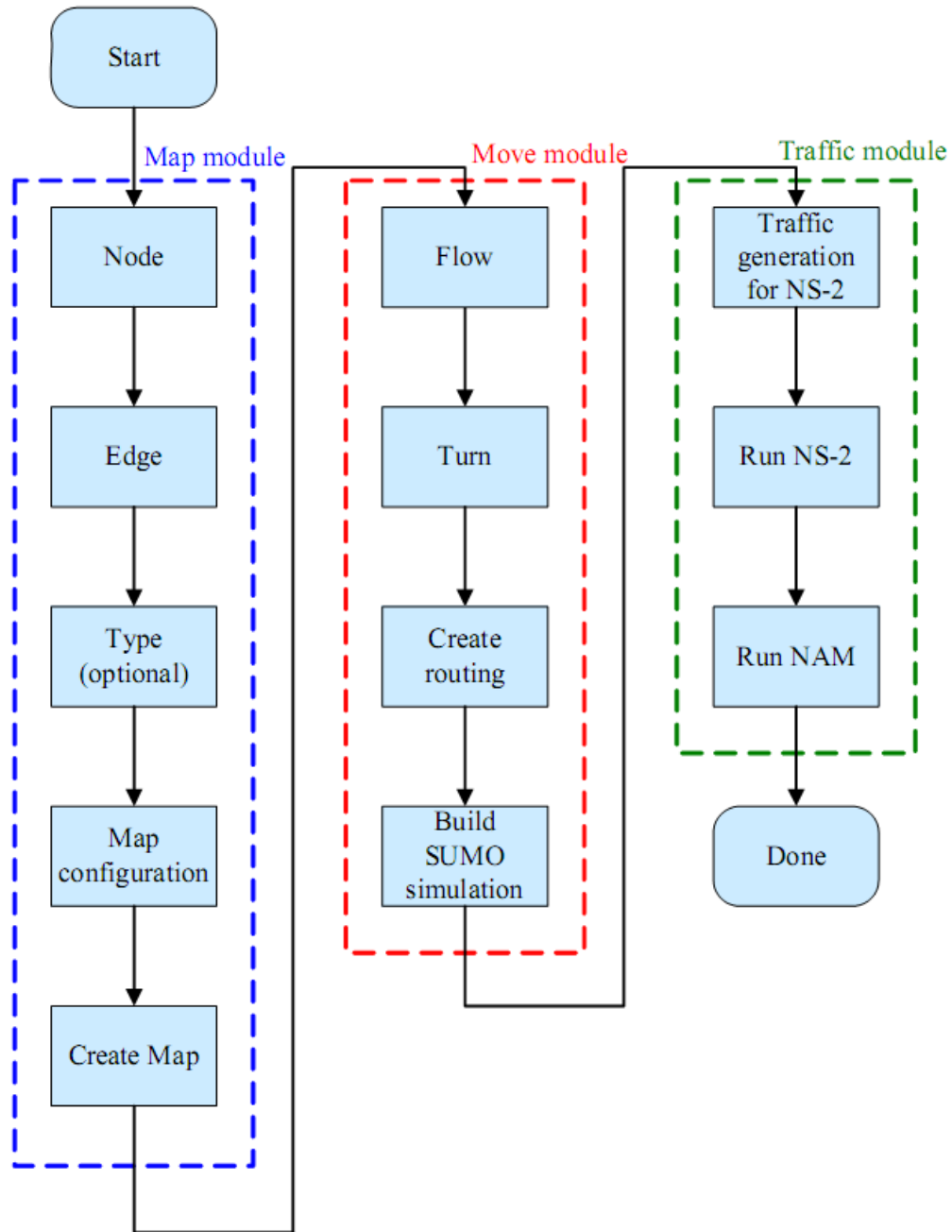
MOVE consists of two main components: Mobility model and traffic model generator.

1. **Mobility Model Generator:** It provides a user friendly interface for generating mobility model for simulations using SUMO. It allows the user to create customized topology or import maps.
2. **Network Traffic Model generator:** It takes the SUMO trace file as the input and generates the network traffic model as required by either NS2 or Qualnet. It provides all the configurable option of NS2 TCL files, like specifying MAC, routing protocol to use, etc.

The use of these two model generators are described in the following sections.

#### **4.3.1 Mobiliy Model Generation**

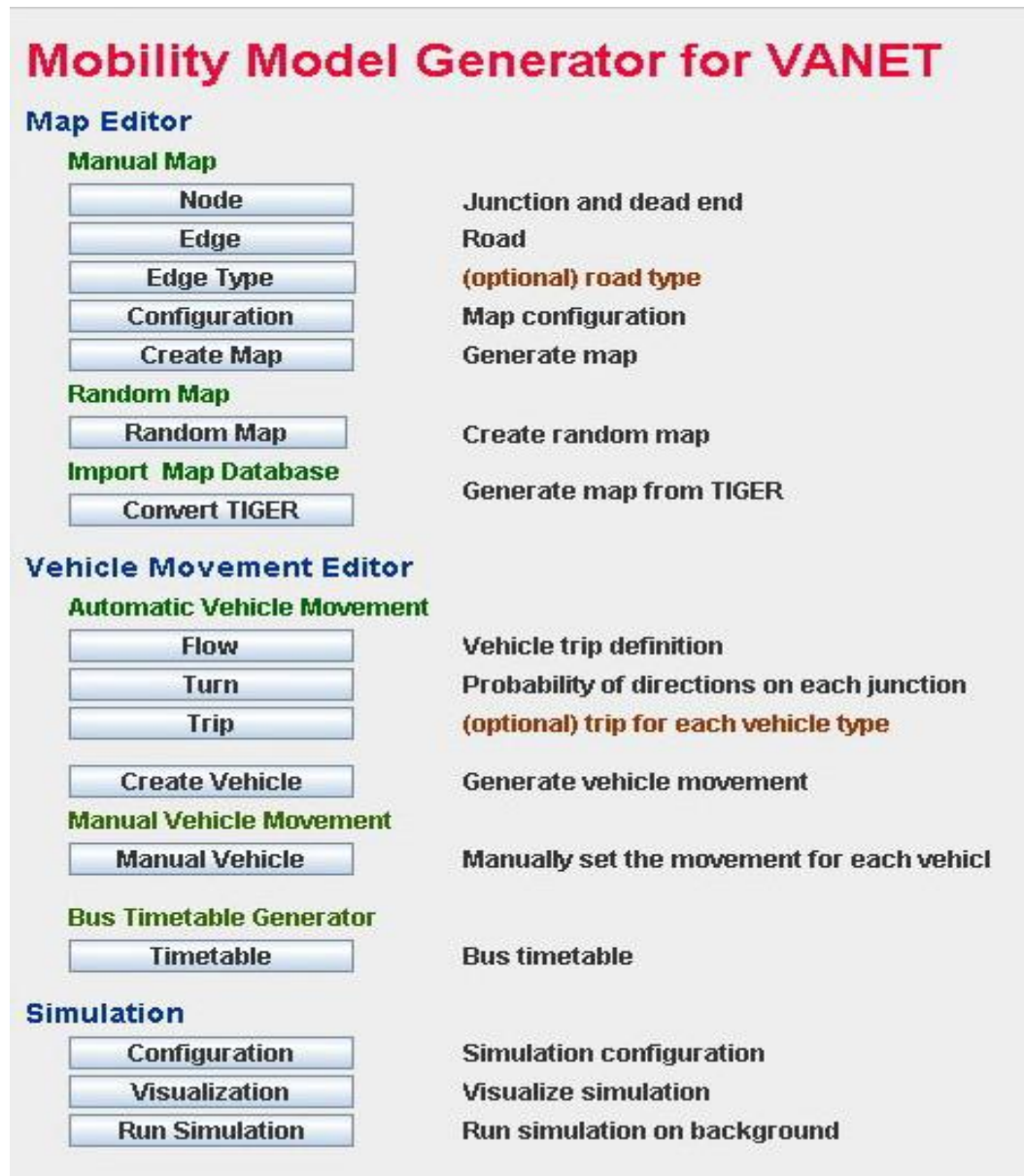
Firstly select "Mobility Model "on the main top level menu (Fig. 4.5). This part of the software (called MOVE - MObility model generator for VEhicular networks) will generate the mobility model created by SUMO. It has three main modules:



**Fig. 4.6: This is the roadmap in the VANET manual [44]**

Fig. 4.6 [44] gives the roadmap which helps user in creating and practicing VANET. Fig. 4.7 is the snapshot of Mobiliy Model Generator. There are two main editors in it:

- a) Map Editor
- b) Vehicle Movement Editor



**Fig. 4.7: Mobily Model Generation Main menu**

**a) Map generation**

The Map Editor is used to create the road topology. Road map can be created using three different ways.

- The map can be manually created.
- Generated automatically
- Imported from existing real world maps such as Google maps.

We have manually created map using following steps:

### Step 1: Manually create your own map nodes

First, select "Node" (Node Editor) from MOVE main menu (Fig. 4.7). This is simply where all the map nodes are. In Fig. 4.8 is the snapshot of Map Nodes Editor with its values used in this thesis. Here IDs can be set automatically and X-Y are the coordinates. After entering the values save the file as \* .nod.xml

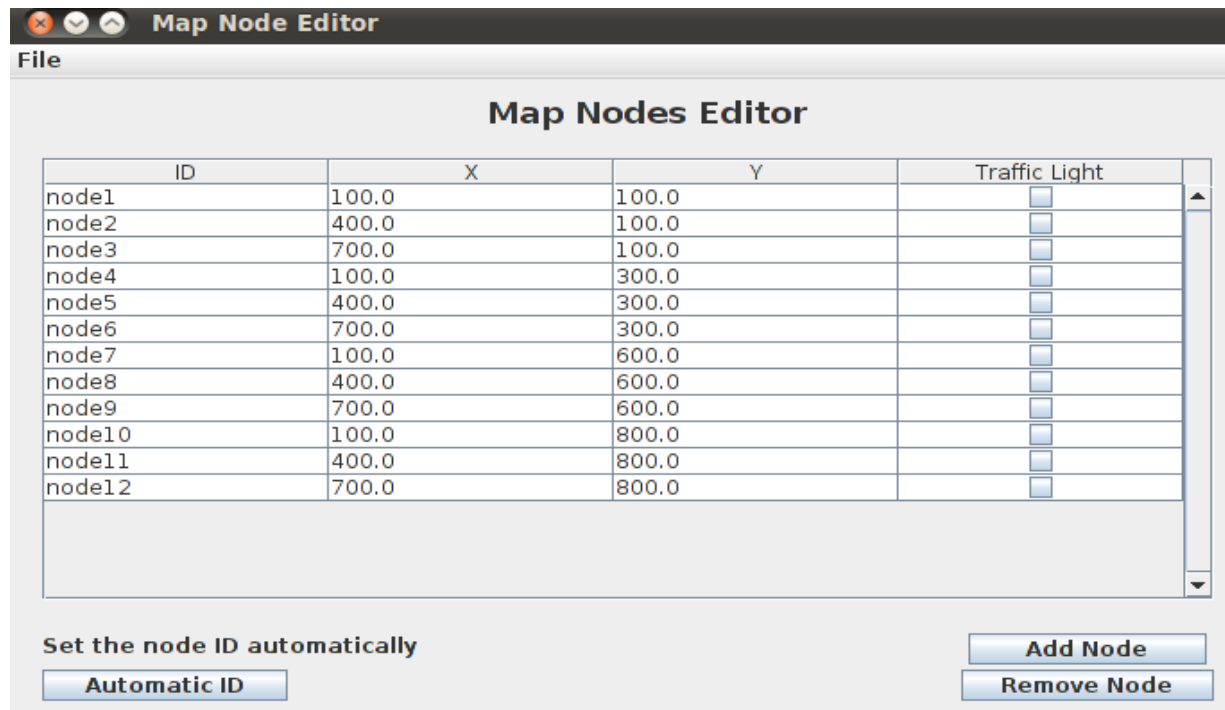


Fig. 4.8: Map Nodes Editor

### Step 2: Manually create your own map edge

Select "Edge" (Roads Editor) from MOVE main menu (Fig. 4.7). This is similar to node editor. This is where you can specify all the roads (a road will create a connection between two nodes created previously) and their properties (priority/lanes/speed/length). Snapshot of roads editor is shown in Fig. 4.8. In this No\_Lanes is the number of lanes in particular road. Speed is measured in m/s and determines the maximum speed a vehicle is able to run on this road. Priority will determine the "probability" a vehicle will turn to this road when facing a junction. Therefore higher priority will have more traffic.

Save the edge file with extension .edg.xml

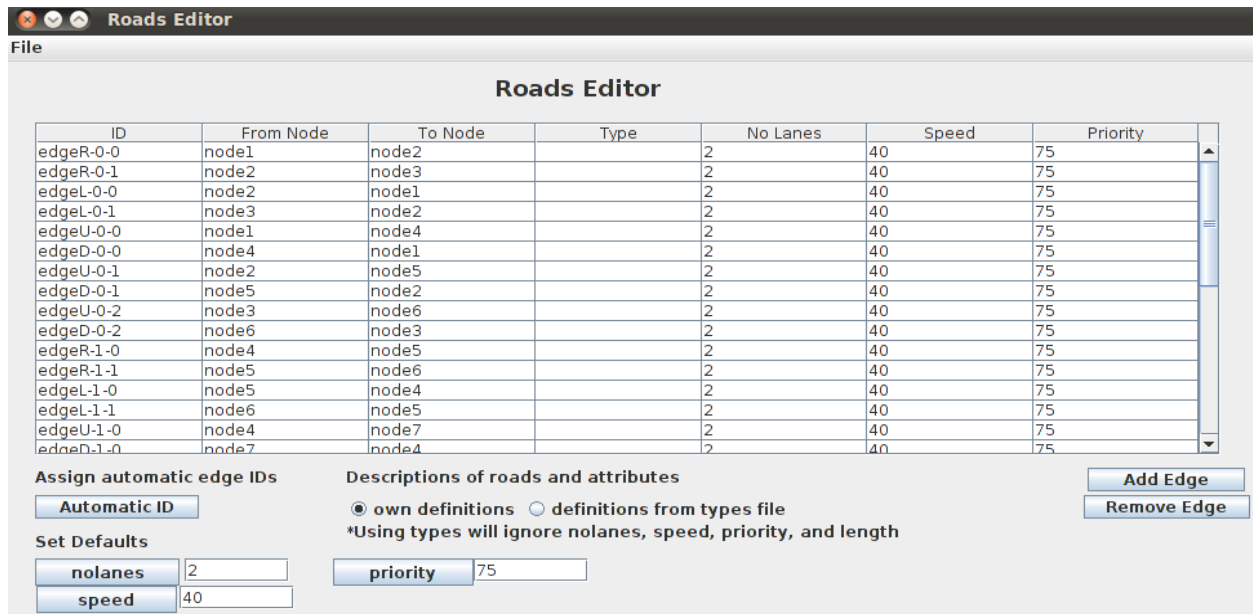


Fig. 4.9: Roads Editor

### Step 3: Map configuration editor

Choose “Configuration” in Map Editor from MOVE main menu (Fig. 4.7) and fill the details as shown in Fig. 4.10. Save the file with extension .netc.cfg

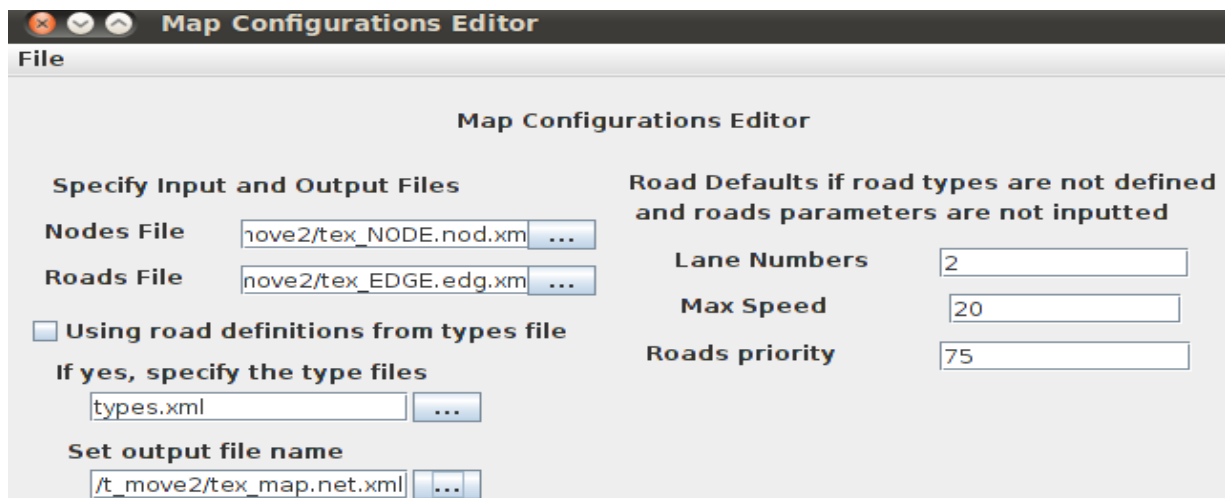


Fig. 4.10: Map Configuration Editor

#### Step 4: Generate the Map

Finally, select “Create Map” from MOVE main menu Fig. 4.7. Simply select previously created .netc.cfg file and click OK. A file with extension .net.xml (tex\_Map.net.xml) will be automatically

generated. This (tex\_Map.net.xml) is your map file.

#### b) Vehicle movement generation

After the map is created, it is time to generate the movements. There are three modules:

- Automatic vehicle movements
- Manual vehicle movements
- Bus Timetable Generator

In this thesis “Automatic vehicle movements” is used. Under this following steps are followed to generate vehicle movement.

#### Step 1: Using flow definition

Simply select “Flow” from MOVE main menu (Fig. 4.7). This editor will specify the groups of vehicle movements flow on simulation. Using these groups of flows, movements of the vehicles will be generated between the two specified edges as shown in Fig. 4.11. In this snapshot (Fig. 4.11) number of vehicles are specified as 10. After specifying flow definition, save the file as \*.flow.xml



Fig. 4.11: Vehicle Flow Definition Editor

### Step 2: Using junction turning ratio

Simply select “Turn” from MOVE main menu (Fig. 4.7). This is when you want to specify a turning ratio for each junction as shown in Fig. 8. In the editor, you must group percentages together so they sum up as 1.0 (100%).

Save this file with extension .turn.xml

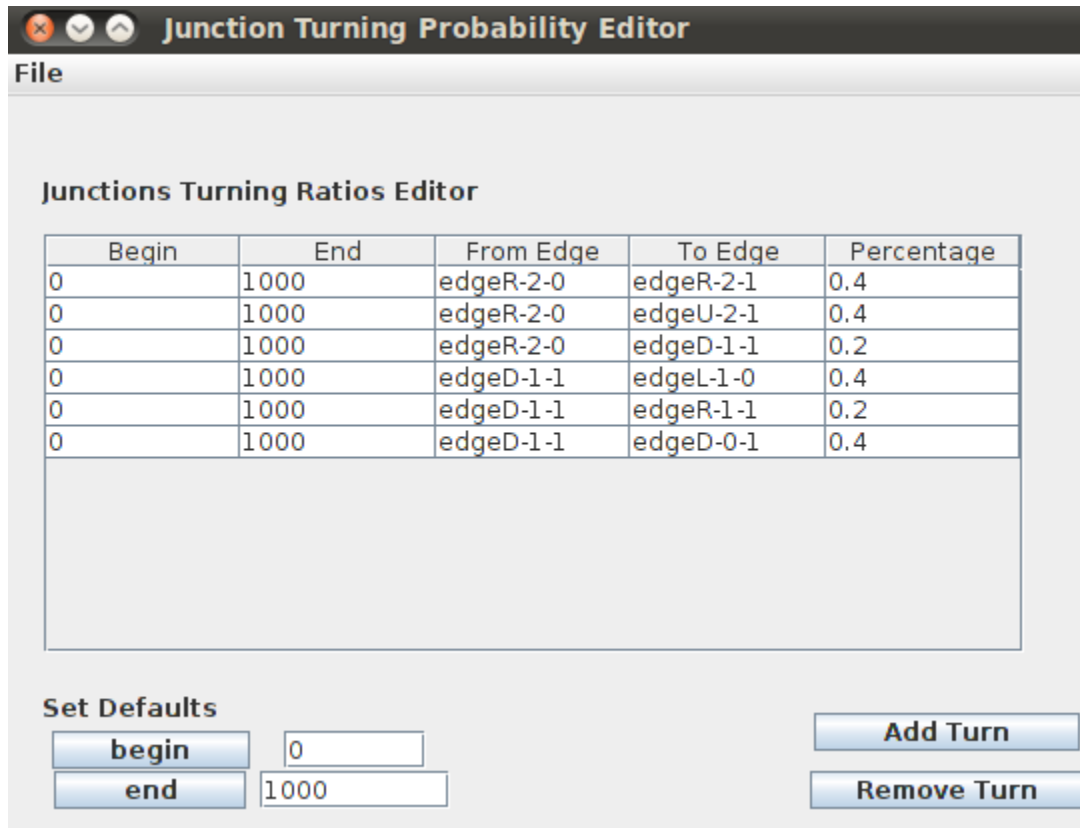
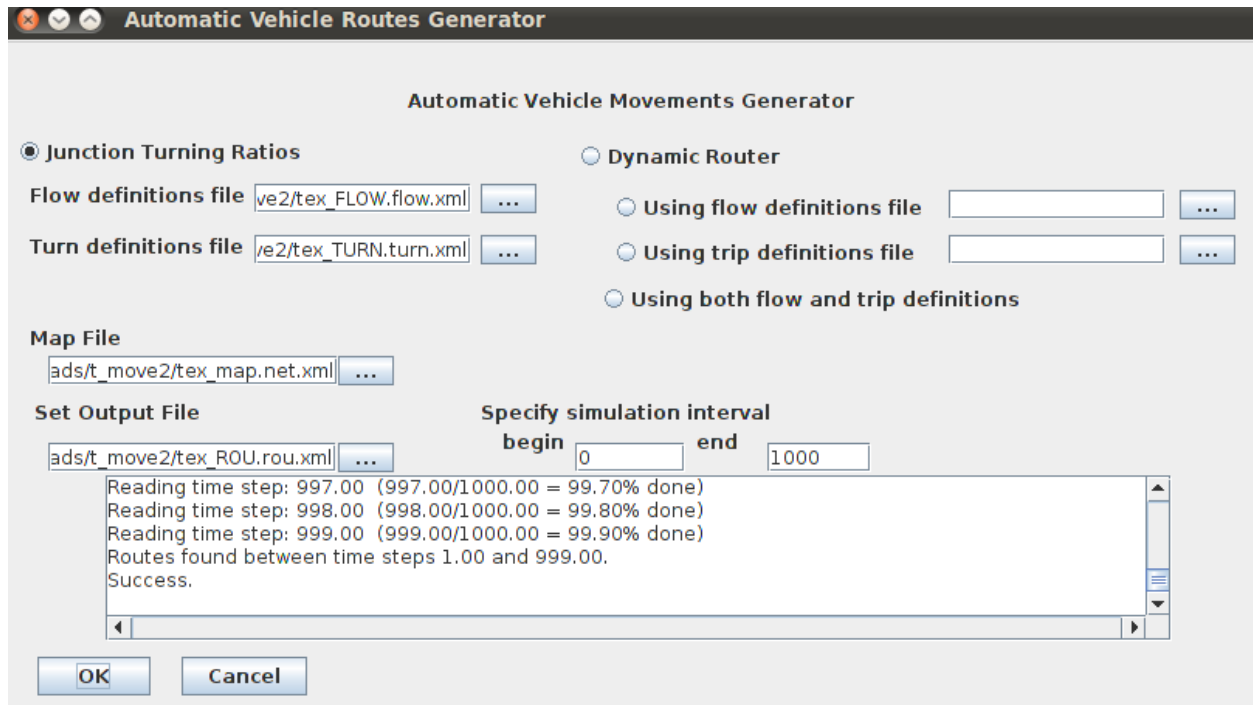


Fig. 4.12: Junction Turning Probability Editor

### Step 3: Automatic vehicle movements

Simply select “Create Vehicle” from MOVE main menu (Fig. 4.7). This will simply create a number of vehicles at the start of simulation. We have used junction turning ratios function. Select previously created map file (e.g. tex\_Map.net.xml). Specify output file location and name it as <name>.rou.xml (tex\_ROU.rou.xml). Set the beginning and end time of simulation. Finally click OK. The \*.rou.xml file will be automatically generated as shown in Fig. 4.13.



**Fig. 4.13: Automatic Vehicle Route Generator**

#### **Step 4: Simulation setup**

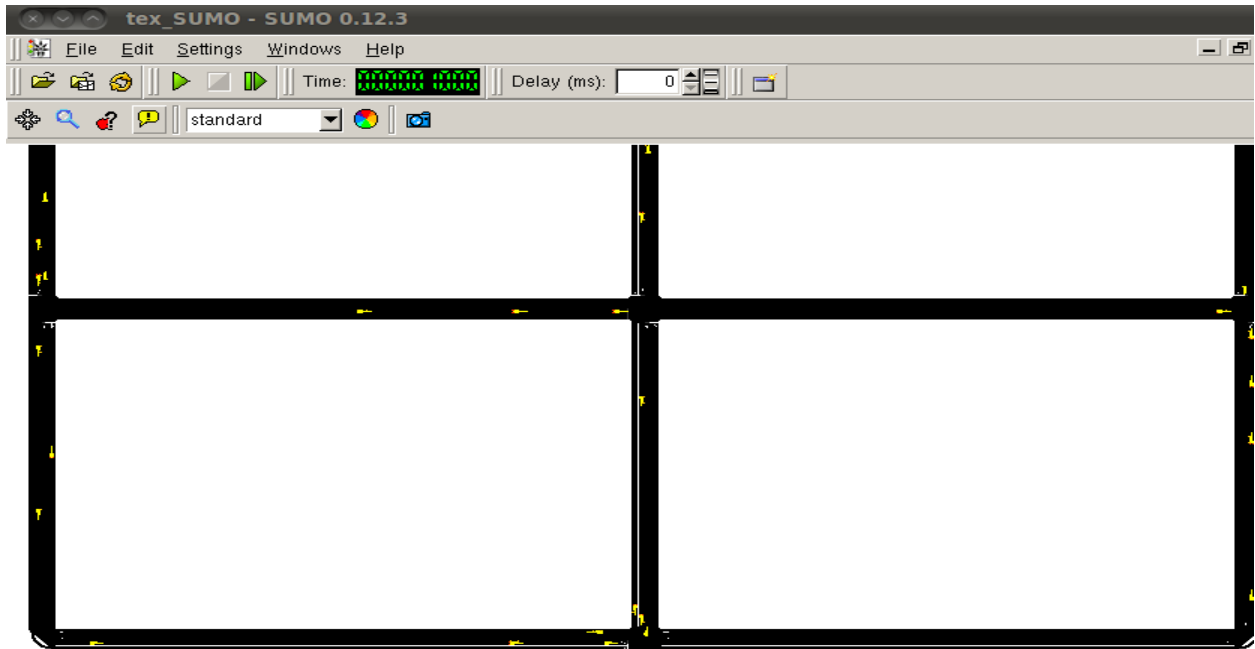
After the map and movement is complete, its need to specify the configurations of the simulation. Select “Configuration” at the bottom on MOVE main menu (Fig. 4.7).

Specify the <name>.net.xml (i.e. tex\_Map.net.xml) and <name>.rou.xml (i.e. tex\_ROU.rou.xml) location and specify the beginning and end time of simulation. Check the checkbox to create the trace file specify trace output name (e.g. <name>.sumo.tr). Then save the file with extension .sumo.cfg .

#### **Step 5: Visualize Simulation**

Then, one can select “Visualization” button and can visualize the generated mobility trace by seeing the movements of vehicles. The snapshot of this is shown in Fig. 4.14.

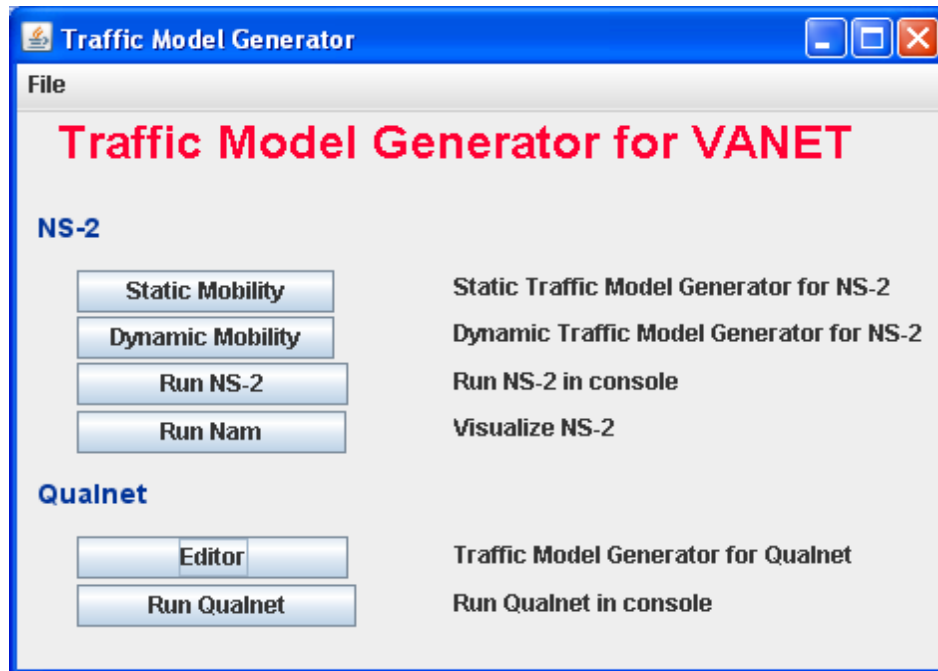
In this figure \*.sumo.cfg file has been loaded first and then by clicking the “play” button, it shows the movement of vehicles. This snapshot is taken at 1000 time step.



**Fig. 4.14: SUMO Visualization**

### 4.3.2 Traffic Model Generation

First, Select “Traffic Model” from MOVE main menu (Fig. 4.5) entering traffic model Fig. 4.15,



**Fig. 4.15: Traffic Model**

The traffic model generator consists of two main sections: for NS-2 and Qualnet. We have used NS-2 and Static Traffic Model generator in this thesis.

### Step 1: Traffic Model Generator for NS-2

Select “static mobility” option under NS2. This editor will generate the traffic simulation file (a tcl file) for NS-2 simulation tool. First import MOVE Trace \*.sumo.tr file and \*.net.xml file for script generator.

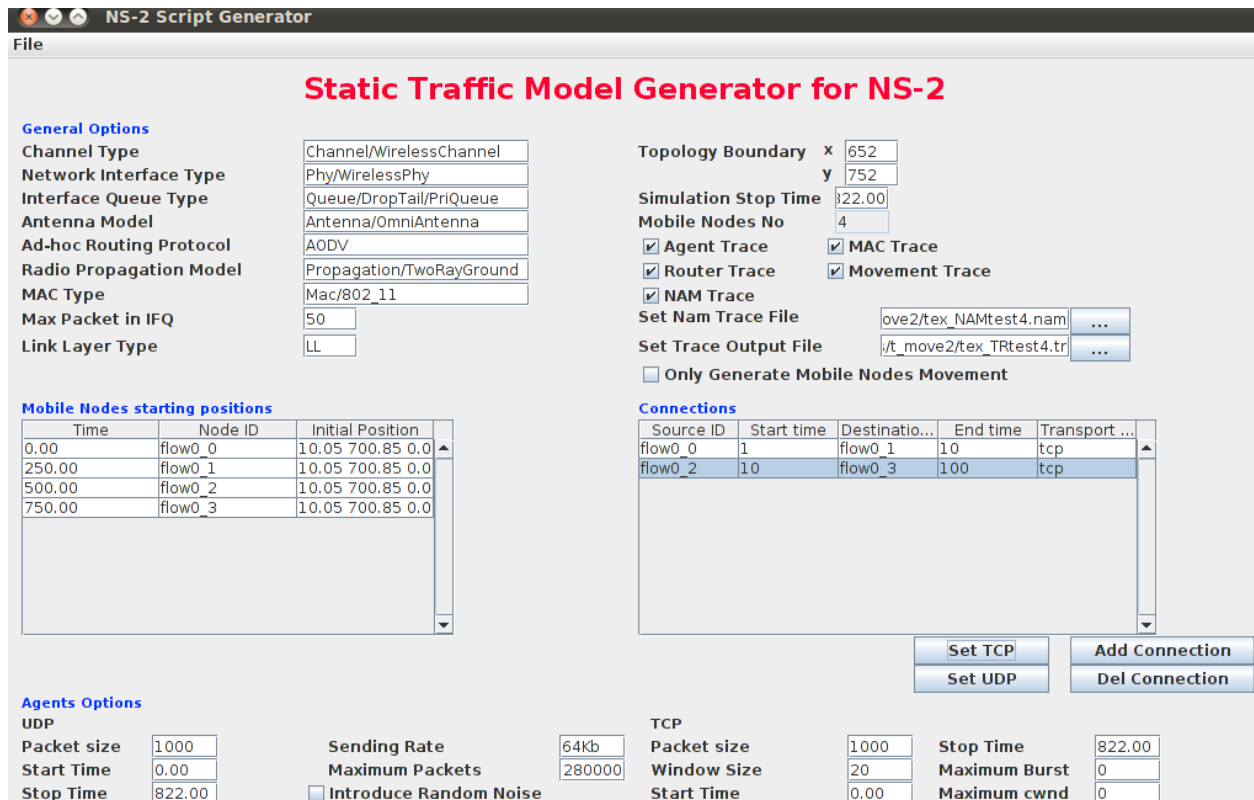


Fig. 4.16: Static Traffic Model Generator for NS-2

Choose NAM trace file and specify the file location with extension .nam. Then set the location for Trace Output file and name it with an extension .tr. Then we add the connections between each mobile node specified in the table on the left side and assign the transport protocol (e.g. tcp/udp).

When you are done, select File->Save or Save As and put it as <name>.tcl

**Simulation configuration:** The configurations set as per the assumed simulation context is shown in Table 4.2.

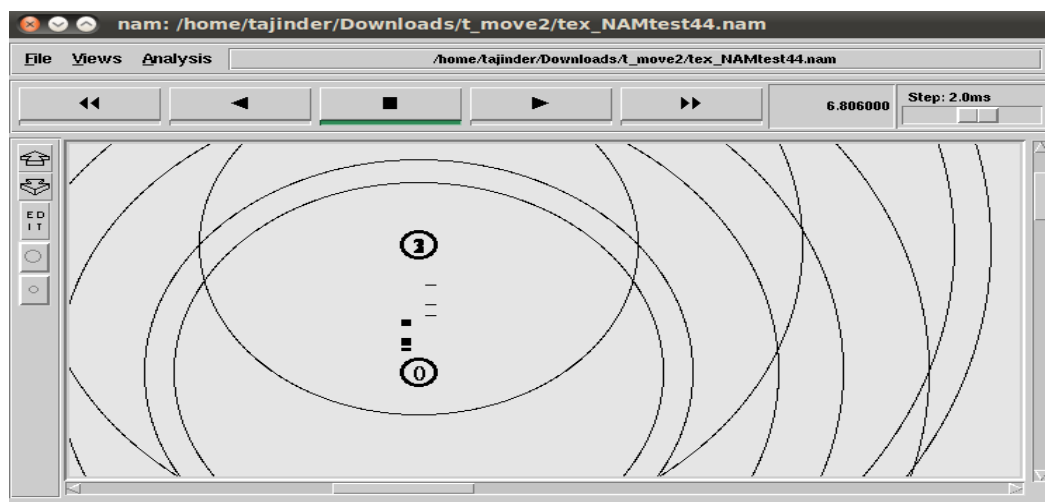
**Table 4.2: Simulation Configuration**

Channel type	Wireless
Network Interface type	Physical wireless
Routing protocol	AODV (NS2 default)
Interface queue type	Priority queue
Queue Length	50 packets
Number of nodes in topography	4, 10, 25
X and Y Dimensions of topography	652*752 sq.m
Time of Simulation end	100 simulation seconds
Traffic Type	TCP
Number of Road Lanes	2
Speed	40 m/s
Radio Propagation Model	Two Ray Ground
MAC protocol	IEEE 802.11

**Step 2: Run NS-2 in background console**

After, select “Run NS-2” runs NS-2 in console. Then run the simulation with <name>.tcl

**Step 3: Run NAM visualize NS-2 in example**



**Fig. 4.17 NAM Visualization**

Then, the NAM trace runner can be run from the main menu. One can play NAM and see the actual movements as in Fig. 4.17. This snapshot is taken at an event time of 6.806 for 4 nodes.

Next tracegraph has been used to plot graphs from the trace file created while making tcl file.

## 4.4 Tracegraph

Tracegraph [46] is a free network trace files analyser developed for network simulator ns-2 trace processing. Trace graph can support any trace format if converted to its own or ns-2 trace format.

Tracegraph supports various ns-2 trace file formats like wired, satellite, Wireless(old and new trace), new trace and wired-wireless.

There are various program features (version 2.05) available like 238 2D graphs; 12 3D graphs; delays, jitter, processing times, round trip times, throughput graphs and statistics etc.

### 4.4.1 Installation Steps

Installation instructions on Ubuntu 10.04:

1. Download tracegraph and mglinstaller from the webpage
2. Extract *tracegraph202linux.tar.gz* to the home folder (/home/tajinder). This would result in /home/tajinder/tracegraph202 with following command:  
**root@ubuntu:/home/tajinder# tar -xvzf tracegraph202.linux.tar.gz**
3. **root@ubuntu:/home/tajinder# gunzip mglinstaller.gz**  
This would uncompress the file "mglinstaller.gz" and would give the file "mglinstaller".
4. **root@ubuntu:/home/tajinder# ./mglinstaller**  
It would then be prompted with an information, all we have to do is to hit ENTER. This would create a new folder named glnx86 in the following location:  
/home/tajinder/tracegraph202/bin/
5. **root@ubuntu:/home/tajinder# export**  
**LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:/home/micman/tracegraph202/bin/g**  
**lnx86**

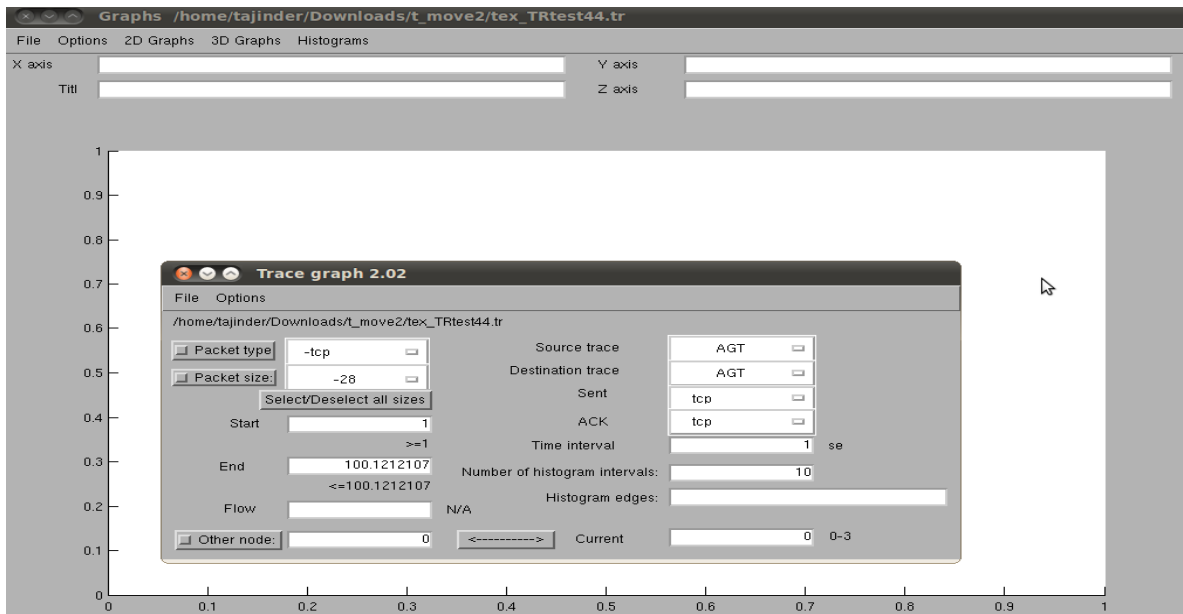
6. To run tracegraph every time, just navigate to

/home/tajinder/tracegraph202/bin/glnx86 in Terminal and execute trgraph as follows:

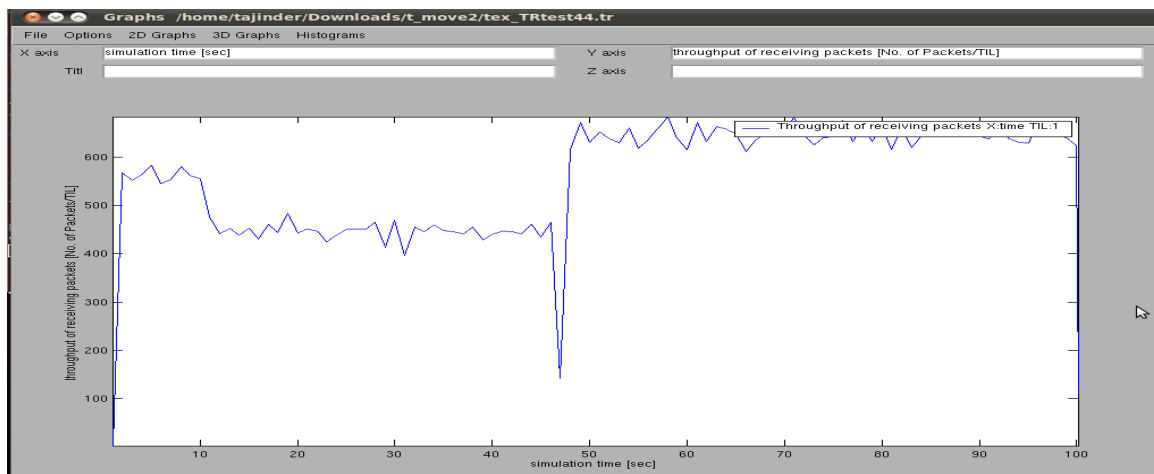
```
root@ubuntu:/home/tajinder# cd tracegraph202
```

7. root@ubuntu:/home/tajinder/tracegraph202#./trgraph

The figure below Fig. 4.18 is the snapshot of tracegraph after it has loaded trace file (\*.tr) for 4 nodes.



**Fig. 4.18: Tracegraph**



**Fig. 4.19: Graph plotted by Tracegraph**

Fig. 4.19 shows graph plotted by tracegraph for throughput of receiving packets when number of nodes are four.

## CHAPTER 5 RESULTS & EVALUATION

---

Experiment has been carried out for three different numbers of nodes under various cases and results are drawn and evaluated. The field configurations used is: 652 m x 752 m field with following number of nodes:

- I. 4 nodes
- II. 10 nodes
- III. 25 nodes

Various parameters used for performance evaluation are:

- i. **Throughput:** It is the amount of data per time unit that is delivered from one node to another via a communication link. The throughput is measured in Packets per unit TIL or bits per TIL. TIL is Time Interval Length. More is the throughput of sending and receiving packets better is the performance. Lesser is the throughput of dropping packets better is the performance.
- ii. **Average throughput:** It is the average of total throughput. It is also measured in Packets per unit TIL or bits per TIL.
- iii. **Packet Drop:** It shows total number of data packets that could not reach destination successfully. The reason for packet drop may arise due to congestion, faulty hardware and queue overflow etc. Lower packet drop rate shows higher protocol performance.
- iv. **Packet size:** Size of packets in bytes.
- v. **Average simulation End to End delay (End2End delay):** This metric gives the overall delay, from packet transmission by the application agent at the source node till packet reception by the application agent at the destination node. Lower delay shows higher protocol performance. The following equation is used to calculate the average end-to-end delay,

**Average End to End Delay** =  $(T_{\text{DataR}} - T_{\text{DataS}})$ , Where

$T_{\text{DataR}}$  = Time data packets received at destination node

$T_{\text{DataS}}$  = Time data packets sent from source node.

The end to end delay is important metrics because VANET needs a small latency to deliver quick messages. It shows the suitability of the protocol for the VANET.

vi. **Simulation time:** Total time taken for simulation. It is measured in seconds.

Results are compared for following cases:

CASE 1: Throughput of sending packets.

CASE 2: Throughput of receiving packets.

CASE 3: Throughput of dropping packets.

CASE 4: Packet Size vs Average throughput of sending packets.

CASE 5: Packet Size vs Average throughput of receiving packets.

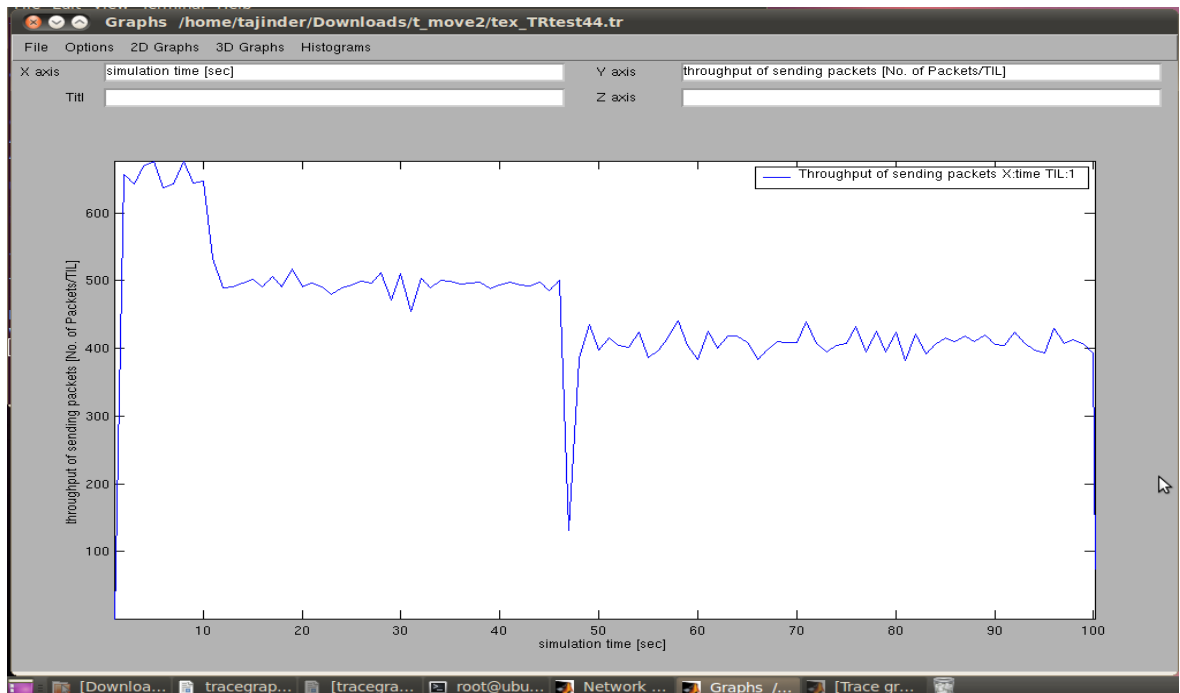
CASE 6: Packet Size vs Average throughput of dropping packets.

CASE 7: Throughput of sending bits vs Average simulation End2End delay.

CASE 8: Throughput of receiving bits vs Average simulation End2End delay.

### 5.1 CASE 1: Throughput of sending packets.

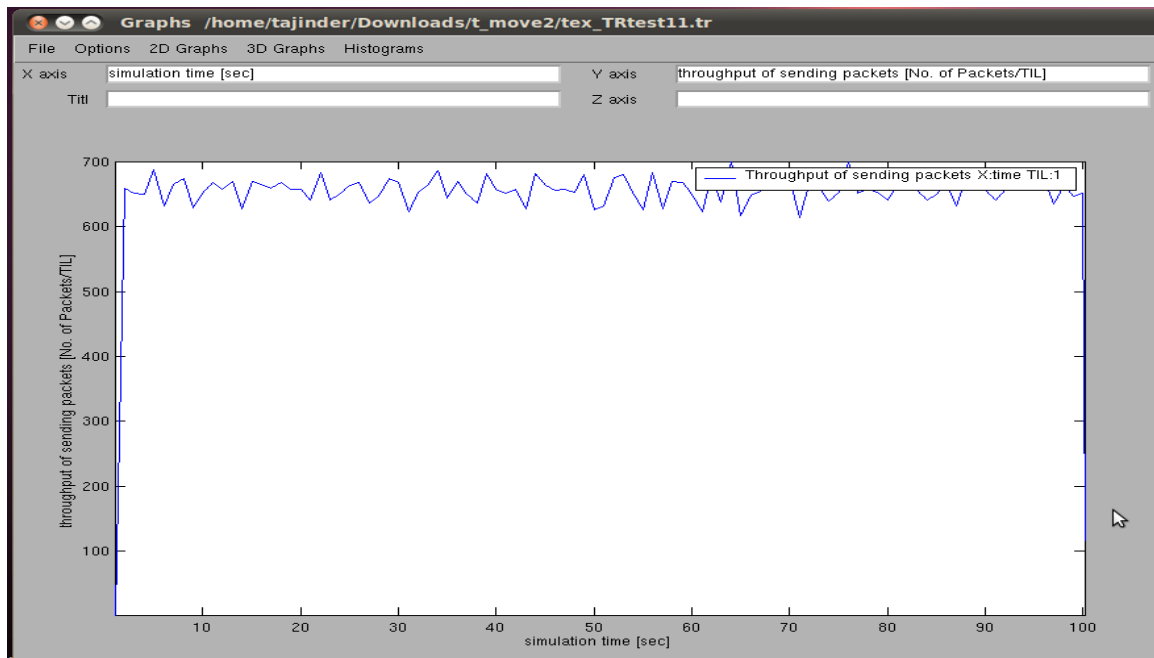
The graph is plotted for the throughput of sending packets against the simulation time. Throughput is the number of packets sent per unit TIL. TIL is the Time Interval Length. Simulation time is measured in seconds.



**Fig. 5.1: Throughput of sending packets for 4 nodes**

INFERENCE FROM FIG. 5.1: In this graph throughput abruptly increases to 640 packets/TIL in just 2 secs and then it continue to give thoughput of 640 packets/TIL approx. for about 10 secs , then it suddenly goes down to 500 packets/TIL and remain there with little variation for about 35secs. Then at 47 secs, it again drops to 102 packets/TIL and rises within 1 secs to 400 approx. Then it continues to give average throughput of 400 packets/TIL for the rest of the simulation time.

This can be understood as *“the number of packets sent per unit time decreases. The packets sent are maximum in the beginning because in the initial stage of VANET, the nodes are sending beacons in order to setup the network”*.



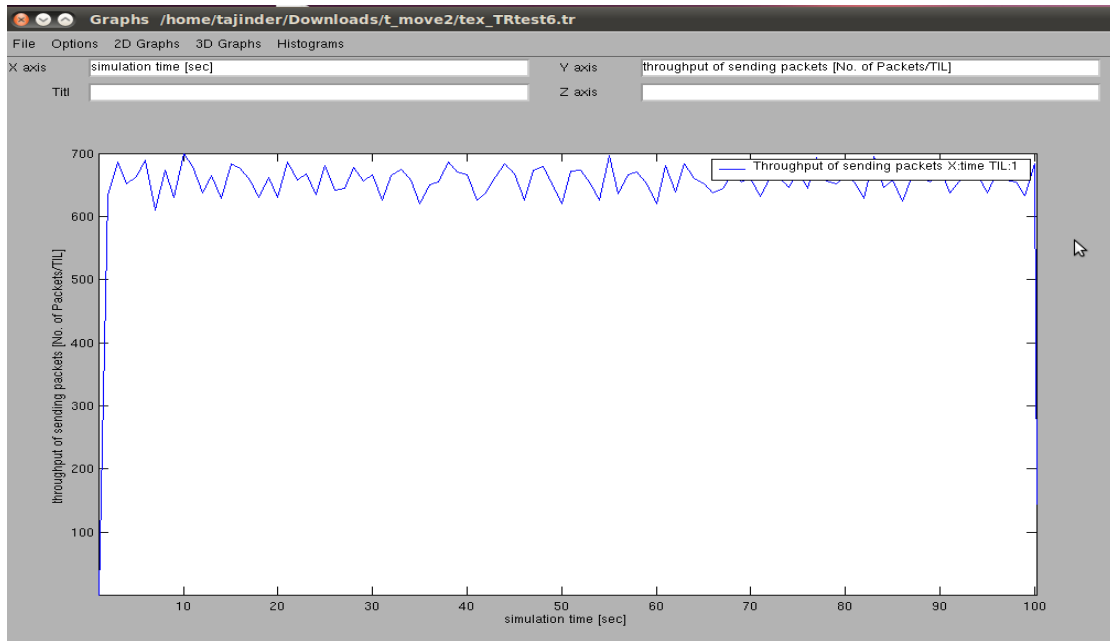
**Fig. 5.2: Throughput of sending packets for 10 nodes**

INFERENCE FROM FIG. 5.2: This graph is showing that throughput increases to 650 packets/TIL in just 1 sec in the beginning and then it keeps on giving an average throughput of 650 packets/TIL with little variation for rest of the simultaion time. Here total simulation time is 100 secs.

INFERENCE FROM FIG. 5.3: This graph is also showing that throughput increases to 650 packets/TIL in just 1 sec in the beginning and then it remains in range 600-700 packets/TIL

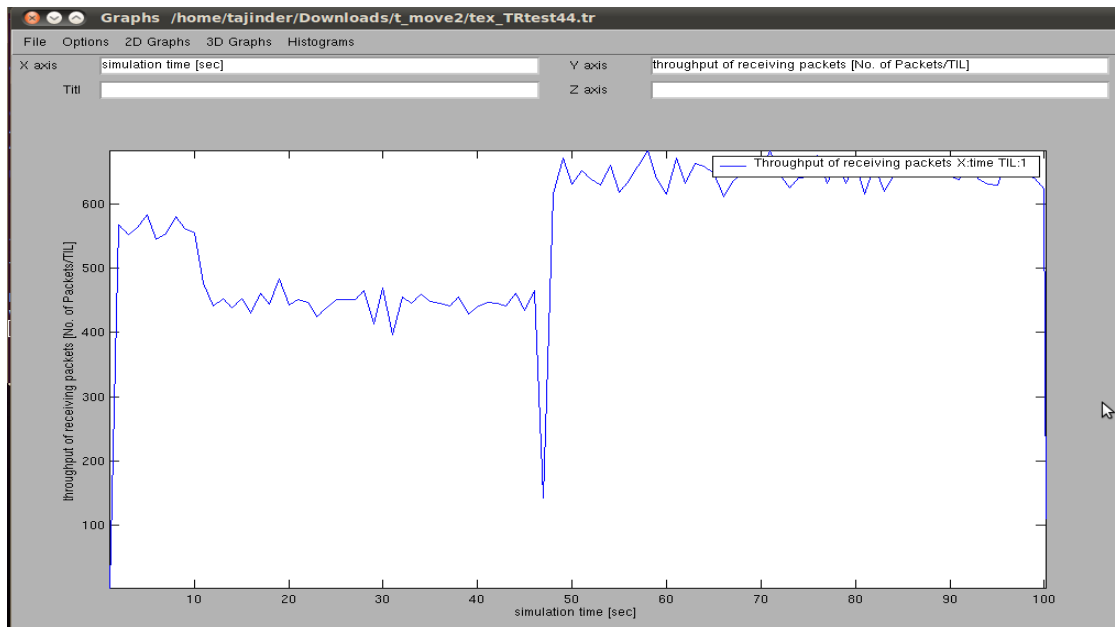
for rest of the simultaion time.

So, graphs for 10 and 25 nodes are more uniform than for 4 nodes.



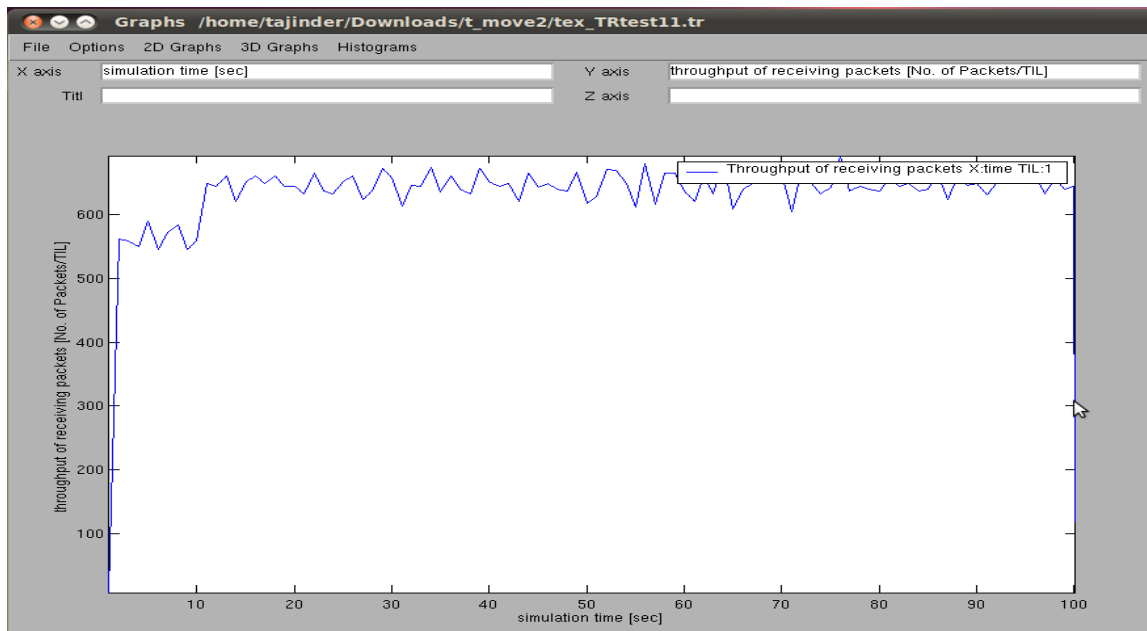
**Fig. 5.3: Throughput of sending packets for 25 nodes**

## 5.2 CASE 2: Throughput of receiving packets



**Fig. 5.4: Throughput of receiving packets for 4 nodes**

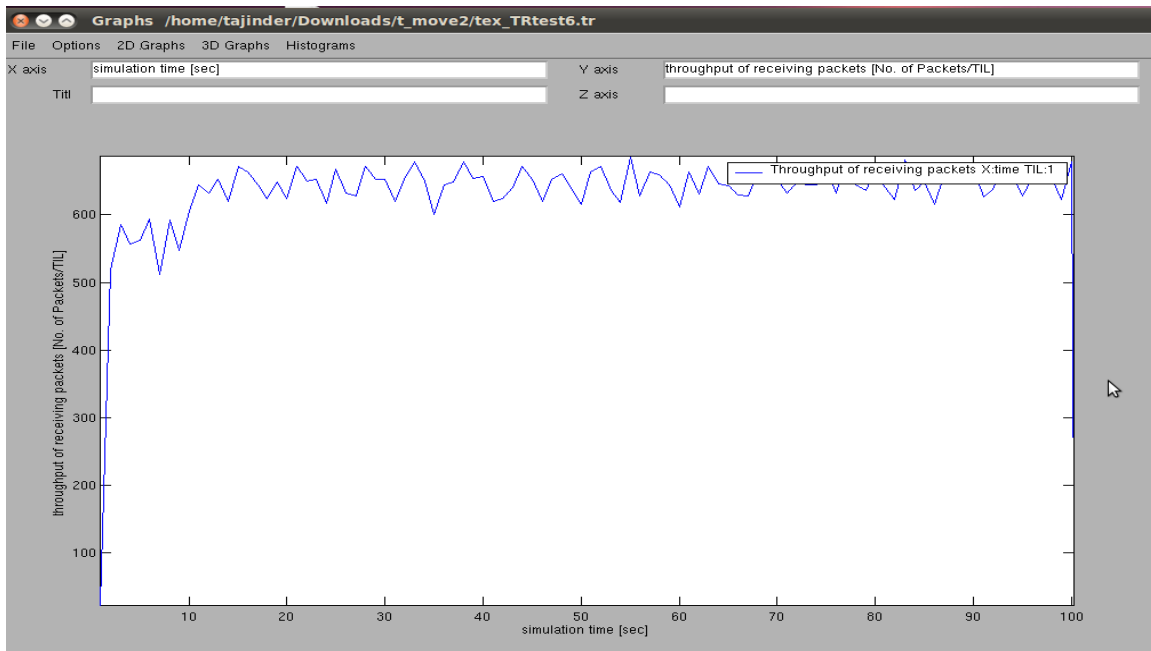
INFERENCE FROM FIG. 5.4: This graph shows that throughput peaks to 570 packets/TIL in just 2 secs initially, then it remains there for approx. 10 secs but then drops to 450 packets/TIL and continue to give same throughput for about 35 secs. Then it drops suddenly to 150 packets/TIL and rises to 630 packets/TIL in just 2 secs. Then for rest of the simulation time it gives an average throughput in the range of 610-690 packets/TIL approximately.



**Fig. 5.5: Throughput of receiving packets for 10 nodes**

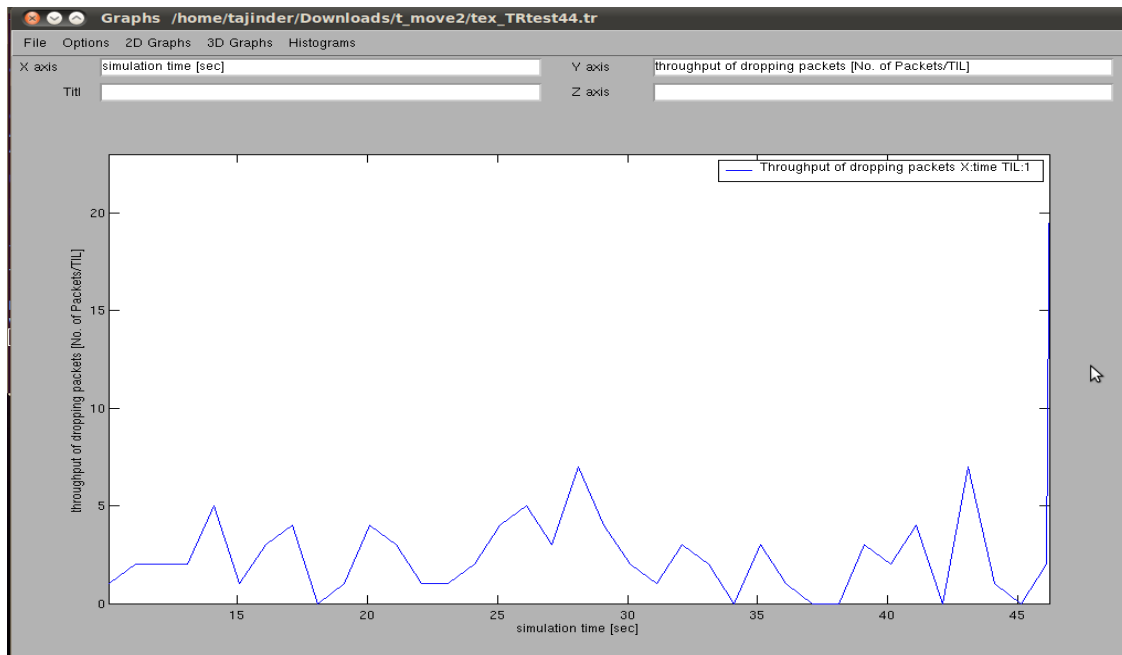
INFERENCE FROM FIG. 5.5: This graph is showing that throughput increases to 560 packets/TIL within 2 secs in the beginning and then it remains in the range 550-600 packets/TIL for 10 secs and then it rises suddenly to 650 packets/TIL then it keeps on giving throughput in the range of 610-690 packets/TIL approximately for rest of the simulation time.

INFERENCE FROM FIG. 5.6: This is a more uniform graph then for 4 nodes and 10 nodes. Here throughput rises to 580 packets/TIL in 3 secs then it remains in range of 500-600 packets/TIL for 10 secs approx. and then it rises above to 630 packets/TIL in 2 secs and then it remains in the range of 610-690 packets/TIL uniformly for rest of the simulation time.



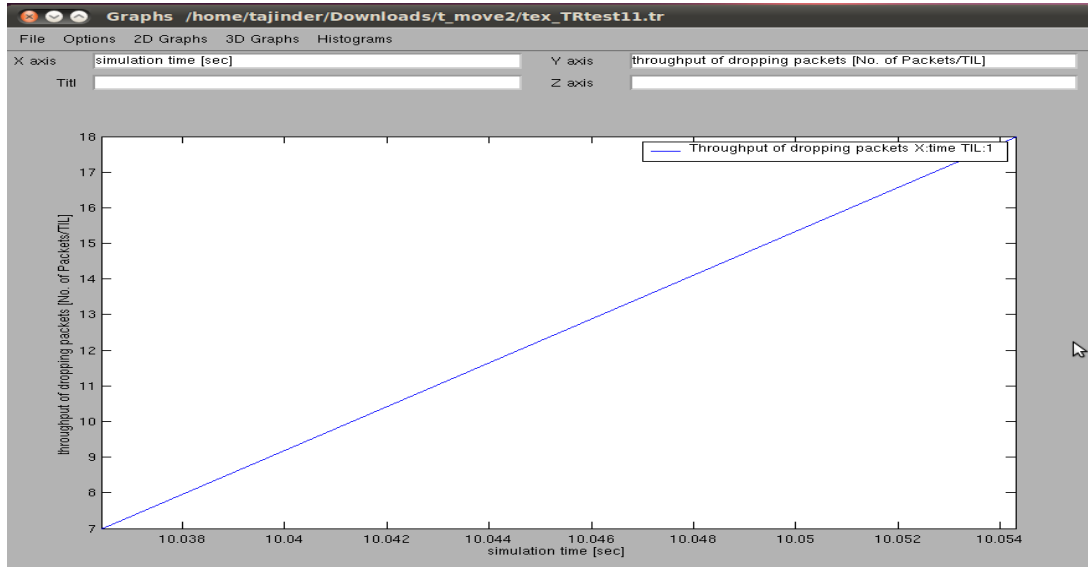
**Fig. 5.6: Throughput of receiving packets for 25 nodes**

### 5.3 CASE 3: Throughput of dropping packets



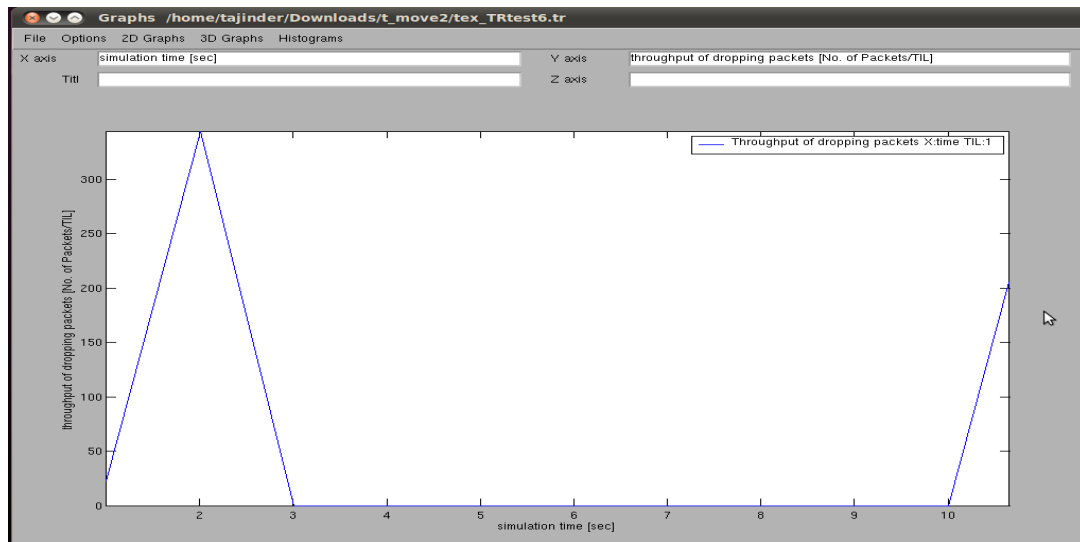
**Fig. 5.7: Throughput of dropping packets for 4 nodes**

INFERENCE FROM FIG. 5.7: This graph shows that throughput rises to only 3 packets/TIL in the beginning and rises once to 5 packets/TIL and drops to 2 packets/TIL in just 15 secs. Then in the simulation time of 46 secs, it rises twice to above 5 packets/TIL and dropped to zero for 5 times.



**Fig. 5.8: Throughput of dropping packets for 10 nodes**

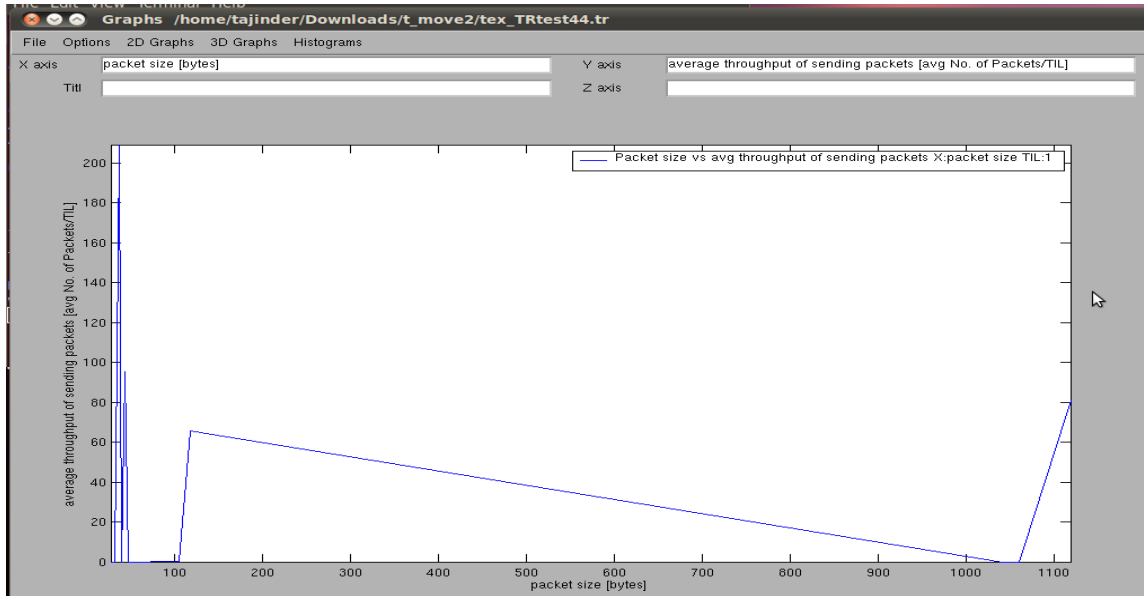
INFERENCE FROM FIG. 5.8: Here from this graph it can be easily analysed that number of packets dropped has increased constantly to 18 packets/TIL in just 0.016secs (10.038-10.054secs).



**Fig. 5.9: Throughput of dropping packets for 25 nodes**

INFERENCE FROM FIG. 5.9: This graphs shows that throughput of drooping packets has increased to 350 packets/TIL in first 2 secs and then dropped to zero in next one sec. Then number of packets dropped per unit time remains to be at zero for about 7 secs. After which it again rises to 200 packets/TIL.

#### 5.4 CASE 4: Packet Size vs Average throughput of sending packets

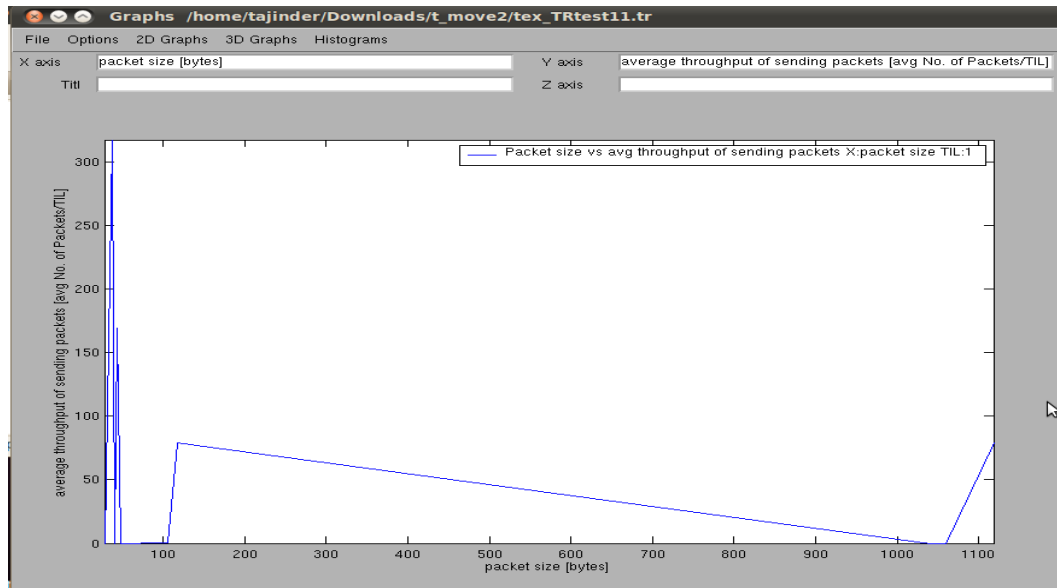


**Fig. 5.10: Packet Size vs Average throughput of sending packets for 4 nodes**

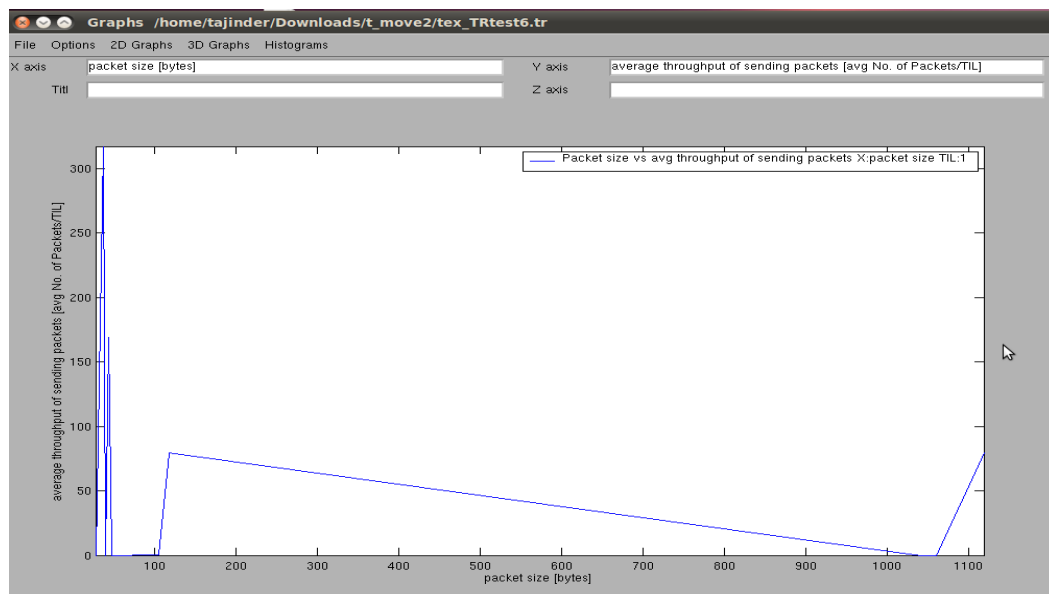
INFERENCE FROM FIG. 5.10: This graph shows that average throughput of small sized packets (1-20 bytes) is more (205 packets/TIL) than large sized packets 1000 bytes)

INFERENCE FROM FIG. 5.11: This graph is almost similar to the graph for 4 nodes with a difference that here average throughput of small sized packets has reached to 320 packets/TIL.

INFERENCE FROM FIG. 5.12: This graph is almost similar to the graph for 4 nodes and 10 nodes. Here also throughput is maximum for small sized packets and it drops to zero for packet size of range 20-100 bytes, after which it increases to 70 packets/TIL for a packet size of 110 bytes and it drops constantly to zero as packet size increases to 1050 bytes.



**Fig. 5.11: Packet Size vs Average throughput of sending packets for 10 nodes**



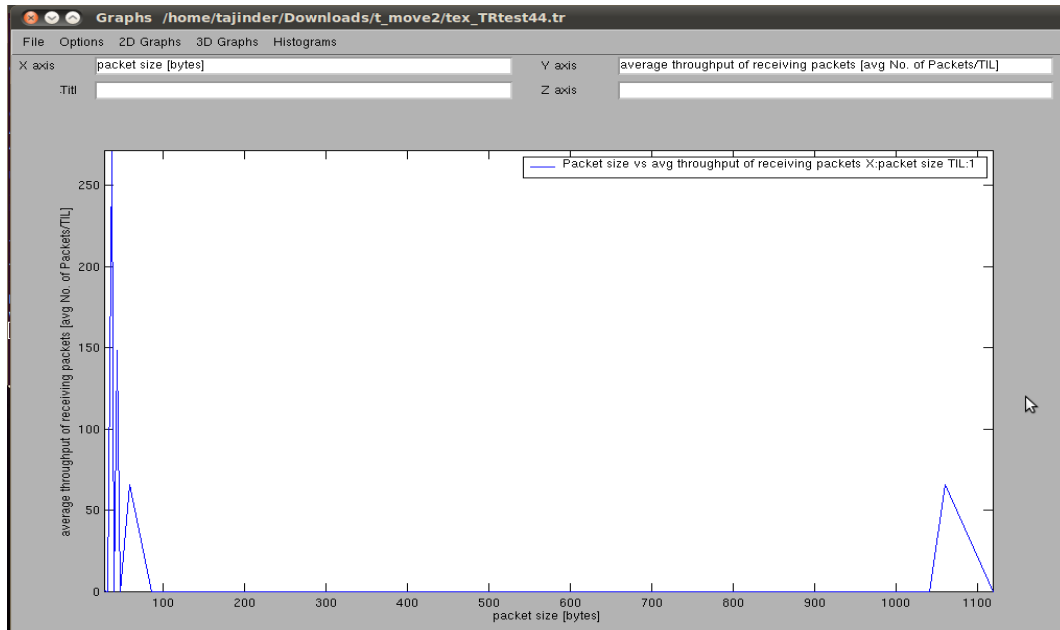
**Fig. 5.12: Packet Size vs Average throughput of sending packets for 25 nodes**

## 5.5 CASE 5: Packet Size vs Average throughput of receiving packets

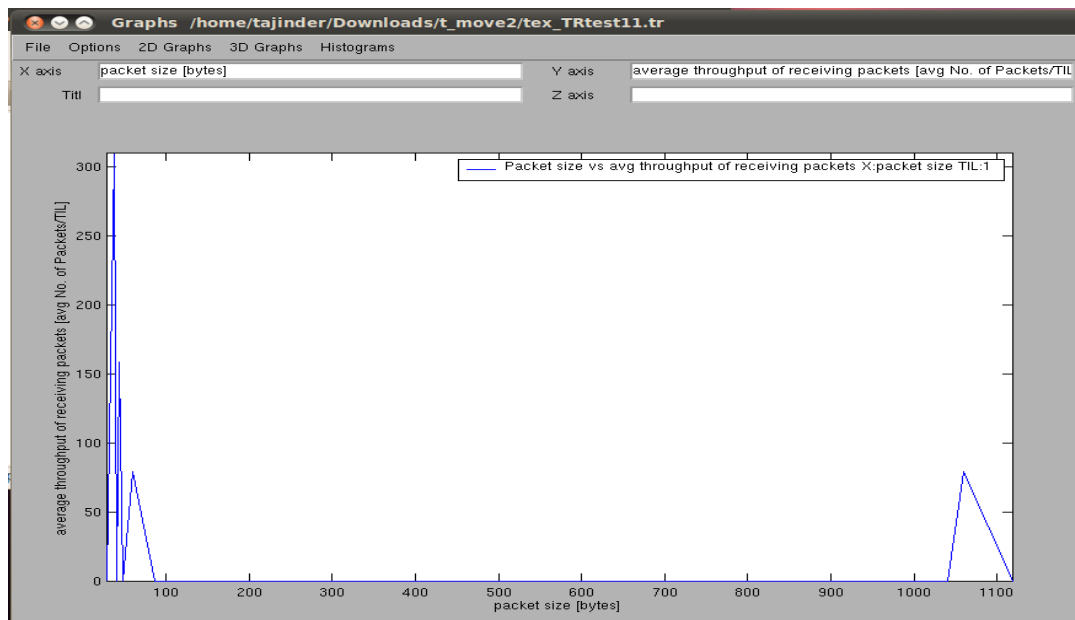
INFERENCE FROM FIG. 5.13: This graph shows that throughput is maximum for small sized packets and zero for packet size of range 90-1030 bytes.

INFERENCE FROM FIG. 5.14: This graph is almost same as for 4 nodes.

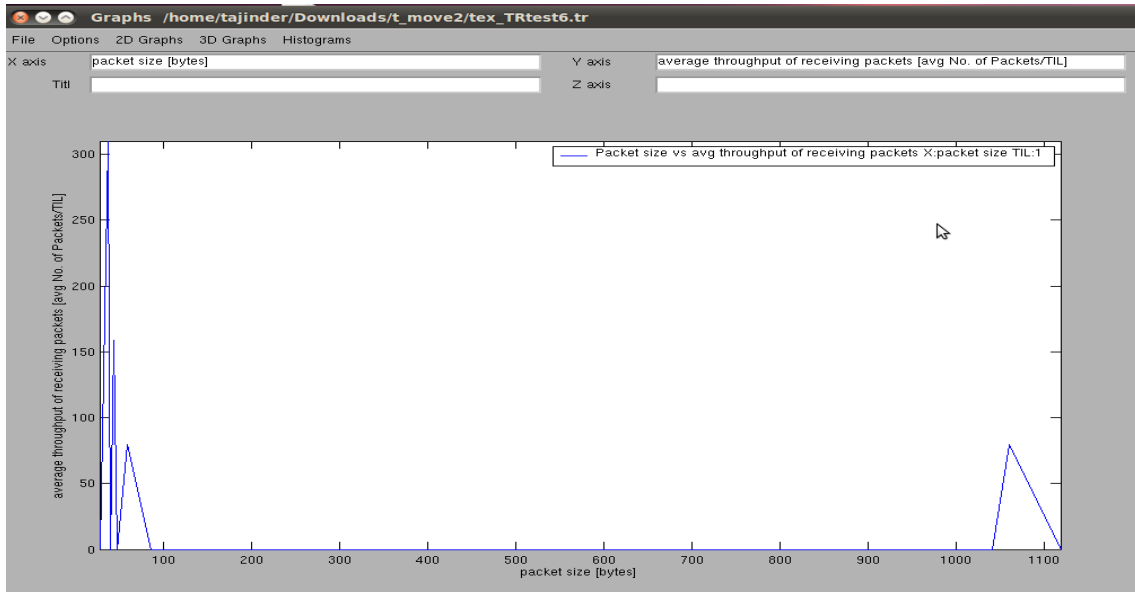
INFERENCE FROM FIG. 5.15: This is also same as for 4 and 10 nodes.



**Fig. 5.13: Packet Size vs Average throughput of receiving packets for 4 nodes**

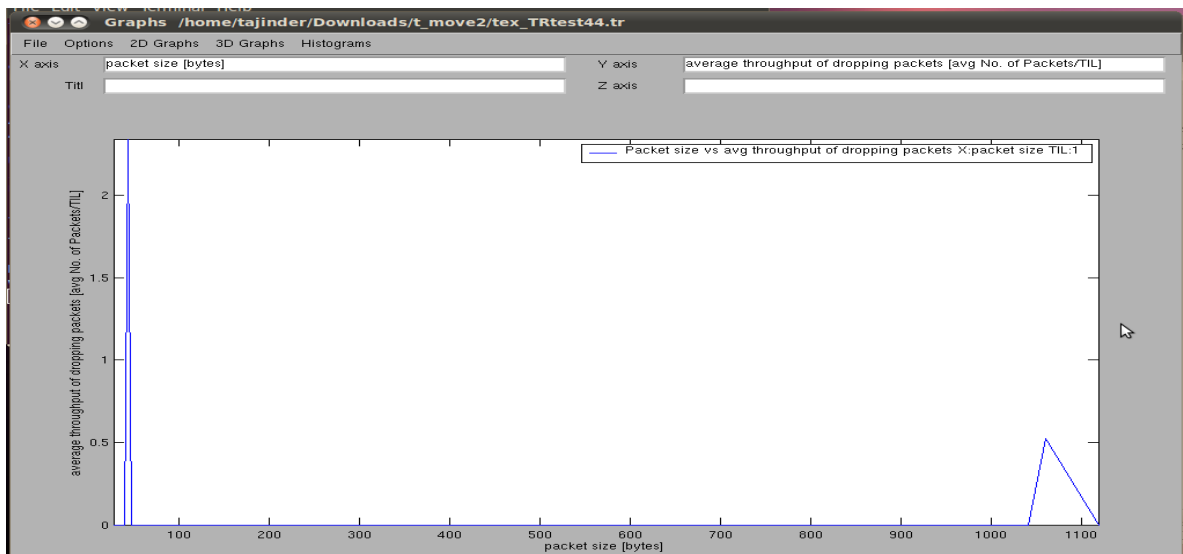


**Fig. 5.14: Packet Size vs Average throughput of receiving packets for 10 nodes**



**Fig. 5.15: Packet Size vs Average throughput of receiving packets for 25 nodes**

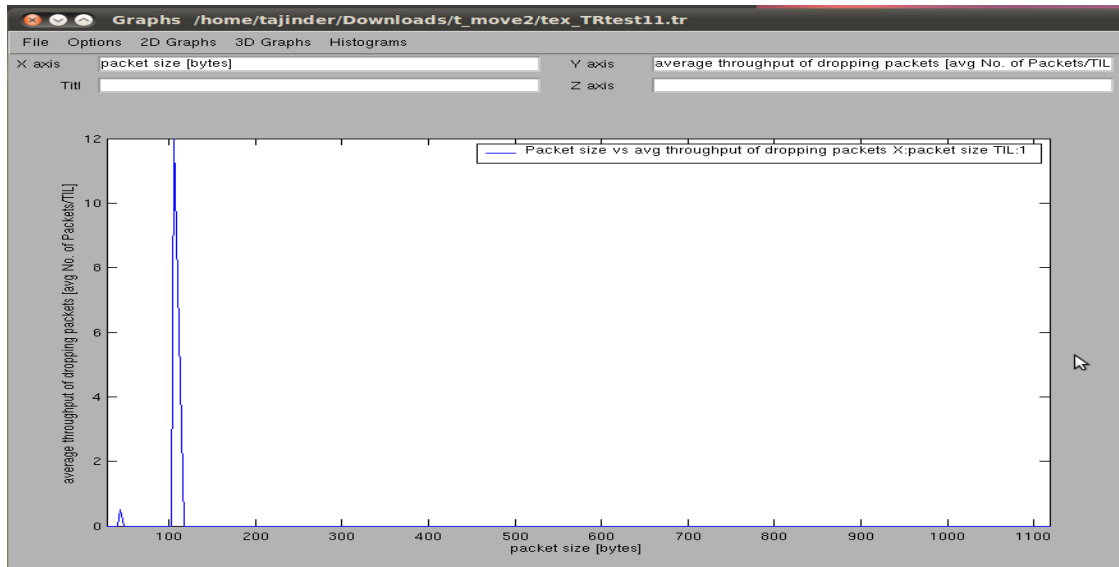
## 5.6 CASE 6: Packet Size vs Average throughput of dropping packets



**Fig. 5.16: Packet Size vs Average throughput of dropping packets for 4 nodes**

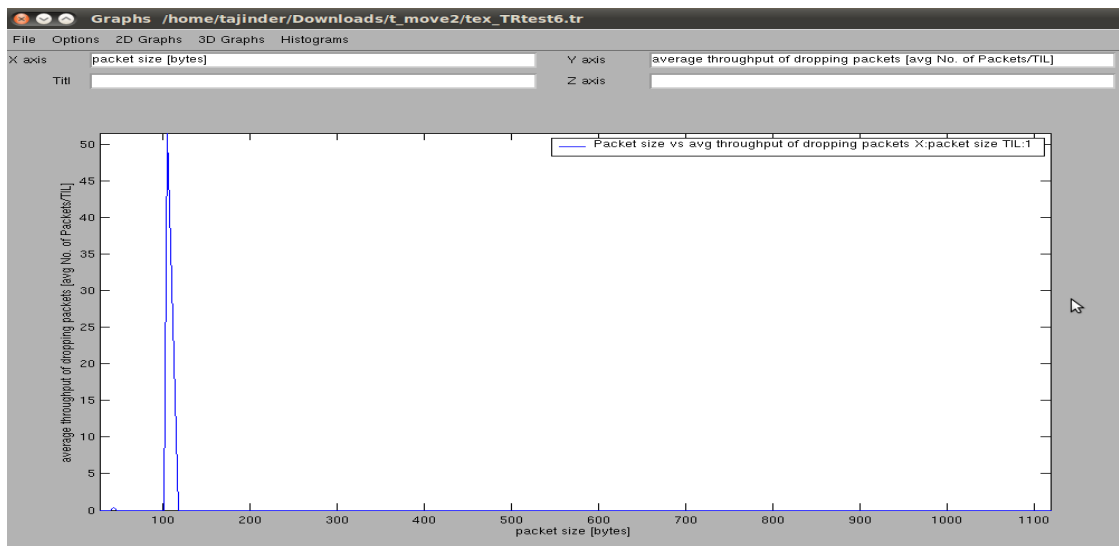
INFERENCE FROM FIG. 5.16: Here it can be analysed that average throughput of dropping packets is maximum for vry small range of packet size i.e. 10-20 bytes. And for packet size in range 20-1040 bytes there are no packets dropped which indicate that these packet size are

not used by the network. There is a small number packets dropped for a packet size of 1050-1100 bytes.



**Fig. 5.17: Packet Size vs Average throughput of dropping packets for 10 nodes**

INFERENCE FROM FIG. 5.17: This graph shows a difference in the throughput interval than for 4 nodes. Here it has reached to 12 packets/TIL for packet size of 100-120 bytes. There is also one small peak where throughput is 0.5 packets/TIL for packet size 10-20 bytes.



**Fig. 5.18: Packet Size vs Average throughput of dropping packets for 25 nodes**

INFERENCE FROM FIG. 5.18: This graph shows a difference in the throughput interval than for 4 and 10 nodes. Here it has reached to more than 50 packets/TIL for packet size of 100-120 bytes. There is no other small peak.

### 5.7 CASE 7: Throughput of sending bits vs Average simulation End2End delay

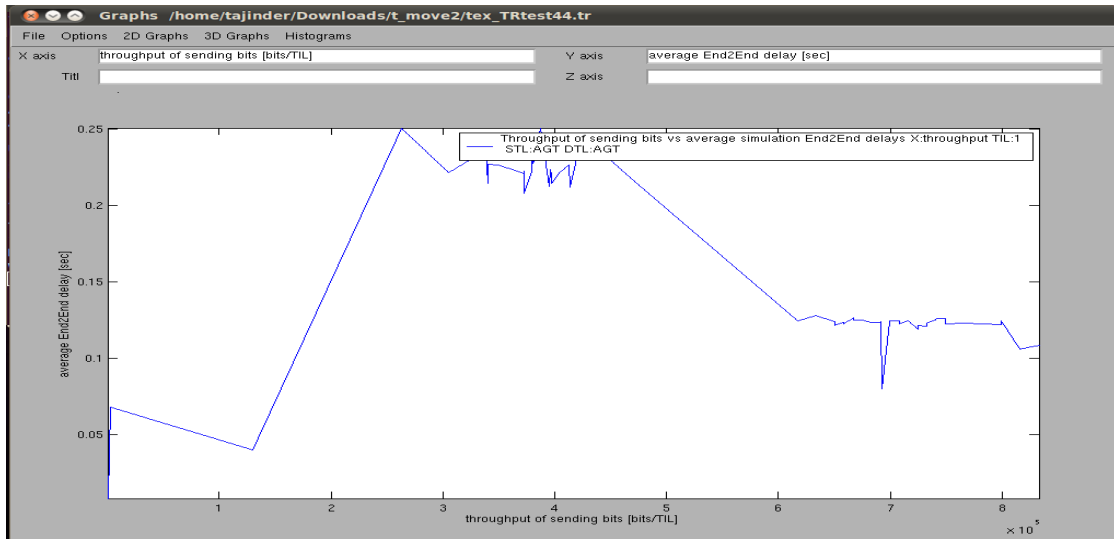


Fig. 5.19: Throughput of sending bits vs Average simulation End2End delay for 4 nodes

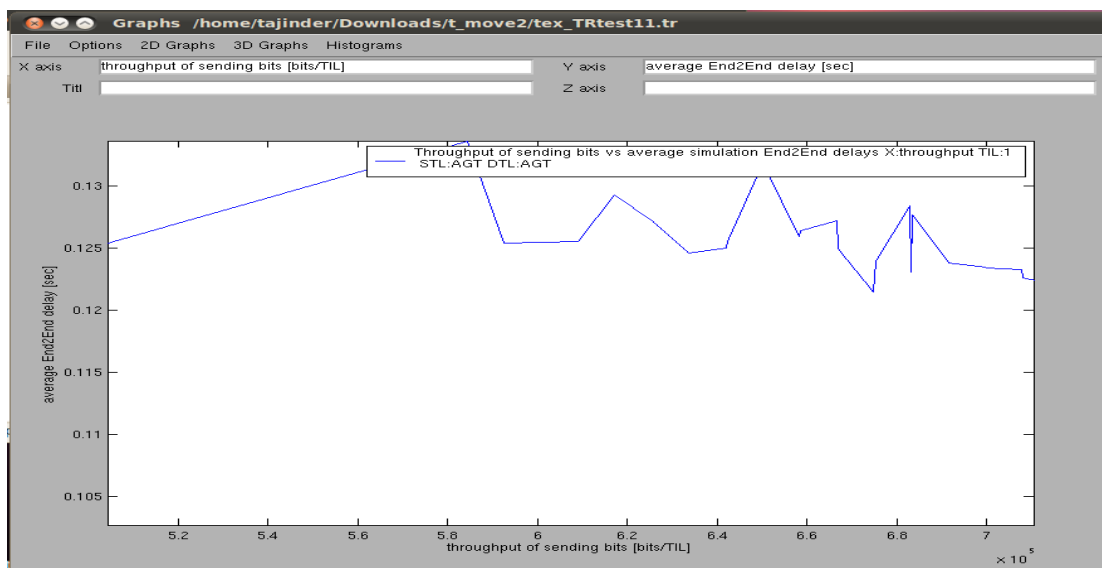
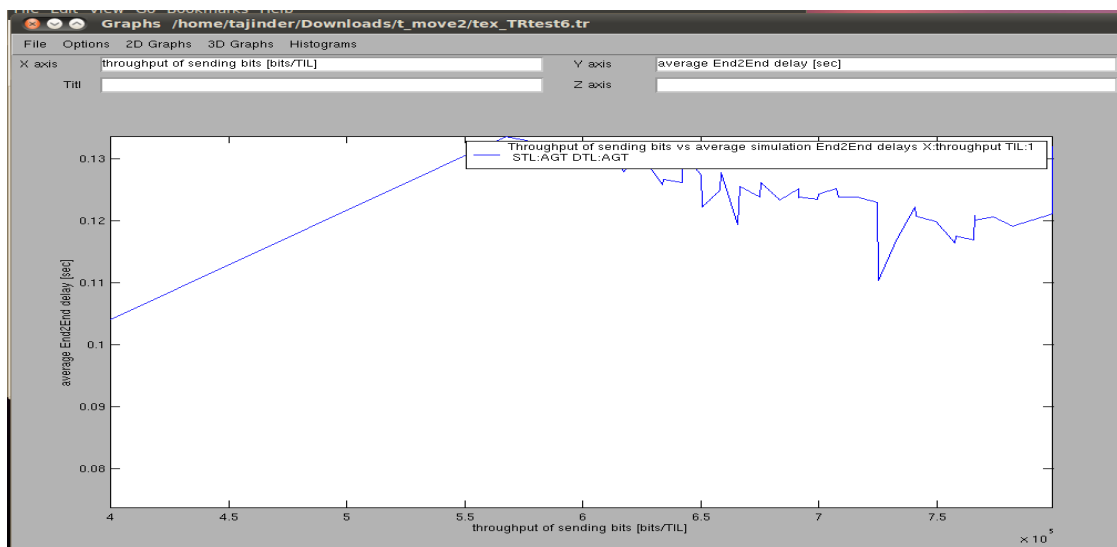


Fig. 5.20: Throughput of sending bits vs Average simulation End2End delay for 10 nodes

INFERENCE FROM FIG. 5.19: Here the delay time interval is for few milli seconds. Within this interval, delay is maximum for a throughput range between 2.5-4.5(bits/TIL). This may be due to more congestion in the network. After this, average delay time decreases to 0.12secs at 6 packets/TIL throughput and remains almost consistent afterwards. *This can be understood as if when throughput of sending packets increases, there are more packets which get dropped which results in decrease in congestion in the network and decrease in delay time.*

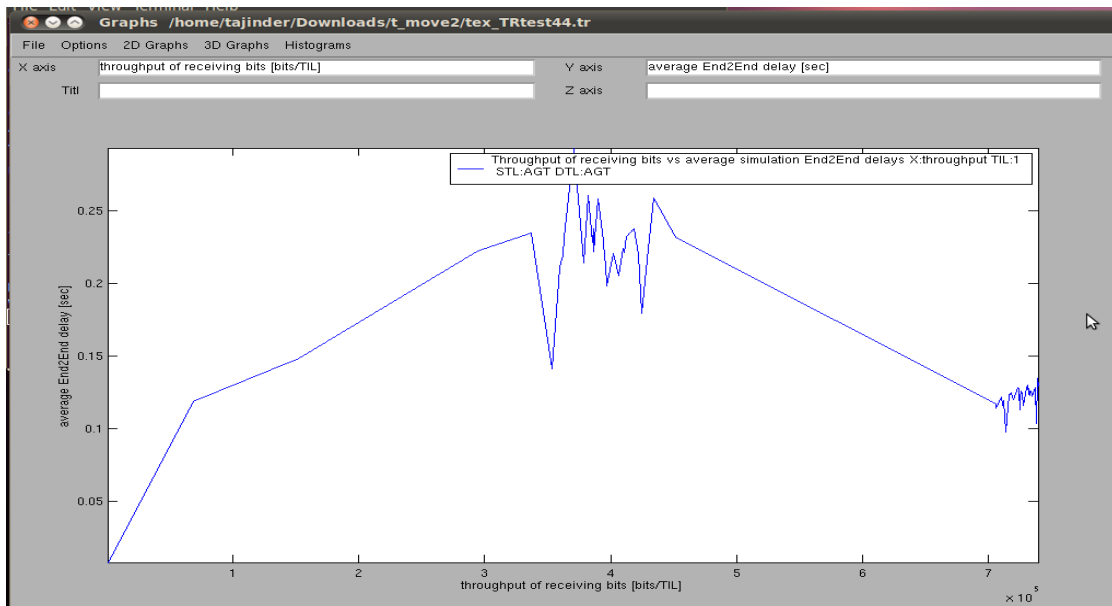
INFERENCE FROM FIG. 5.20: As compared to for 4 nodes here maximum delay is 0.135secs which is reached when the throughput reaches 5.8 packets/TIL and as the throughput of sending bits increases, delay time remains in the range of 0.13-0.123secs.



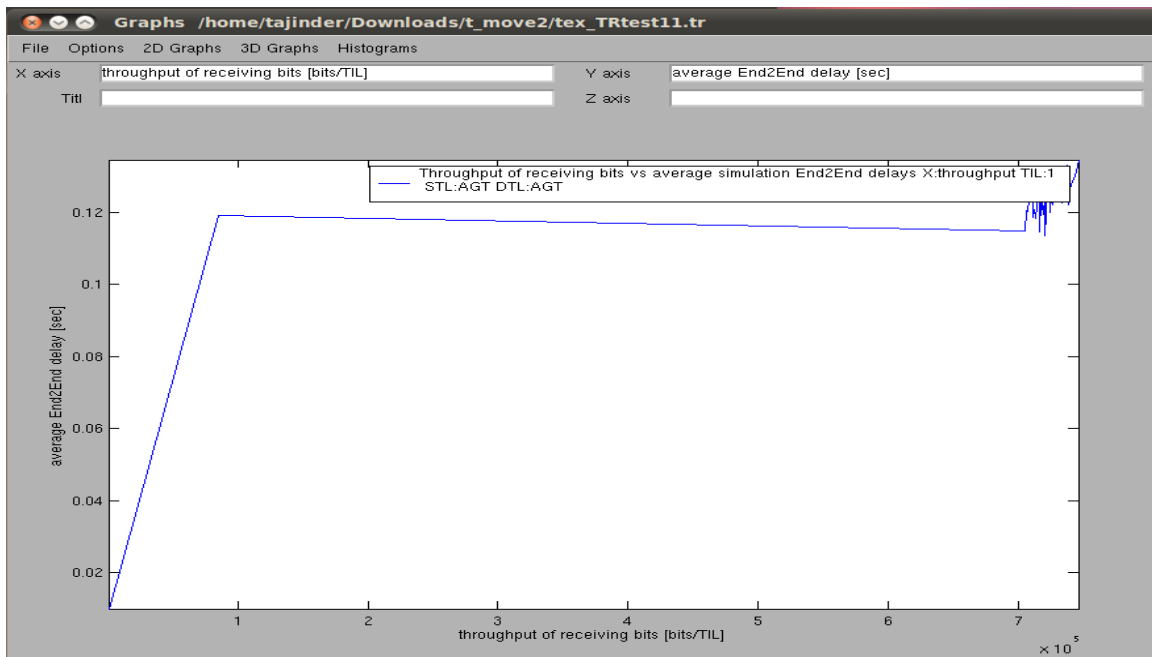
**Fig. 5.21 Throughput of sending bits vs Average simulation End2End delay for 25 nodes**

INFERENCE FROM FIG. 5.21: Here delay time increases constantly from 0.06 secs to 0.13 secs as throughput of sending bits reaches to 570000 bits/TIL and after that it remains in the range 0.13-0.11 secs.

## 5.8 CASE 8: Throughput of receiving bits vs average simulation End2End delay



**Fig. 5.22: Throughput of receiving bits vs Average simulation End2End delay for 4 nodes**

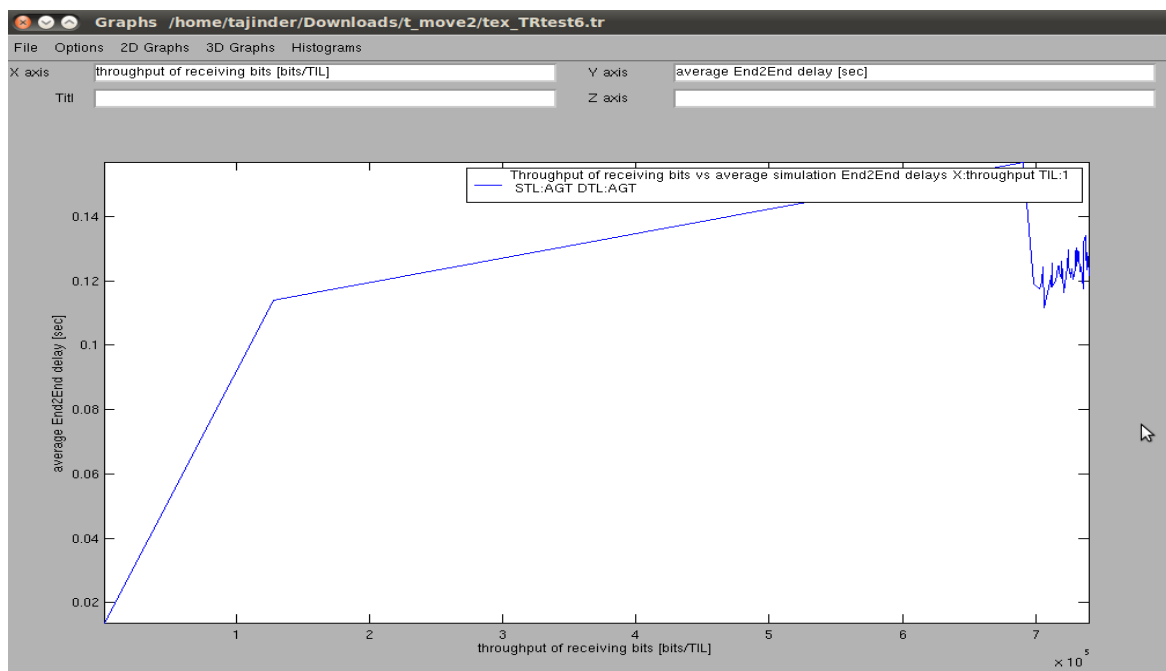


**Fig. 5.23: Throughput of receiving bits vs Average simulation End2End delay for 10 nodes**

INFERENCE FROM FIG. 5.22: From this graph it can be inferred that as number of bits received per unit TIL increases, delay time also increases concurrently till throughput reaches 325000 bits/TIL. Then for the throughput range 325000-450000 bits/TIL delay time varies a lot and decreases afterwards which indicates a good network has been established.

INFERENCE FROM FIG. 5.23: From this graph it can be inferred that delay time increases from 0 to 0.12secs for a throughput of 100000 bits/TIL. This can be understood as some time is required to establish a network in the starting therefore it took more time to find the route to the destination node. But after this, delay time is not much affected by the increase in number of bits received.

INFERENCE FROM FIG. 5.24: From this graph it can be inferred that as number of bits received per unit TIL increases to 1.25 bits/TIL delay time also increases rapidly till it reaches to 0.12 secs but after this the delay time increases slowly. This means delay is not affected much by the throughput.



**Fig. 5.24: Throughput of receiving bits vs Average simulation End2End delay for 25 nodes**

## CHAPTER 6

### CONCLUSION & FUTURE SCOPE

---

VANETs is an active research field these days. Various types of challenges in vehicular communications have been identified and addressed. This report explores the different mobility models and their classifications according to the level of details they generate. Further, a large number of routing protocols have been proposed for VANET and an in-depth analysis of AODV routing protocol for VANET has been carried out in this thesis.

In this thesis, AODV is simulated with realistic mobility model. For this MOVE is used along with NS2 and SUMO. Then graphs are plotted using Tracegraph for evaluation. AODV's performance is analysed for three different number of nodes i.e. 4, 10 and 25 nodes with respect to various parameters like throughput, packet size, packet drops, delay time etc. The simulation results for various cases can be summarized as below:

**CASE 1: Throughput of sending packets:** Results shows that for lesser no. of nodes i.e. 4 nodes, throughout drops with time in steps, but for more nodes like 10 and 25 ,throughput of sending packets is almost uniform.

**CASE 2: Throughput of receiving packets:** Results shows that throughput of receiving packets becomes more uniform with increase in number of nodes.

**CASE 3: Throughput of dropping packets:** Results shows that number of packets dropped in intial few secs is more in a network where number of nodes are more like in case of 25 ,it has reached to 350. While for fewer nodes like 4, it is quite less (less then 5 approx.)

**CASE 4: Packet Size vs Average throughput of sending packets:** Average throughput of sending packets is not affected much by the change in number of nodes.

**CASE 5: Packet Size vs Average throughput of receiving packets:** Average throughput of receiving packets is not affected by the change in number of nodes in a network.

**CASE 6: Packet Size vs Average throughput of dropping packets:** Average throughput of dropping packets is affected by the change in number of nodes in a network. It increases with increase in number of nodes.This means more number of small sized packets are dropped in a network with larger number of nodes.

**CASE 7: Throughput of sending bits vs average simulation End2End delay:** From results it can be concluded that there is more delay in a network of lesser number of nodes like 4 nodes than the delay in 10 and 25 nodes network.

**CASE 8: Throughput of receiving bits vs average simulation End2End delay:** It can be concluded from the results that delay time increases in the beginning of the network establishment to a point (0.12 secs) and then it is not effected much by the throughput of receiving bits in case of 10 and 25 nodes. But, in case of 4 nodes it increases till 0.25 secs. This might be because there are very few nodes in the network to communicate well with each other.

In future, it can be simulated and analyzed for higher number of nodes upto 50 and 100. It would be interesting to see how AODV performs in high node density network. Here it has been implemented for single mobility model and manually generated maps. Further, the performance of AODV can be compared for different mobility models and also, for different maps, such as: random maps, spider topology and maps imported from TIGER database [47].

- 
- [1] Jim Geier, in *Wireless Networks first-step*, Cisco Press, August 03.2004.
- [2] Rob Flickenger, in *Building Wireless Community Network*, 2nd Edition, O'Reilly Media, June 2003.
- [3] M. Mauve, A. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *Network, IEEE*, vol. 15, no. 6, pp. 30 - 39, Nov/Dec 2001.
- [4] <http://www.car-accidents.com/pages/fatal-accident-statistics.html>
- [5] H.P. Glathe, L. Karlsson, G.P. Brusaglino, L. Calandrino, "The PROMETHEUS Programme - Objectives, Concepts and Technology for Future Road Traffic", *XXIII FISITA Congress*, May 1990, pp. 477-484
- [6] Vanet Simulator, Report for the Computer Security exam at the Politecnico di Torino Walter Dal Mut, Armand Sofack.
- [7] Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "A stable routing protocol to support its services in vanet networks", *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337–3347, November 2007.
- [8] Jinyuan, S., Chi, Z. & Yuguang, F., "An ID-based Framework Achieving Privacy and Non-Repudiation", in *Proceedings of IEEE Vehicular Ad Hoc Networks, Military Communications Conference (MILCOM 2007)*, pp. 1 – 7, October 2007.
- [9] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan, "Vehicular Ad Hoc Networks (VANETs): Status, Results, and Challenges," *Telecommunication System*, Vol. 51, Issue 2&3, October/November 2010
- [10] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems – 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ASTM E2213-03, Sep 2003
- [11] Kudoh, Y., "DSRC Standards for Multiple Applications," in *Proceedings of 11th World Congress on ITS*, Nagoya, Japan, 2004.
- [12] IEEE, "IEEE Std. 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE Standard 802.11, 2007.

- [13] "IEEE P802.11p/D3.0, Draft Amendment for Wireless Access in Vehicular Environments (WAVE)," July 2007
- [14] Fan Li; Yu Wang;, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE* , vol.2, no.2, pp.12-22, June 2007.
- [15] Lee U., Zhou B, Gerla, M., Magistretti, E., Bellavista, P.; Corradi, A. (2006), "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network," *Wireless Communications, IEEE* , vol.13, no.5, pp.52-57, October 2006.
- [16] Pei, G., Gerla, M., and Chen, T.-W. (2000), "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *Proc. ICC 2000*, New Orleans, LA, June 2000.
- [17] Perkins, C. E. and Royer, E. M. (1999), "Ad-Hoc On-Demand Distance Vector Routing," *Proc. IEEE WMCSA '99*, New Orleans, LA, pp. 90–100, Feb. 1999.
- [18] Lee, K. C., Lee, U., & Gerla, M. (2010), "Survey of Routing Protocols in Vehicular Ad Hoc Networks", *Advances in Vehicular Ad-Hoc Networks: Developments and Challenge*, Watfa, M. (Ed.), (pp. 149-170), 2010.
- [19] Naumov, V., Baumann, R., Gross, T. (2006), "An evaluation of Inter-Vehicle Ad Hoc Networks Based on Realistic Vehicular Traces," *Proc. ACM MobiHoc '06 Conf.*, May, 2006.
- [20] Johnson, D. B. and Maltz, D. A. (1996), "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, T. Imielinski and H. Korth, Eds., Ch. 5, Kluwer, pp. 153–81, 1996.
- [21] Park, V.D., Corson, M.S. (1997), "A highly adaptive distributed routing algorithm for mobile wireless networks," *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* , vol.3, no., pp.1405-1413 vol.3, 7-12 Apr 1997.
- [22] Flury, R. and Wattenhofer, R. (2006), "MLS: an efficient location service for mobile ad hoc networks." *In MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 226–237, New York, NY, USA, 2006.
- [23] Aamir Hassan, "VANET Simulation", Technical report, IDE0948, May 2009.

- [24] C. Perkins, E. Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF, RFC 3561, 2003.
- [25] Royer, E.M.; Perkins, C.E.; , "An implementation study of the AODV routing protocol," *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE* , vol.3, no., pp.1003-1008 vol.3, 2000.
- [26] C. Perkins and P. Bhagwat. "Routing over Multihop Wireless Network of Mobile Computers." *SIGCOMM 1994: Computer Communications Review*, 24(4): 234-244, Oct.1994.
- [27] Johnson, D. B. and Maltz, D. A. (1996), "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, T. Imielinski and H. Korth, Eds., Ch. 5, Kluwer, 1996, pp. 153-81.
- [28] Marc Greis. "Ns Tutorial."Internet: <http://www.isi.edu/nsnam/ns/tutorial/index.html> [Dec, 2011].
- [29] "MObility model generator for VEhicular networks (MOVE)." Internet: <http://www.cs.unsw.edu.au/klan/move/> [Feb, 2012].
- [30] "Simulation of Urban MObility (SUMO)." Internet: <http://sumo.sourceforge.net/> [Feb, 2012].
- [31] "OPNET Simulator." Internet: <http://www.opnet.com/> [Feb, 2012].
- [32] "QualNet Network Simulator." Internet: <http://www.scalablenetworks.com/> [Feb, 2012].
- [33] "Pamramics: Microscopic traffic Simulation." Internet: <http://www.paramics-online.com/> [Feb, 2012].
- [34] "CORSIM: Microscopic Traffic Simulation Model," Internet: <http://mctrans.ce.ufl.edu/featured/tsis/version5/corsim.htm> [Feb, 2012].
- [35] Al-Sakib Khan Pathan. "Visual Simulator Tool – ViSim 1.0." Internet: <http://sites.google.com/site/spathansite/research/visim-simulation-tool> [Feb, 2012].
- [36] "VanetMobiSim." Internet: <http://www.eurecom.fr/fr/publication/3390/detail/vehicular-mobility-simulation-with-vanetmobisim> [Feb, 2012].

- [37] Amit Kumar Saha and David B. Johnson. "Modeling mobility for vehicular ad hoc networks." *In Proc. Of the 2nd ACM International Workshop on Vehicular Ad hoc Networks (VANET)*, October 2004.
- [38] John Heidemann, Nirupama Bulusu, Jeremy Elson, Chalermek Intanagonwiwat, Kun chan Lan, Ya Xu, Wei Ye, Deborah Estrin, and Ramesh Govindan. "Effects of detail in wireless network simulation." *In Proc. Of Communication Networks and Distributed Systems Modeling and Simulation Conference*, January 2001.
- [39] "PPA for Wouter Horré." Internet: [https://launchpad.net/~wouterh/+archive/ppa?field.series\\_filter=lucid](https://launchpad.net/~wouterh/+archive/ppa?field.series_filter=lucid) [Jan, 2012].
- [40] Valeri Naoumov, Thomas Gross, "Simulation of Large Ad Hoc Networks," *In Proc. of The Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2003) in conjunction with ACM MobiCOM 2003*, September 2003.
- [41] "NAM Network Animator". Internet: <http://www.isi.edu/nsnam/nam/> [March, 2012].
- [42] Paul Meenaghan and Declan Delaney, "An Introduction to NS, Nam and OTcl scripting," <http://www.cs.nuim.ie/research/reports/2004/nuim-cs-tr-2004-05.pdf>.
- [43] "SUMO GUI." Internet: <http://sumo.sourceforge.net/doc/current/docs/userdoc/SUMO-GUI.html> [March, 2012].
- [44] Rapid Generation of Realistic Simulation for VANET Manual Updated 12 June 2011. Available at: <http://lens1.csie.ncku.edu.tw/MOVE/Example%20for%20step-by-step.pdf>
- [45] <http://www.ubuntuupdates.org/package/core/lucid/main/security/openjdk-6-jdk> accessed on Feb, 2012
- [46] <http://ubuntuforum.blogspot.in/2009/01/trace-graph-ns2-for-linux.html> accessed on 24th April, 2012.
- [47] U.S. Census Bureau - Topologically Integrated Geographic Encoding and Referencing (TIGER) system, <http://www.census.gov/geo/www/tiger>.

### **Published**

- Tajinder Kaur, Karamjit Cheema and A. K. Verma, “A Review of Routing Protocols in VANETs ”, *Advances in Communication, Navigation and Computer Networks (ACNCN-2012)*, Andhra Pradesh, India, March 17-18, 2012.

### **Communicated**

- Tajinder Kaur and A. K. Verma, “Simulation and Analysis of AODV Routing Protocol in VANETs”, *International Journal of Soft Computing and Engineering*, vol.-2, issue-3, July 2012.[Accepted]
- Tajinder Kaur, Avleen Kaur Malhi and A. K. Verma, “Simulation and Comparison of AODV and DSR Routing Protocols in VANETs”, *2<sup>nd</sup> International Conference on Issues and Challenges in Networking, Intelligence and Computing Technologies (ICNICT-2012)* , Ghaziabad, India, September 7-8, 2012.