SIMULATION AND ANALYSIS OF DSR PROTOCOL IN VANETs

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

Master of Engineering
in
Computer Science & Engineering

By:
Avleen Kaur Malhi
(801032004)

Under the supervision of:
Dr. A. K. Verma
Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

MAY 2012

I hereby certify that the work which is being presented in the thesis entitled, **"Simulation And Analysis Of DSR 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* 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.

(Avleen Kaur Malhi)

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

(Dr. A. K. Verma)
Associate Professor
Computer Science & Engineering Department
Thapar University
Patiala.

Countersigned by:

(Dr. Maninder Singh)
Assoc. Professor & Head
Computer Science & Engineering Department,
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. A. K. Verma****, Associate Professor, Computer Science and Engineering Department, Thapar University, who has been very concerned and has aided for all the material essential for the preparation of this thesis report. He has helped me to explore this vast topic 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 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.

Avleen Kaur Malhi
801032004

Abstract

VANET (Vehicular Adhoc Network) is a new concept in the field of wireless networks. The main objective of VANET is to build a powerful network between mobile vehicles so that the vehicles can talk to each other for the safety of the humans. The various protocols are being used in VANET environment. The aim of work is to simulate and analyze the DSR (Dynamic Source Routing) protocol in VANET. DSR protocol is simulated under realistic scenario with the help of mobility models. Mobility models, or the movement patterns of nodes communicating wirelessly, play a vital role in the simulation-based evaluation of Vehicular Ad Hoc Networks (VANETs). The work has been carried out with the help of open-source simulation tools on realistic scenario of traffic. The traffic is simulated with the help of simulator ns-2 and SUMO. MOVE allows the rapid generation of mobility models for the use in VANET simulation. The performance of DSR protocol is tested under different parameters such as packet size, throughput and the number of dropped packets during the simulation and results are also compared by varying the number of nodes used for simulation. The analysis indicate that there is varying effect on the various parameters used as the number of nodes are changed.

Keywords: VANET, DSR, NS-2, SUMO, MOVE.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACK	Acknowledgement
AGF	Advanced Greedy Forwarding
AODV	Ad-Hoc On Demand Distance Vector
CAR	Connectivity-Aware Routing
CBR	Continuous Bit Rate
CCK	Complementary Code Keying
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communications
EMM	Entity Mobility Model
FSR	Fisheye State Routing
FTP	File Transfer Protocol
GEOppS	Geographic Opportunistic Routing for Vehicular Networks
GMM	Group Mobility Model
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
ITS	Intelligent Transportation System
LAN	Local Area Network
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
MIMO	Multiple Input/Multiple Output
MM	Mobility Model
MOVE	MObility model generator for VEhicular networks
NAM	Network Animation
NS	Network Simulator
RFC	Request For Comments
RPGMM	Reference Point Group Mobility Model
RREP	Route REPly

RREQ	Route REQuest
RWPMM	Random Waypoint Mobility Model
STAR	Source Tree Adaptive Routing
SUMO	Simulation of Urban Mobility
TCL	Tool Command Language
TCP	Transmission Control Protocol
TORA	Temporally Ordered Routing Algorithm
UDP	User Datagram Protocol
VANET	Vehicular Ad-Hoc Network
WAVE	Wireless Access in Vehicular Environments
WLAN	Wireless local Area Network
WSN	Wireless Sensor Network

CHAPTER 1
INTRODUCTION

1.1 WIRELESS NETWORKS

Wireless networks [13] refer to any kind of networking that does not involve cables. It is a technique that helps entrepreneurs and telecommunications networks to save the cost of cables for networking in specific premises in their installations. The transmission system is usually implemented and administrated via radio waves where the implementation takes place at physical level. An example of wireless networks is shown in fig. 1.1.

Fig. 1.1: Wireless Networks[18]

A wireless network[13] uses radio waves, just like cell phones, televisions and radios do. In fact, communication across a wireless network is a lot like two-way radio communication. The procedure is:

- A computer's wireless adapter[5] translates data into a radio signal and transmits it using an antenna.
- A wireless router[5] receives the signal and decodes it. The router sends the information to the Internet using a physical, wired Ethernet connection.

The process also works in reverse, with the router receiving information from the Internet, translating it into a radio signal and sending it to the computer's wireless adapter.

Following reasons favour the use of wireless networking over traditional wired networks :

- Running additional wires or drilling new holes in a home or office could be prohibited (because of rental regulations), impractical (infrastructure limitations), or too expensive.

- Flexibility of location and data ports is required.
- Roaming capability is desired; e.g., maintaining connectivity from almost anywhere inside a home or business.
- Network access is desired outdoors; e.g., outside a home or office building.

1.1.1 The Benefits

Small businesses can experience many benefits from a wireless network, including:

- Convenience. Access your network resources from any location within your wireless network's coverage area or from any WiFi hotspot.
- Mobility. You're no longer tied to your desk, as you were with a wired connection. You and your employees can go online in conference room meetings, for example.
- Productivity. Wireless access to the Internet and to your company's key applications and resources helps your staff get the job done and encourages collaboration.
- Easy setup. You don't have to string cables, so installation can be quick and cost-effective.
- Expandable. You can easily expand wireless networks with existing equipment, while a wired network might require additional wiring.
- Security. Advances in wireless networks provide robust security protections.
- Cost. Because wireless networks eliminate or reduce wiring costs, they can cost less to operate than wired networks.

1.1.2 Wireless Networks usage

As we know that Wireless network is usually related with a telecommunications that works between nodes and executed without the use of wires. The usage of wireless networking increasing day by day because it has influenced significant impact on the world therefore its uses have appreciably grown-up.

- Radio frequency signals used in a wireless network therefore you can move about and get admittance to the network while you are working an outdoor location.
- Through Wireless Networks you can send information over the world using satellites and other signals.
- Now days wireless networks used in emergency services like police department where wireless network utilize to commune significant information speedily.
- The growth of wireless network increasing both in people and businesses to send and share data swiftly It doesn’t matter be in a small office or across the world.

- Another vital exercise for wireless networks is as a cheap and fast way to be linked to the Internet in regions especially where telecom transportation is meager and no source for communication.
- To make use of Wireless Networks you can get access to other network resources like Library Online System because to move your laptop anywhere is not enough difficult now. Wireless networks make easy of file sharing, the use of printer and other documents with high security.

1.2 IEEE 802.11 Specifications

The IEEE (Institute of Electrical and Electronic Engineers)[19] released the 802.11 specifications in June 1999. The initial specification, known as 802.11, used the 2.4 GHz frequency and supported a maximum data rate of 1 to 2 Mbps. The 802.11b specification increased the performance to 11 Mbps in the 2.4 GHz range while the 802.11a specification utilized the 5 GHz range and supported up to 54 Mbps. Unfortunately, the two new specifications were incompatible because they used different frequencies. This means that 802.11a network interface cards (NICs) and access points cannot communicate with 802.11b NICs and access points. This incompatibility forced the creation of the new draft standard known as 802.11g. 802.11g supports up to 54 Mbps and is interoperable with 802.11b products on the market today.

1.2.1 IEEE 802.11a

802.11a operates in the 5 - 6 GHz range with data rates commonly in the 6 Mbps, 12 Mbps, or 24 Mbps range. Because 802.11a uses the orthogonal frequency division multiplexing (OFDM) standard, data transfer rates can be as high as 54 Mbps. OFDM breaks up fast serial information signals into several slower sub-signals that are transferred at the same time via different frequencies, providing more resistance to radio frequency interference. Though regionally deployed, it is not a global standard like 802.11b.

1.2.2 IEEE 802.11b

The 802.11b standard (also known as Wi-Fi) operates in the 2.4 GHz range with up to 11 Mbps data rates and is backward compatible with the 802.11 standard. 802.11b uses a technology known as complementary code keying (CCK) modulation, which allows for higher data rates with less chance of multi-path propagation interference (duplicate signals bouncing off walls).

1.2.3 IEEE 802.11g

802.11g is the most recent IEEE 802.11 draft standard and operates in the 2.4 GHz range with data rates as high as 54 Mbps over a limited distance. 802.11g offers the best features of both 802.11a and 802.11b, but as of the publication date of this document, this standard has not yet been certified, and therefore is unavailable.

1.2.4 IEEE 802.11n

The draft 802.11n standard defines a new physical layer for increasing the throughput of wireless local area networks. 802.11n is based on MIMO (multiple input/multiple output) OFDM technology, which allows the transmission of up to 100 Mbps over a much wider range than earlier versions. MIMO uses multiple transmitters and receivers to allow for increased throughput through spatial multiplexing and increased range.

1.2.5 IEEE 802.11p

IEEE 802.11p is an approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments (WAVE). It defines enhancements to 802.11 (the basis of products marketed as Wi-Fi) required to support Intelligent Transportation Systems (ITS) applications. This includes data exchange between high-speed vehicles and between the vehicles and the roadside infrastructure in the licensed ITS band of 5.9 GHz (5.85-5.925 GHz). 802.11p will be used as the groundwork for Dedicated Short Range Communications (DSRC).

Table 2.1 Comparison of IEEE 802.11 standards

Factors	**802.11a**	**802.11b**	**802.11g**	**802.11n**	**802.11p**
Range	5-6GHz	2.4 GHz	2.4GHz	OFDM	5.9GHz
Data Rates(Mbps)	6,12,24, as high as 54 mbps	Upto 11 mbps	Upto 54 mbps	100 mbps	Very high data rates
Features	Regionally deployed	W-Fi	Not yet certified	Uses spatial multiplexing	Dedicated short range communication
Methodology	OFDM	CCK	Best features of 802.11a and 802.11b	MIMO	WAVE

1.3 Types

The 802.11 standard offers two methods for configuring a wireless network

- Ad Hoc[7][8]
- Infrastructured[7]

1.3.1 Infrastructured Mode

With a wireless access point, the wireless LAN can operate in the infrastructure mode. This mode lets you connect wirelessly to wireless network devices within a fixed range or area of coverage. The access point has one or more antennas that allow you to interact with wireless nodes. In infrastructure mode, the wireless access point converts airwave data into wired Ethernet data, acting as a bridge between the wired LAN and wireless clients. Connecting multiple access points via a wired Ethernet backbone can further extend the wireless network coverage. As a mobile computing device moves out of the range of one access point, it moves into the range of another. As a result, wireless clients can freely roam from one access point domain to another and still maintain seamless network connection.

1.3.2 Ad Hoc Mode (Peer-to-Peer Workgroup)

In an ad hoc network, computers are brought together as needed; thus, the network has no structure or fixed points—each node can be set up to communicate with any other node. No access point is involved in this configuration. This mode enables you to quickly set up a small wireless workgroup and allows workgroup members to exchange data or share printers. The various variations of ad hoc mode are MANET, VANET and Wireless Sensor Networks.

(a) MANET

A mobile ad-hoc network (MANET)[3][31] is a self-configuring infrastructure less network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. An example of MANET is shown in the fig.1.2.

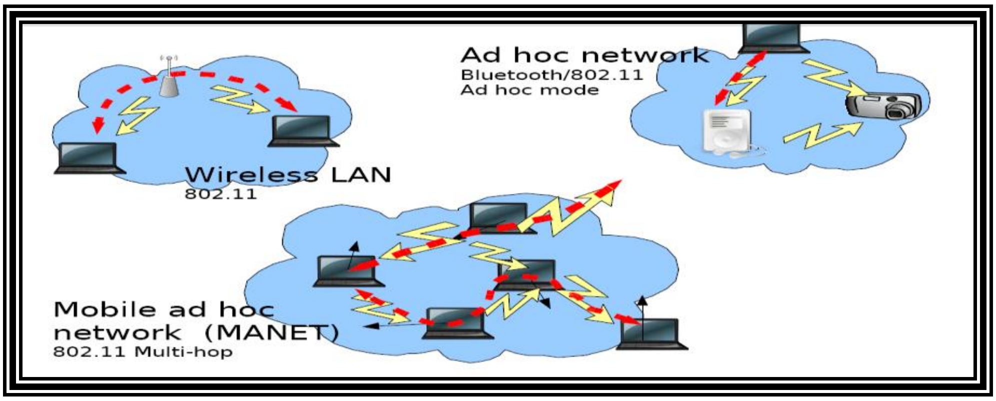


Fig.1.2: MANET[31]

The importance in mobile ad hoc networking is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Such networks are forecasted to have dynamic, sometimes rapidly changing, random, multihop topologies, which are likely composed of relatively bandwidth-constrained wireless links. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet.

MANETs are a kind of wireless ad-hoc networks that usually has a routeable networking environment on top of a Link Layer ad hoc network.

(b) VANET

A Vehicular Ad-Hoc Network, or VANET[40][47] is a technology that uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 metres of each other to connect and, in turn, create a network with a wide range. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is created, a view of such a network is shown in fig 1.3.

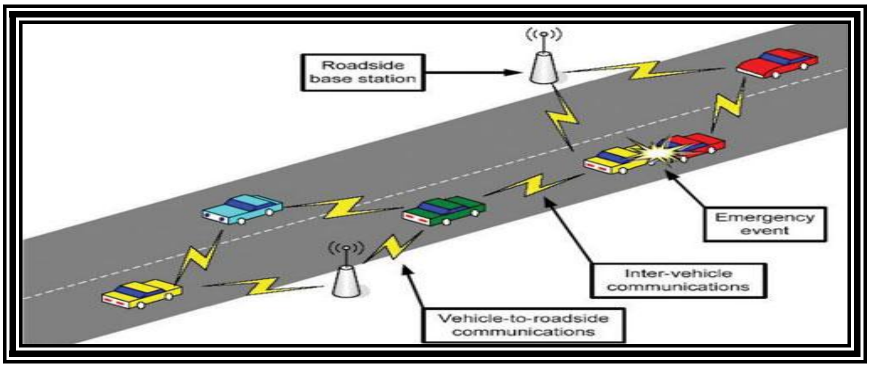


Fig.1.3: VANET[9]

It is estimated that the first systems that will integrate this technology are police and fire vehicles to communicate with each other for safety purposes. Further, A novel type of wireless access called Wireless Access for Vehicular Environment (WAVE) is dedicated to vehicle-to-vehicle and vehicle-to-roadside communications. While the major objective has clearly been to improve the overall safety of vehicular traffic, promising traffic management solutions.

(c) Wireless Sensor Networks

A wireless sensor network (WSN)[6][54] consists of spatially distributed autonomous sensors to *monitor* physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location. An example of wireless sensor networks is shown in fig.1.4.

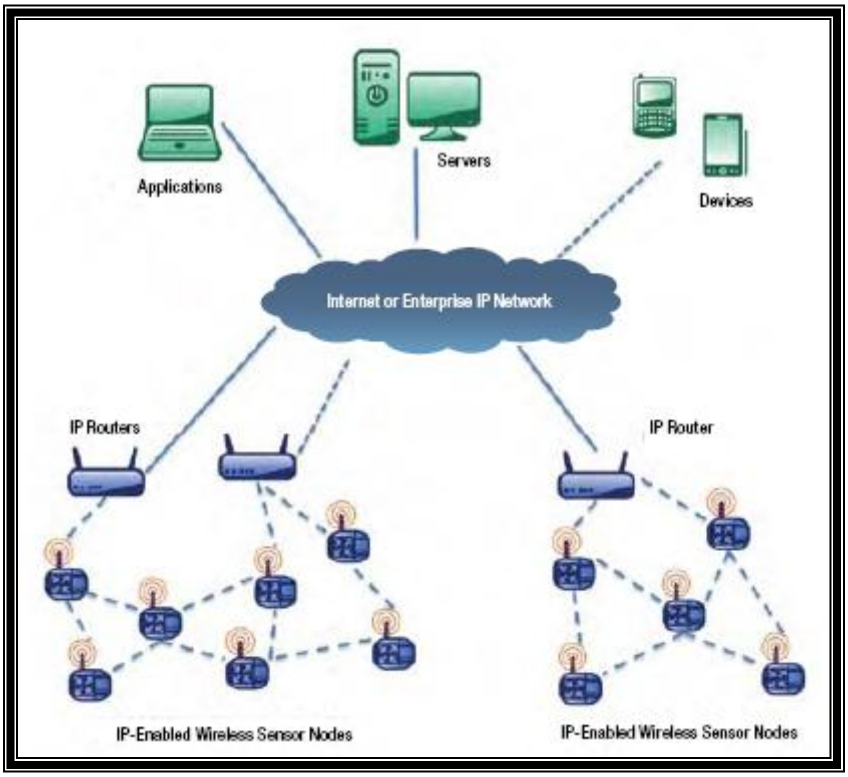


Fig.1.4: Wireless Sensor Networks[2]: A network of end to end wireless network nodes, IP routers and internet applications.

The wireless sensor network was primarily developed as a military application to survey a battlefield; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

CHAPTER 2

LITERATURE REVIEW

2.1 VANETs

The cost reduction and fast evolution experienced by wireless communication technologies have made them suitable for a wide spectrum of applications. One of them is Vehicular Ad hoc Networks. These wireless devices will have a high penetration rate among vehicles when being able to offer safety-related inter-vehicle communications and convenience and personalized applications. Vehicular Ad hoc Networks[40][47] are not just for fun, their aim is even to avoid accidents (e.g. using periodic broadcast of messages containing vehicles status information such as position and speed vector and a safety system aware of its surrounding to detect potential dangerous situations for the driver). While safety applications could avoid injuries, convenience and leisure applications could increase the comfort of the driver and passengers: recommending routes depending on the traffic flow conditions, accessing Internet, gaming, sharing files or offering P2P services.

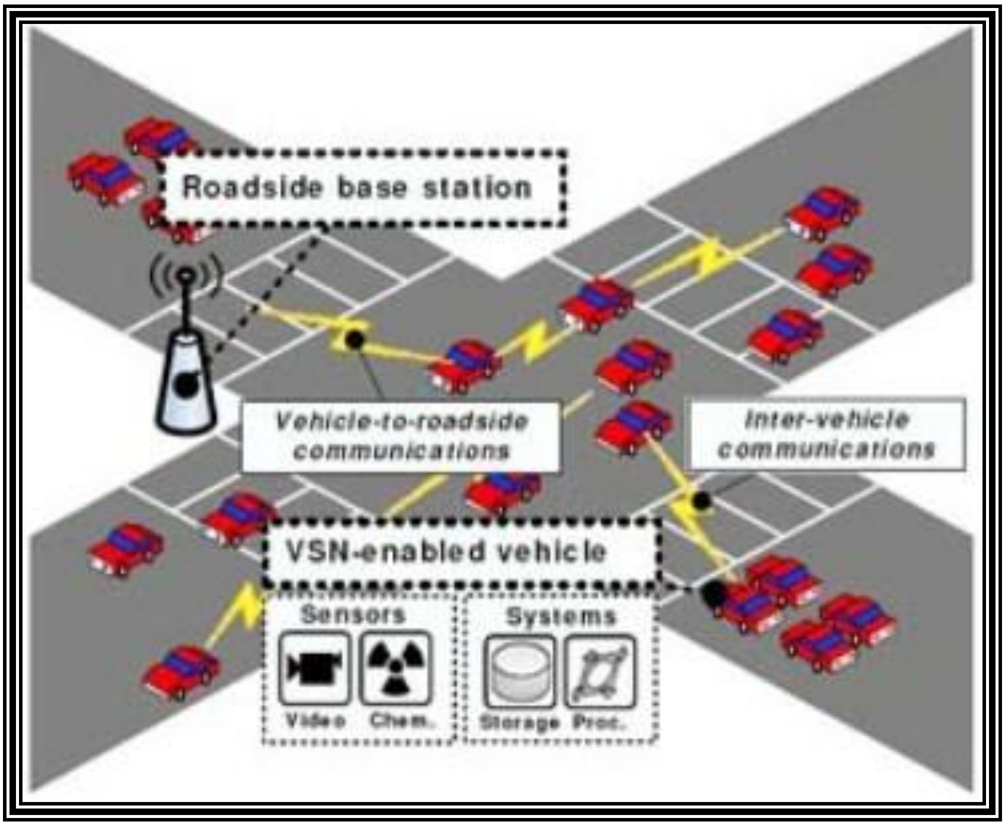


Fig. 2.1: A VANET Example[30]

VANETs are characterized by[51]:

- High velocity of the vehicles
- Environment factors: obstacles, tunnels, traffic jams, etc.
- Determined mobility patterns that depend on source to destination path and on traffic conditions
- Intermittent communications (isolated networks of cars due to the fragmentation of the network)
- High congestion channels (e.g. due to high density of nodes)

2.2 Types of Applications in VANET

Transportation-related applications[15] are those applications that increase the safety of the driver and passengers. Transportation-related applications range from safety applications such as cooperative forward collision warning or extended electronic brake lights to traffic management applications such as road-condition warnings or alternative route warnings. Convenience and personalized applications increase the comfort of the driver and passengers.

Transportation-related applications and Convenience and personalized applications have a set of requirements that is common to almost all applications. The most interesting requirements are: coverage should be in the range of 10 to 1000 meters with a car maximum relative speed of 500 Km/h. Latency ranges between 50 ms to 500 ms.

In general, safety applications should not wait more than 200 ms and safety applications are characterized by being transactions that range data packet sizes from 100 Bytes to 2 KBytes reaching a few number of hops, most applications being for a single hop. A likely situation may arise in case there are traffic jams and redundant packets of multiple nodes consume the bandwidth.

2.2.1 Safety Applications

Typical safety applications[55] are characterized as being applications in which the main objective is to disseminate certain event that have occurred in the vicinity of the sender. Some examples described are:

- Cooperative awareness: to extend non-cooperative driver assistance systems whose perception is limited to the operative range of on-board sensors (adverse weather, obstacles or dangerous road conditions).
- Cooperative assistance: distribution of data (e.g. warning of accidents).
- Cooperative manoeuvring: exchange of relative position and dynamics between vehicles (e.g. LaneMerge/Lane Change Assistance, Cooperative Driving)

Two kind of safety messages may appear in safety applications:

- Periodic messages
- Event driven messages

(a) Periodic messages

Periodic messages are sent with the intention of detecting non-safe situations such as providing information (e.g. position, speed, direction, etc) about surrounding vehicles. Increasing the transmission power improves radio channel communication and higher area of reachability. However, high transmission power also means higher probabilities of collision (e.g. in high node density areas). In safety applications, there should be fairness between different applications and nodes.

(b) Event driven messages

Event-driven messages, are those messages generated on demand. Examples are an alert in a dangerous situation (e.g. front car braking) or a request in a routing protocol. In these type of messages, rapid diffusion of the information is vital (e.g. to avoid an accident).

2.2.2 Convenience And Personalized Applications

Typical Convenience and personalized applications[46] are:

- Car to Mobile devices: those applications between the car and mobile devices (e.g. mobile phone, MP3, laptop, etc).
- Car to Home and Car to Office: communications between the car and private networks either at home or at office.
- Car to Enterprise: communications between the car and companies (e.g. restaurants, gas stations, parking areas, etc) that give road services.
- Car to infrastructure: information received by a car from hot spots giving road and traffic information and car access to Internet.

- Car to Car: exchange of information between car users (e.g. files, platoon traveling, etc).

2.3 Intelligent Transportation System(ITS)

Intelligent Transport Systems (ITS)[21] is essentially the application of computer and communications technologies coming in aid of the transport problems. ITS technologies enable gathering of data or intelligence and then providing timely feedback to traffic managers and road-users. IT enables elements within the transportation system—vehicles, roads, traffic lights, message signs, etc.—to become intelligent by embedding them with microchips and sensors and empowering them to communicate with each other through wireless technologies. Intelligent transport systems vary in technologies applied, from basic management systems such as car navigation; traffic signal control systems; automatic number plate recognition or speed cameras to monitor applications, such as security CCTV systems; and to more advanced applications that integrate live data and feedback from a number of other sources, such as parking guidance and information systems; weather information; etc

2.4 Routing

Routing[24] is a process of determining a path between source and destination upon request of data transmission. Routing protocols for ad hoc networks need to account for several aspects. Since nodes can move around, and enter and leave the network, the network topology can change rapidly. Due to the possible rapid changes in topology, it can require a lot of communication for a node to keep a static picture of the topology. Since the nodes are vehicles or mobiles they operate on battery power, which limits the amount of data they can transmit, before recharging is necessary. Furthermore, the possible bandwidth for mobile devices today is not as high as for stationary networks. Most mobile devices today have less processing power and memory than standard PC's.

Due to these circumstances ad hoc routing protocols must minimize the number of packets used for maintaining the routes and must be able to adapt to changes in the topology. Routing table is the task of the routing algorithm along with the help of the routing protocol for their construction and maintenance.

2.5 Routing Protocols in VANETs

A routing protocol[57] 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. This section describes recent *unicast* routing protocols proposed in the literature where a single data packet is transported to the destination node without any duplication due to the overhead concern. Some of these routing protocols have been introduced in VANETs but have been used for comparison purposes or adapted to suit VANETs' unique characteristics.

The taxonomy[43] of these VANET routing protocols which can be classified as

- Topology-based routing[43]
- Geographic routing

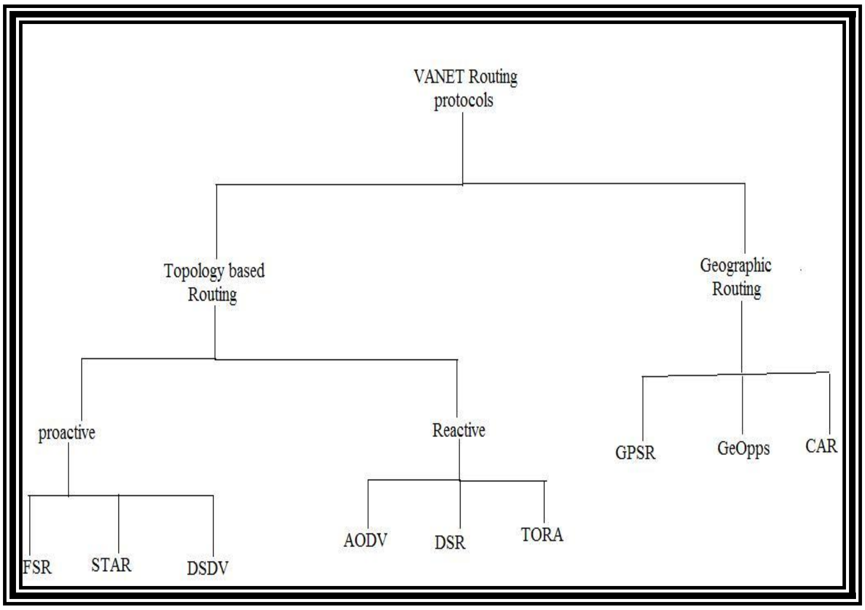


Fig. 2.2: Classification of VANET routing protocols[43]

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)[53]
- Reactive (on-demand) routing[4]

(a) Proactive (table-driven)

Proactive routing carries the distinct feature: 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. STAR(Source Tree Adaptive Routing), DSDV(Destination Sequenced Distance Vector), FSR(Fisheye State Routing) are examples of Proactive Routing.

(i) STAR

The protocol, *Source Tree Adaptive Routing (STAR)*[48], is a table-driven, link-state protocol that can be run with an emphasis on a small routing overhead measured as network load. A router in STAR communicates to its neighbors the parameters of its source routing tree, which consists of each link that the router needs to reach every known destination (and address range) in the ad hoc network or internet.

(ii) FSR

Fisheye State Routing[35] 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.

(iii) DSDV

Destination Sequenced Distance Vector[41] is a table driven Proactive protocol which uses the shortest path algorithm, Bellman- Ford Algorithm to calculate paths. Each node maintains a routing table with entries for all the nodes in the network, and not just the neighbours of the node. Nodes periodically send their routing tables to neighbours and they re-calculate shortest path upon the receipt of a routing table update.

(b) Reactive (On Demand)

Reactive routing[4] 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(Ad Hoc On-Demand Distance Vector Routing), DSR(Dynamic Source

Routing), TORA(Temporally Ordered Routing Algorithm) are the examples of Reactive Routing Protocols.

(i) AODV

In *Ad Hoc On Demand Distance Vector* (AODV)[11][25] routing, upon receipt of a broadcast query (RREQ), nodes record the address of the node sending the query in their routing table . This procedure of recording its previous hop is called *backward learning*. Upon arriving at the destination(fig.2.3a), a reply packet (RREP) is then sent through the complete path obtained from backward learning to the source(fig 2.3b). 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.

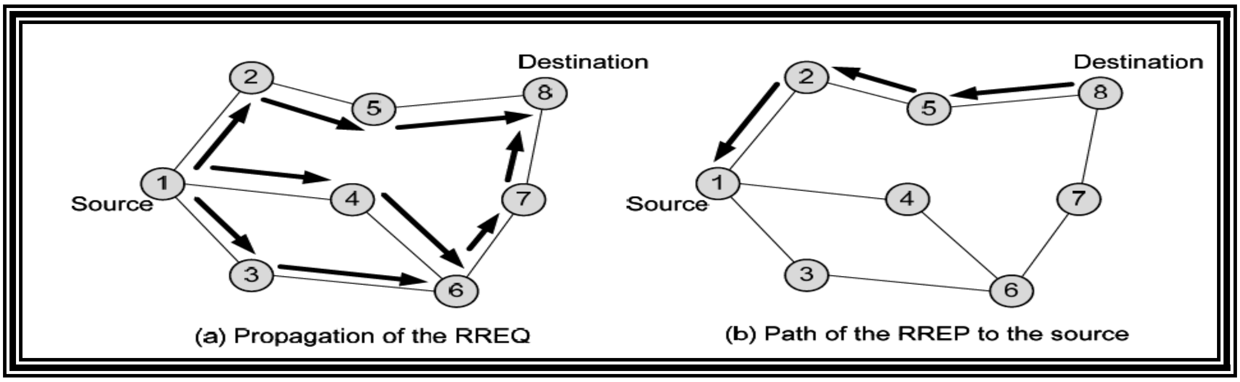


Fig.2.3: AODV Route discovery

(ii) DSR

Dynamic Source Routing (DSR)[25][50] 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 we allow the destination to send multiple route replies, the source node may receive and store multiple routes from the destination. 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. 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.

It forms the basis of the work and is also presented in section 2.6, later in this chapter.

(iii) TORA

Temporally Ordered Routing Algorithm (TORA)[38] 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 neighbour 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, maintenance of these routes can be overwhelmingly heavy, especially in highly dynamic VANETs.

(c) Comparison of Proactive and Reactive routing protocols

The following table 2.1 briefly compares[43] the Proactive(Table-Driven) routing protocol with Reactive(On-Demand) routing protocols.

Table 2.1: Comparison of Proactive and Reactive routing protocols

Proactive Protocols	Reactive protocols
Attempt to maintain consistent, up-to-date routing information from each node to every other node in the network.	A route is built only when required.
Constant propagation of routing information periodically even when topology change does not occur.	No periodic updates. Control information is not propagated unless there is a change in the topology
Incurs substantial traffic and power consumption, which is generally scarce in mobile computers	Does not incur substantial traffic and power consumption compared to Table Driven routing protocols.
First packet latency is less when compared with on-demand protocols	First-packet latency is more when compared with table-driven protocols because a route need to be built
A route to every other node in ad-hoc network is always available	Not available

2.5.2 Geographic (Position-based) Routing

In geographic (position-based) routing[23], 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 neighbours. The position of the destination is stored in the header of the packet by the source. The position of the node's one-hop neighbours is obtained by the beacons sent periodically with random jitter (to prevent collision). Nodes that are within a node's radio range will become neighbours 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 respectively. In other words, route is determined based on the geographic location of neighbouring nodes as the packet is forwarded. There is no need of link state exchange nor route setup.

(a) GPSR

In Greedy Perimeter Stateless Routing (GPSR)[26], a node forwards a packet to an immediate neighbour which is geographically closer to the destination node. This mode of forwarding is termed *greedy mode*. When a packet reaches a local maximum, a recovery mode is used to forward a packet to a node that is closer to the destination than the node where the packet encountered the local maximum. The packet resumes forwarding in greedy mode when it reaches a node whose distance to the destination is closer than the node at the local maximum to the destination.

(b) CAR

Connectivity-Aware Routing (CAR)[34] uses AODV-based path discovery to find routes. However, nodes that form the route record neither their previous node from backward learning nor their previous node that forwards the path reply packet from the destination. Rather, *anchor points*, which are nodes near a crossing or road curve, are recorded in the path discovery packet. A node determines itself as an anchor point if its velocity vector is *not* parallel to the velocity vector of the previous node in the packet. The destination might receive multiple path discovery packets; it chooses the path that provides better connectivity and lower delays. AGF is then used to forward the route reply back to the source via the recorded anchor points. When the source receives the route reply, it records the path to the destination and starts transmitting. Data packets are forwarded in a greedy manner toward the destination through the set of anchor points using AGF (Advanced Greedy Forwarding). In addition to handle mobility by AGF, CAR introduces "guards" to help to tack the current position of a destination. A guarding node can filter or redirect packets or adds information to a packet that will eventually deliver this information to the packet's destination.

(c) GEopps

The protocol, *Geographic Opportunistic routing for Vehicular Networks[44]* employ a store-and-forward message switching in which the fragments of a message (or the whole message) are forwarded and stored from host to host along a path until the message reaches the destination. This data exchange occurs during opportunistic contacts of the hosts. The successful delivery of the message is dependent on the mobility patterns of the hosts and the selection of the next message.

2.6 Comparison of Routing Protocols

In order to select the most suitable protocol to extend, we summarize the characteristics of each routing protocol in table 2.2.

Table 2.2: Comparison of Routing Protocols[53]

	DSDV	AODV	TORA	DSR
Update routing table periodically	Yes	Sends hello	No	No
Support one-way link	No	No	No	Yes
The mechanism of routing	Next hop	Next hop	Next hop	Source routing
The metrics of routing	Shortest path	Shortest path	Shortest path	Shortest path
Need other protocol support	No	No	IMEP	No

2.7 DSR protocol-RFC 4728

The *Dynamic Source Routing* protocol (DSR)[12] is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR is a Transport layer Protocol. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. Network nodes (computers) cooperate to forward packets for each other to allow communication over multiple "hops" between nodes not directly within wireless transmission range of one another. As nodes in the network move about or join or leave the network, and as wireless transmission conditions such as sources of interference change, all routing is automatically determined and maintained by the DSR routing protocol. Since the number or sequence of intermediate hops needed to reach any destination may change at any time, the resulting network topology may be quite rich and rapidly changing.

2.7.1 Assumptions

Some assumptions[10] concerning the behavior of the nodes that participate in the ad hoc network are made. The most important assumptions are the following:

A1. All nodes that participate in the network are willing to participate fully in the protocols of the network.

A2. The diameter of the network are often small, e.g. in the interval of [5:10] nodes.

A3. Nodes can detect and discard corrupted packages.

A4. The speed at which nodes move is moderate with respect to packet transmission latency.

A5. Each node can be identified by a unique id by which it is recognized in the network.

2.7.2 DSR Header

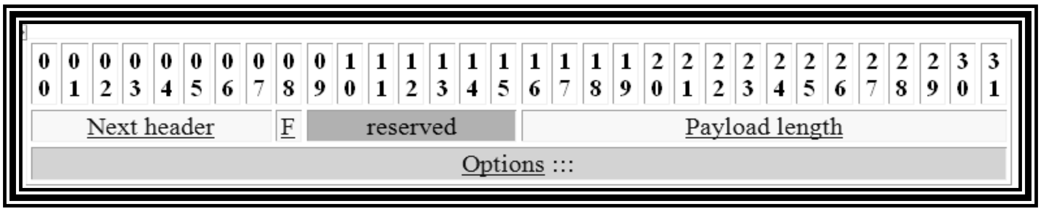


Fig. 2.4: DSR header[42]

Next Header. 8 bits.

Identifies the next protocol header following this header. Uses the same values as the IPv4 protocol field. If no header follows, then this field MUST be set to 59 (no next header).

F, Flow State Header. 1 bit.

If this field is set, the header is a DSR Flow State. Otherwise, the header is a DSR Options header.

Reserved. 7 bits.
Must be cleared to 0.

Payload length. 16 bits.
The size of the Options section in bytes.

Options. Variable length.

2.7.3 Mode of operation

The DSR protocol is composed of two mechanisms that work together to allow the discovery[10] and maintenance[10] of source routes in the ad hoc network:

(a) DSR route discovery

When some node **S** originates a new packet destined to some other node **D**, it places in the header of the packet a *source route* giving the sequence of hops that the packet should follow on its way to **D**. Normally, **S** will obtain a suitable source route by searching its *Route Cache* of routes previously learned, but if no route is found in its cache, it will initiate the Route Discovery protocol to dynamically find a new route to **D**. In this case, we call **S** the *initiator* and **D** the *target* of the Route Discovery.

For example, fig. 2.5 illustrates an example Route Discovery, in which a node **A** is attempting to discover a route to node **E**. To initiate the Route Discovery, **A** transmits a ROUTE REQUEST message as a single local broadcast packet, which is received by (approximately) all nodes currently within wireless transmission range of **A**. Each ROUTE REQUEST message identifies the initiator and target of the Route Discovery, and also contains a unique *request id*, determined by the initiator of the REQUEST. Each ROUTE REQUEST also contains a record listing the address of each intermediate node through which this particular copy of the ROUTE REQUEST message has been forwarded. This route record is initialized to an empty list by the initiator of the Route Discovery.

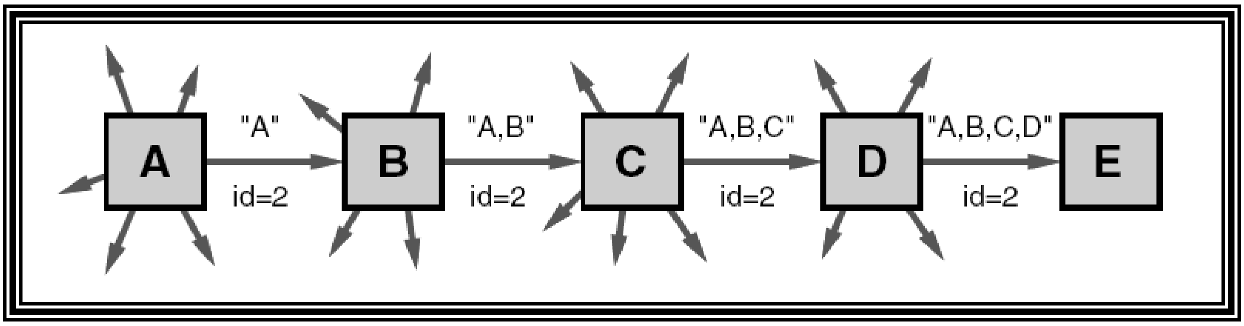


Fig. 2.5 : Route discovery example; node A is the initiator, and node E is the target

When another node receives a ROUTE REQUEST, if it is the target of the Route Discovery, it returns a ROUTE REPLY message to the initiator of the Route Discovery, giving a copy of the accumulated route record from the ROUTE REQUEST; when the initiator receives this ROUTE REPLY, it caches this route in its Route Cache for use in sending subsequent packets to this destination. Otherwise, if this node receiving the ROUTE REQUEST has recently seen another ROUTE REQUEST message from this initiator bearing this same request id, or if it finds that its own address is already listed in the route record in the ROUTE REQUEST message, it discards the REQUEST. Otherwise, this node appends its own address to the route record in the ROUTE REQUEST message and propagates it by transmitting it as a local broadcast packet (with the same request id).

In returning the ROUTE REPLY to the initiator of the Route Discovery, such as node **E** replying back to **A** , node **E** will typically examine its own Route Cache for a route back to **A**, and if found, will use it for the source route for delivery of the packet containing the ROUTE REPLY. Otherwise, **E** may perform its own Route Discovery for target node **A**, but to avoid possible infinite recursion of Route Discoveries, it must piggyback this ROUTE REPLY on its own ROUTE REQUEST message for **A**.

(b) DSR route maintenance

When originating or forwarding a packet using a source route,each node transmitting the packet is responsible for confirming that the packet has been received by the next hop along the source route; the packet is retransmitted (up to a maximum number of attempts) until this confirmation of receipt is received. For example, in the situation illustrated in fig 2.6, node **A** has originated a packet for **E** using a source route through intermediate nodes **B**, **C**, and **D**. In this case, node **A** is responsible for receipt of the packet at **B**, node **B** is responsible for receipt at **C**, node **C** is responsible for receipt at **D**, and node **D** is responsible for receipt finally at the destination **E**. This confirmation of receipt in many cases may be provided at no cost to DSR, either as an existing standard part of the MAC protocol in use, or by a *passive acknowledgement* (in which, for example, **B** confirms receipt at **C** by overhearing **C** transmit the packet to forward it on to **D**). If neither of these confirmation mechanisms are available, the node transmitting the packet may set a bit in the packet's header to request a DSR-specific software acknowledgement be returned by the next hop; this software acknowledgement will normally be transmitted directly to the sending node, but if the link between these two nodes is uni-directional, this software acknowledgement may travel over a different, multi-hop path.

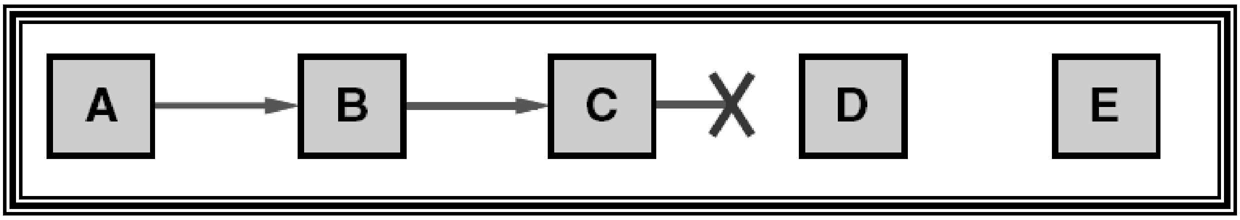


Fig 2.6: Route maintenance example: Node c is unable to forward packet from A to E over its link to next hop D

If the packet is retransmitted by some hop the maximum number of times and no receipt confirmation is received, this node returns a ROUTE ERROR message to the original sender of the packet, identifying the link over which the packet could not be forwarded. For example, in fig. 2.6, if **C** is unable to deliver the packet to the next hop **D**, then **C** returns a ROUTE ERROR to **A**, stating that the link from **C** to **D** is currently "broken." Node **A** then removes this broken link from its cache; any retransmission of the original packet is a function for upper layer protocols such as TCP. For sending such a retransmission or other packets to this same destination **E**, if **A** has in its Route Cache another route to **E** (for example, from additional ROUTE REPLYs from its earlier Route Discovery, or from having overheard sufficient routing information from other packets), it can send the packet using the new route immediately. Otherwise, it may perform a new Route Discovery for this target.

2.8 Mobility Models

To evaluate the performance of protocol in VANET, the protocol should be tested under realistic conditions such as – transmission range, data traffic, movement of vehicles (nodes) etc. There have been a wide variety of mobility models (MM)[36] proposed and it is expected the MM should attempt to mimic the movement of real vehicle nodes, the changes in speed and direction must occur in reasonable time slots.

The MM[22][33] can further be classified as :

• Entity Mobility Model (EMM), and
• Group Mobility Model (GMM)[17]

There are seven different categories of EMM[28][32], as defined below:

- **Random Walk Mobility Model[39]** is a simple mobility model based on random directions and speeds.

- **Random Waypoint Mobility Model[39]** includes pause times between changes in destination and speed.
- **Random Direction Mobility Model**[39] is the model that forces vehicle nodes to travel to the edge of the simulation area before changing direction and speed.
- **Boundless Simulation Area Mobility Model** converts a 2D regular simulation area into a torus-shaped simulation area.
- **Gauss-Markov Mobility** Model uses one tuning parameter to vary the degree of randomness in the mobility pattern.
- **Probabilistic Version of the Random Walk Mobility Model** utilizes a set of probabilities to determine the next position of a vehicle node.
- **City Section Mobility** Model is a simulation area that represents streets within a city.

The five different categories of GMM[56]17] are:

- **Exponential Correlated Random Mobility Model** uses a motion function to create movements.
- **Column Mobility Model**, the set of mobile nodes form a line and are uniformly moving forward in a particular direction.
- **Nomadic Community Mobility Model**, a set of mobile nodes moves together from one location to another.
- **Pursue Mobility Model** is a GMM where a set of mobile nodes follow a given target.
- **Reference Point Group Mobility Model**, the group movements are based upon path traveled by a logical center.

2.8.1 Random Waypoint Mobility Model

The Random Waypoint Mobility Model[16] is extensively used in simulation studies of VANET. In this mobility model a node selects its destination and its speed. The node keeps moving until it reaches its destination at that speed. A vehicle node begins the simulation by waiting a specified pause-time. After this time it selects a random destination in the area and a random speed distributed uniformly between 0 m/s and Vmax. After reaching its destination point, the node waits again pause for time seconds before choosing a new waypoint and speed.

The vehicle nodes are initially distributed over the simulation area. This distribution is not representative to the final distribution caused by node movements. To ensure a random initial configuration for each simulation, it is necessary to discard a certain simulation time and to start registering simulation results after that time.

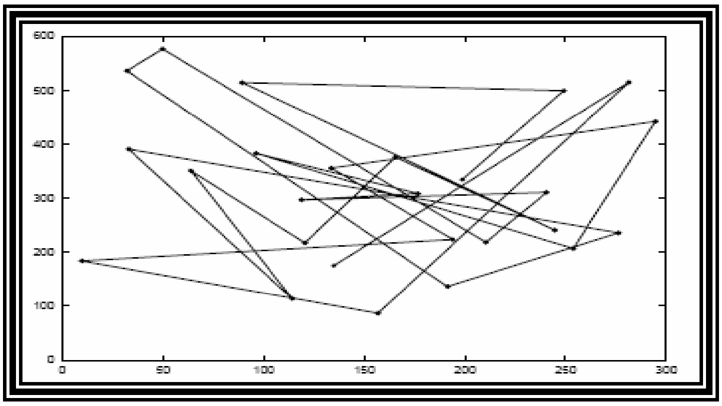


Fig 2.7: Traveling pattern of different mobile nodes in a given simulation area (rectangular shaped) based upon RWPMM.

Characteristics of Random Waypoint

The Random Waypoint model uses the concepts of *epoch* and *pause* making it a little bit more similar to realistic user mobility model. It is widely accepted mainly due to its simplicity of implementation and analysis.

2.8.2 Reference Point Group Mobility Model

In this model, each group has a logical center (group leader) that determines the groups motion behaviour[16]. Initially, each member of the group is uniformly distributed in the neighborhood of the group leader. Subsequently, at each instant, every node has a speed and direction that is derived by randomly deviating from that of the group leader.

Applications: Group mobility can be used in military battle field communications where the commander and soldiers form a logical group.

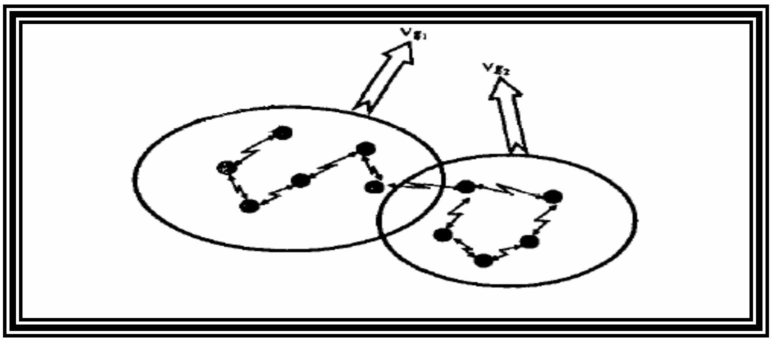


Fig.2.8: RPGM Model

2.8.3 Freeway Mobility Model

In this model we use maps. There are several freeways on the map and each freeway[54] has lanes in both directions.

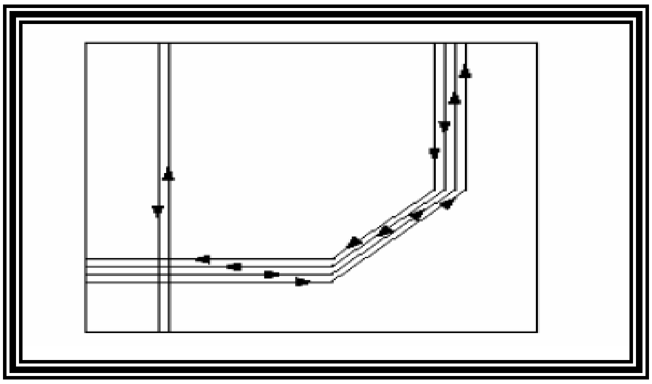


Fig. 2.9: Freeway Map

The differences between Random Waypoint and Freeway are the following:

- Each vehicle node is restricted to its lane on the freeway.
- The velocity of vehicle node is temporally dependent on its previous velocity.
- If two vehicle nodes on the same freeway lane are within the safety distance (SD), the velocity of the following node cannot exceed the velocity of preceding node.

Applications: It can be used in exchanging track status or tracking a vehicle on a freeway.

2.8.4 Manhattan Mobility Model

The Manhattan Model[20] is used to emulate the movement pattern of mobile nodes on streets defined by maps, also termed as City Section Mobility Model.

The map is composed of a number of horizontal and vertical streets. Each street has two lanes for each direction (north and south direction for vertical streets, east and west for horizontal streets). The vehicle node is allowed to move along the grid of horizontal and vertical streets on the map. At an intersection of a horizontal and a vertical street, the vehicle node can turn left, right or go straight. This choice is probabilistic: the probability of moving on the same street is 0.5, the probability of turning left is 0.25 and the probability of turning right is 0.25.

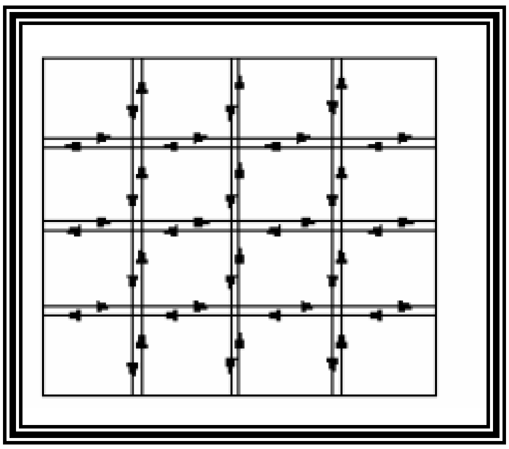


Fig. 2.10: Manhattan map

However, it differs from the Freeway model in giving a node some freedom to change its direction.

Applications: It can be useful in modeling movement in an urban area[21] where a pervasive computing service between portable devices is provided.

In this work, manually created map is being used which is similar to Manhattan Mobility Model in case of VANET. So, all the results have been taken by taking this manually created map for the realistic scenario of the traffic.

CHAPTER 3
PROBLEM STATEMENT

3.1 Problem Statement

Most current VANET routing protocols assume that every node in the network strictly follow the routing behaviour and is willing to forward packets for other nodes. Most of these protocols cope well with the dynamically changing topology. But, many misbehavior nodes are present in the network. A commonly observed misbehavior is packet dropping. Practically, in a VANET, due to the limitation of resources, some devices would not like to forward the packet for the benefit of others and they drop packets not destined to them which leads to degradation of performance and throughput.

Many routing protocols have been proposed by the researchers but in this work, performance evaluation of DSR protocol is being tested against various parameters such as sum of dropping packets, average throughput and packet size. Simulation of DSR protocol has been carried out on manually created map which is similar to Manhattan Mobility Model. The mobile nodes used for the traffic are varied to test the performance of the protocol in various cases and compare the results. The simulation has been carried out with the help of MOVE, SUMO and NS-2. MOVE provides the rapid generation of realistic mobility models for the use in simulation and SUMO shows the actual movement of vehicles on the mobility model used. NS-2 gives the actual simulation of the traffic by showing the movement of packets among the nodes.

3.2 Objectives and subtasks

The main objective of this work is to compare the performance results of DSR protocol when the number of nodes are varied against various parameters such as cumulative sum of number of dropped packets, throughput and packet size. The variation in the number of nodes is done as 4 nodes,10 nodes and 15 nodes. The results are evaluated in all cases and compared with each other.

The various scenarios used for the comparisons are:

- Cumulative sum of dropped packets vs. Event time
- Throughput of sending packets

- Throughput of receiving packets
- Throughput of dropping packets
- Packet size vs average throughput of sending packets
- Packet size vs average throughput of receiving packets
- Packet size vs average throughput of dropping packets

The result is evaluated for each of the above cases for 4 nodes , 10 nodes and 15 nodes.

CHAPTER 4
DESIGN AND IMPLEMENTATION

The work carried out in this thesis is by using open source Simulation tools on realistic scenario of traffic. In this section, the working principle of simulation tools used , scenarios used for analysis, simulation setup, metrics used, and performance comparison using tracegraph are described. The performance of the proposed protocol has been studied using simulation tools mainly Network Simulator (NS) and MOVE (*MObility model generator for VEhicular networks*) over SUMO.

4.1 MOVE

MOVE (MObility model generator for VEhicular networks)[37] is a Java-based application built on SUMO (Simulation of Urban Mobility) with a facility of GUI. In this work, a tool MOVE (MObility model generator for VEhicular networks) has been used to facilitate users to rapidly generate realistic mobility models for VANET simulations. MOVE is built on top of an open source micro-traffic simulator SUMO. The output of MOVE is a mobility trace file that contains information of realistic vehicle movements which can be immediately used by popular simulation tools such as NS-2. In addition, MOVE provides a set of Graphical User Interfaces that allows the user to quickly generate realistic simulation scenarios without the hassle of writing simulation scripts as well as learning about the internal details of the simulator.

Fig. 4.1 : MOVE[14]

In fig. 4.1, Mobility model generates the mobility model and traffic . It also configures and visualizes the vehicles over the map. Traffic model generates the tcl script needed by the ns2 for performing the simulation and also runs the NS-2 at backend as well as NAM.

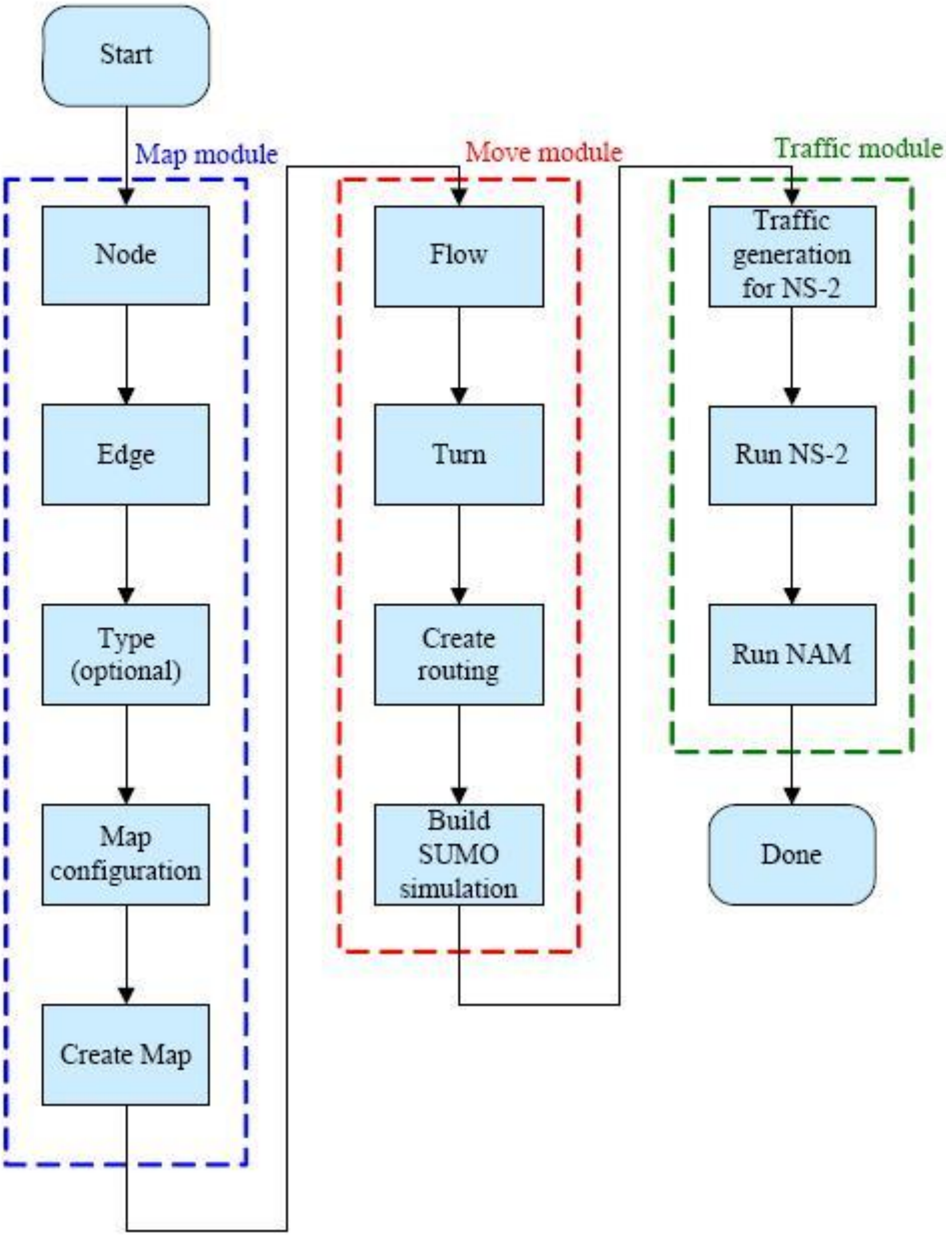


Fig. : 4.2: Roadmap of MOVE

Fig. 4.2 shows the roadmap of VANET MOVE**.** It shows step by step procedure of generating the map, then creating the vehicles, creating route for vehicles, visualize the real time scenario of traffic moving over lanes and then finally generating the tcl file which is needed by NS2 for running NS-2 and NAM.

4.1.1System Requirement

To use this software, it will need the following softwares installed on the system:

- Java SDK 1.4.2 or later

- SUMO 0.12.3
- NS-2 version 2.33

4.1.2 Mobiliy Generation

This part of the software (called MOVE - MObility model generator for VEhicular networks) will generate the mobility model created by SUMO.

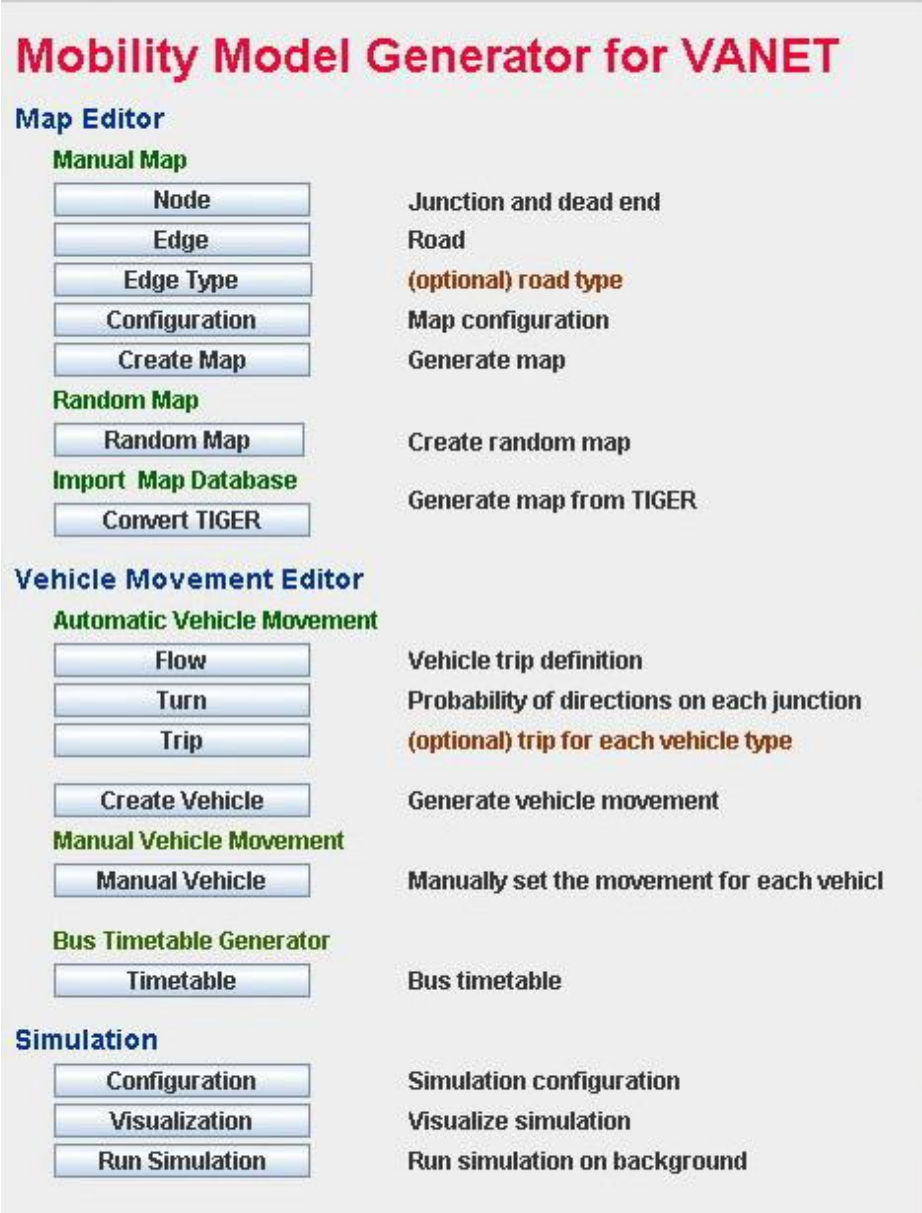


Fig. 4.3 : Mobility Model Generator

In fig. 4.3, it shows the general options for the creation of map, vehicle, vehicle movement and visualization.

By specifying the nodes and edges we can create the map or we can randomly create the map or it can also be generated from tiger. Then vehicles are created by specifying the number of vehicles, flow of vehicles and turn associated with the flow. When create vehicle is clicked, the routes of the vehicles on the map are configured. We can visualize the actual movement of vehicles on the map generated over SUMO. Thus , each and every button has its own functionality in simulation of traffic environment.

(a) Map generation

(i) Manually created map

Manually create your own map by specifying the nodes, edges and configuration. Save the file as <name>.net.xml.

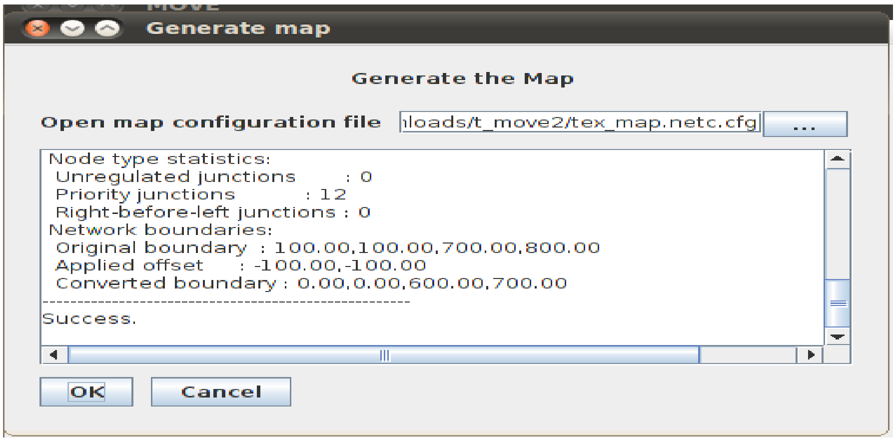


Fig. 4.4: Generating the map by configuring nodes and edges**.**

(ii)Automatically generated map

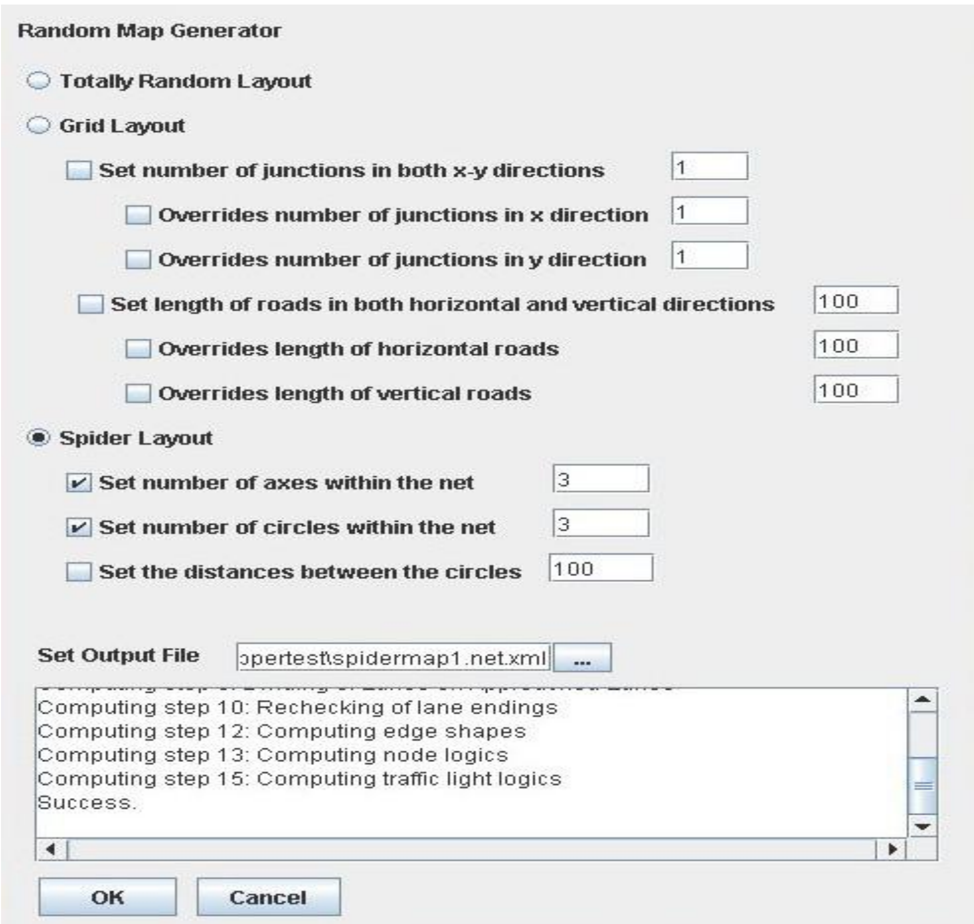


Fig.4.5: Creating Random Maps

Choose "Random Map" from fig. 4.3, MOVE main menu to do this. Fig. 4.5 shows that any of the three layouts of the map are possible whether it can be grid, spider or totally random. The output file is set by the user by giving any name, and the map will be generated by that name.

Three types of maps can be created using this;

- Grid
- Spider
- Totally random

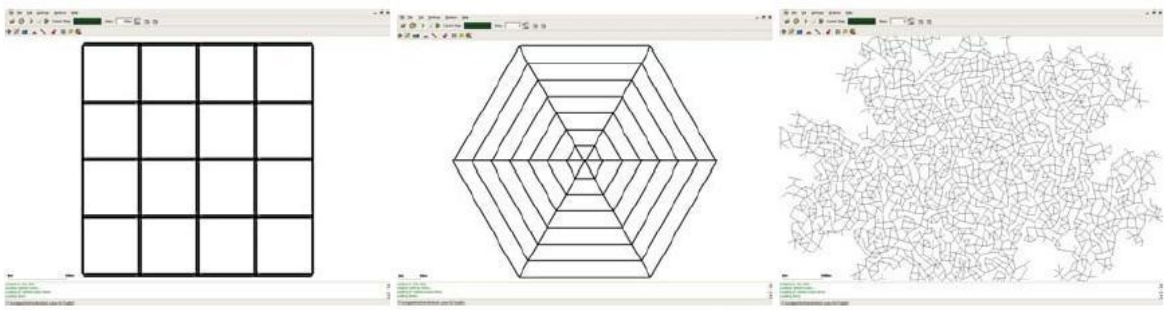


Fig. 4.6 : (a) Grid (b) Spider (c) Totally Random

(b) Vehicle Movements Generation

After the map is created, it is time to generate the movements.

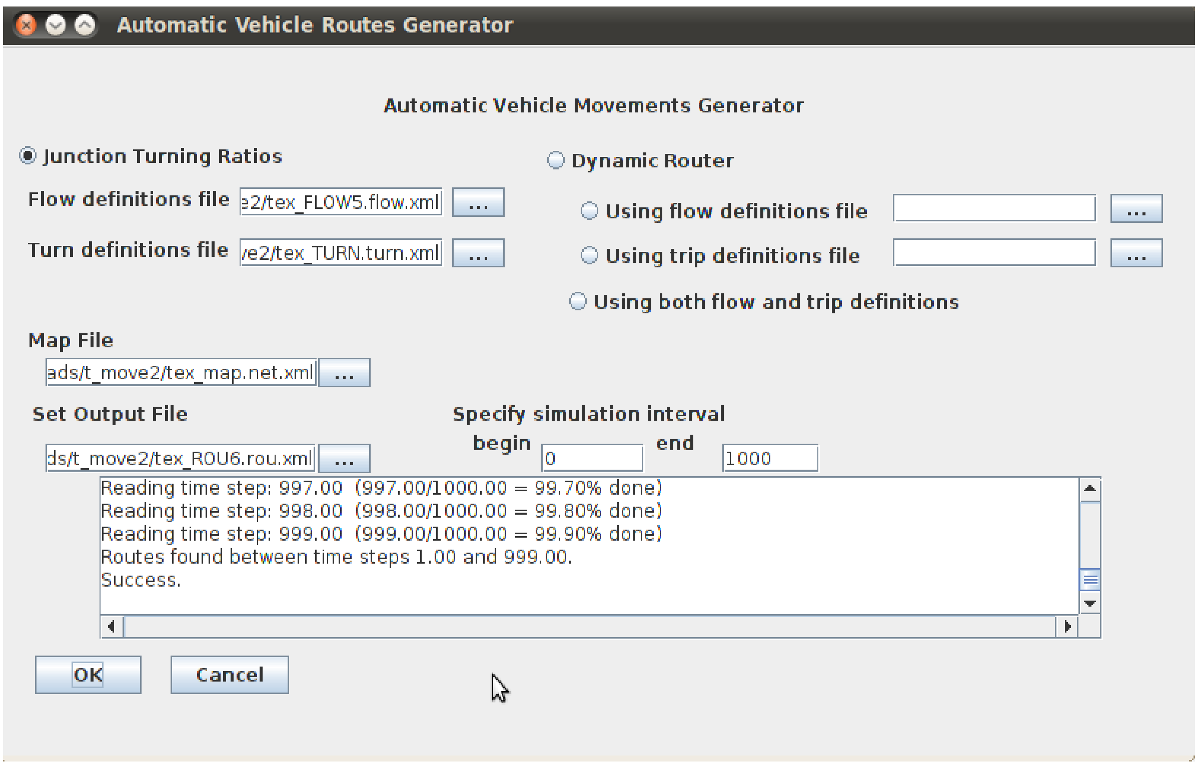
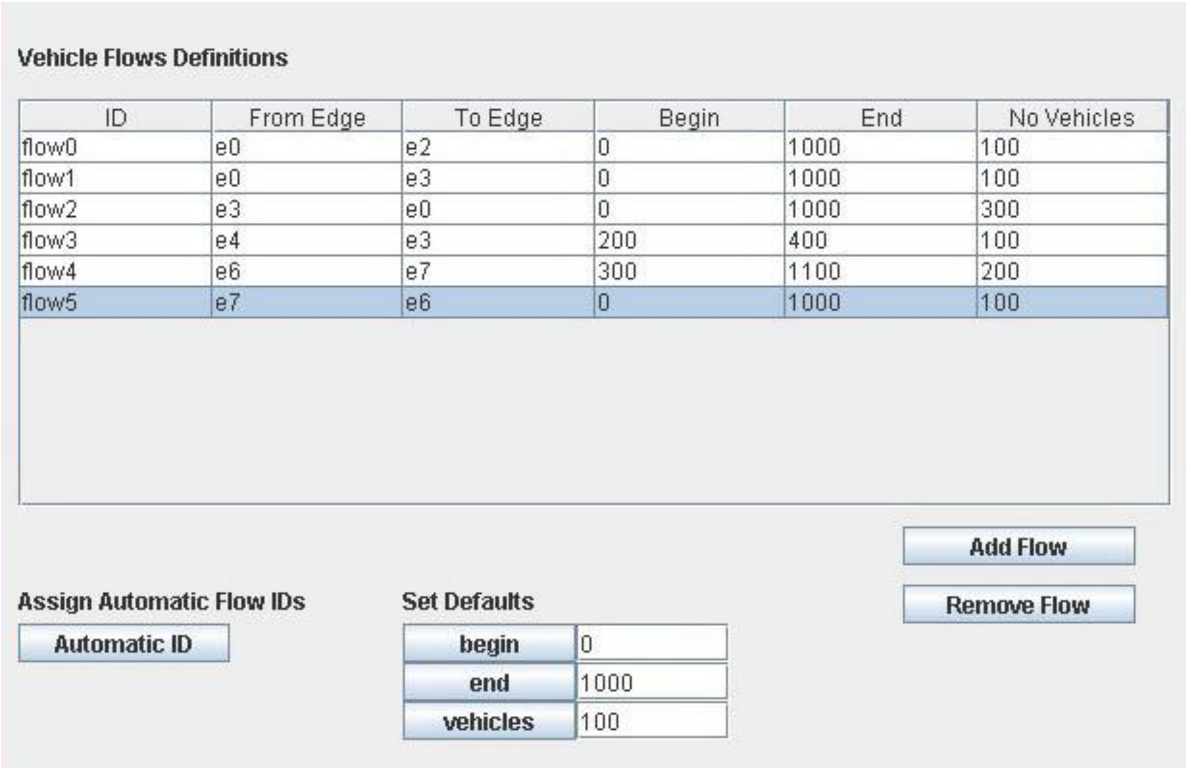


Fig. 4.7 : Creating Vehicles

To Create Flow and Turn of the Vehicles.

Simply select "Flow" from fig. 4.3, MOVE main menu. This editor will specify the groups of vehicle movements flow on the simulation. The IDs can be assigned automatically . Save the file as <name>.flow.xml



ID	From Edge	To Edge	Begin	End	No Vehicles
flow0	e0	e2	0	1000	100
flow1	e0	e3	0	1000	100
flow2	e3	e0	0	1000	300
flow3	e4	e3	200	400	100
flow4	e6	e7	300	1100	200
flow5	e7	e6	0	1000	100

Fig. 4.8 : Vehicle Flows Definitions

Using Turn Definitions

In fig. 4.7, when junction turning ratio is selected, turn file also need to be specified.
It defines the probability of directions on each junction. Save the file as <name>.turn.xml.
The turn definitions are specified in fig. 4.9.

Create Vehicle

When create vehicle is selected from fig. 4.3, MOVE main menu, there are two options in it in fig.4.7, either junction turning ratio can be selected in which both flow as well as turn definitions need to be specified. If dynamic router is specified, then only flow definition need to be defined. In junction turning ratio, the vehicles follow different paths along different directions wheres in dynamic router, all the vehicles follow the same path specified in flow definition.

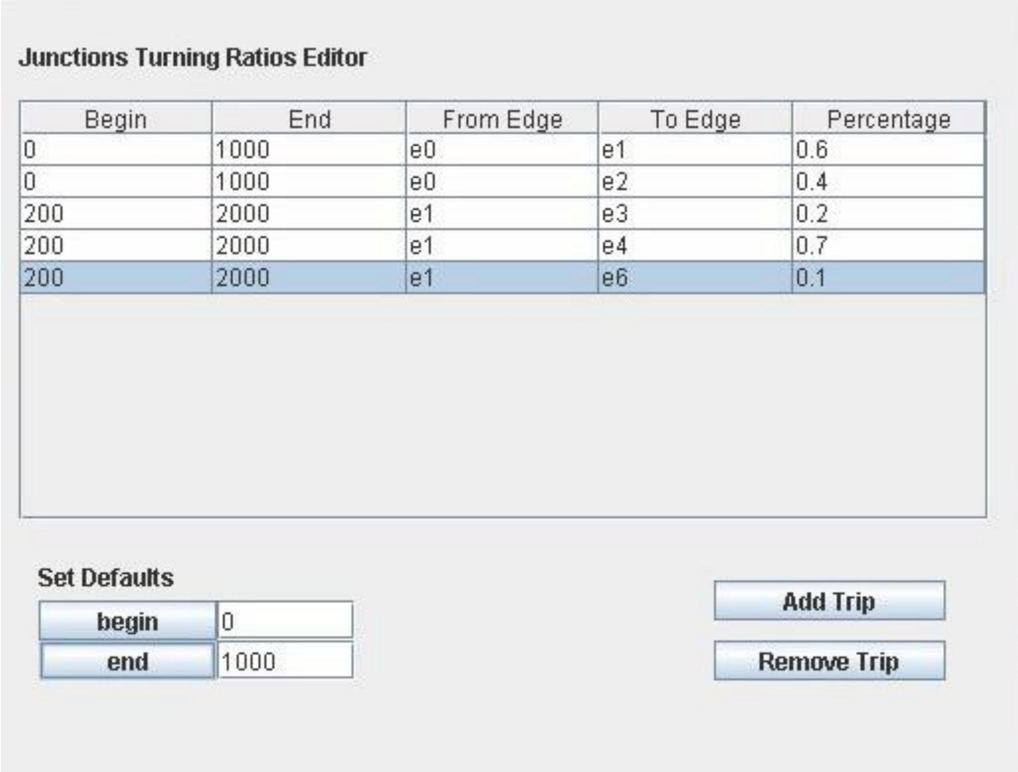


Fig. 4.9 : Junction Turning Ratios Editor

(c) Simulation Scenario

After the map and movement is complete, there is need to specify the configurations of the simulation. Select "Configuration" at the bottom on fig. 4.3, MOVE main menu.

In fig. 4.10, Specify the <name>.net.xml (map) and <name>.rou.xml (movements) locations and specify the beginning and end time of simulation. If we want to create the trace file, don't forget to check the checkbox and specify your trace output name. For example <name>.move.trace. Then save the file as <name>.sumo.cfg.

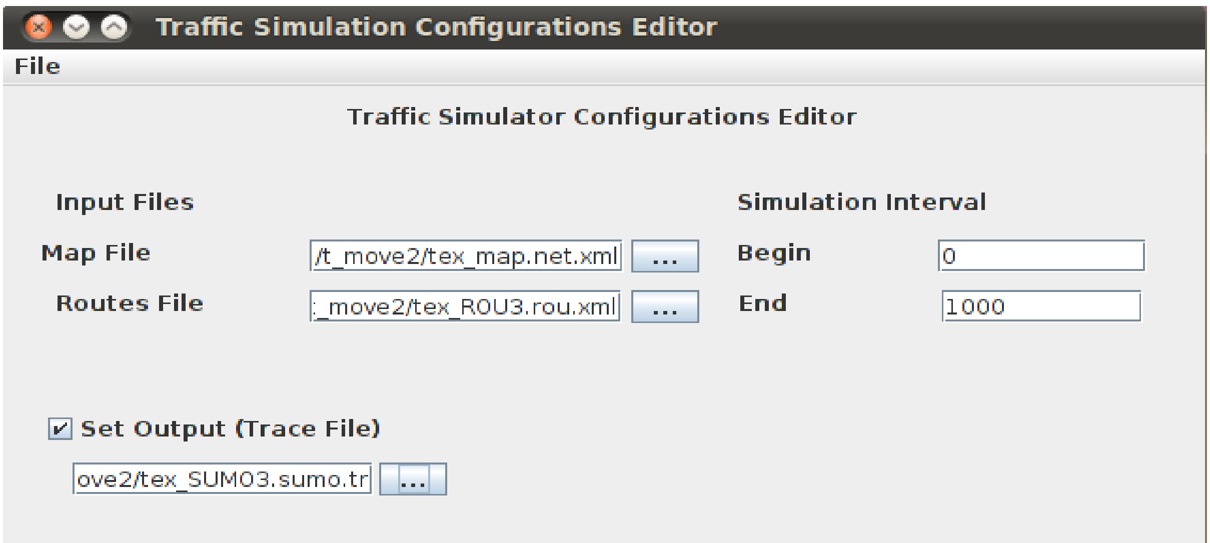


Fig. 4.10 Traffic Configuration

4.1.3 Traffic Model Generation

On MOVE main menu in fig. 4.1, select the "Traffic Model" and traffic model generator will be generated as shown in fig. 4.11

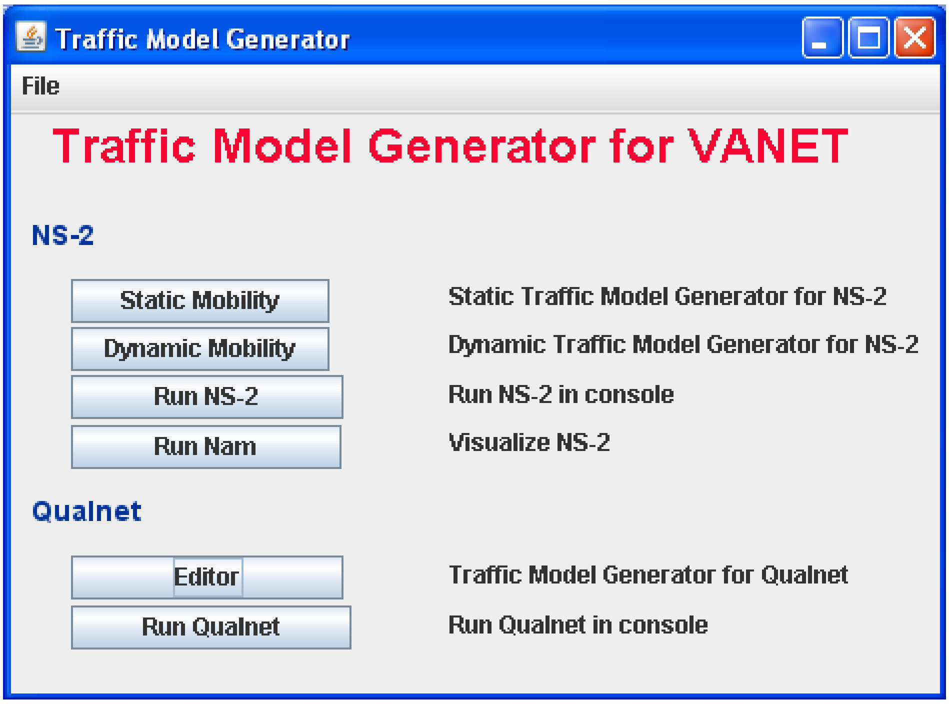


Fig.4.11: Traffic Model Generator

The traffic model generator consists of two main sections: for NS-2 and Qualnet. In this study, NS-2 is used . Right now, MOVE only supports static mobility.

This static mobility will generate the traffic simulation file (a tcl file) for NS-2 simulation tool. By clicking on the 'static mobility', the static mobility generator for NS-2 will be generated as shown in the following fig. 4.12.

To generate the NS-2 tcl script file, two files are needed: a MOVE trace file and its map file. First, select File->Import MOVE Trace then select <name>.move.trace file and <name>.net.xml file from the simulation directory. If the option to generate the trace file from the MOVE's simulation configuration is not checked previously, the trace file will not be created.

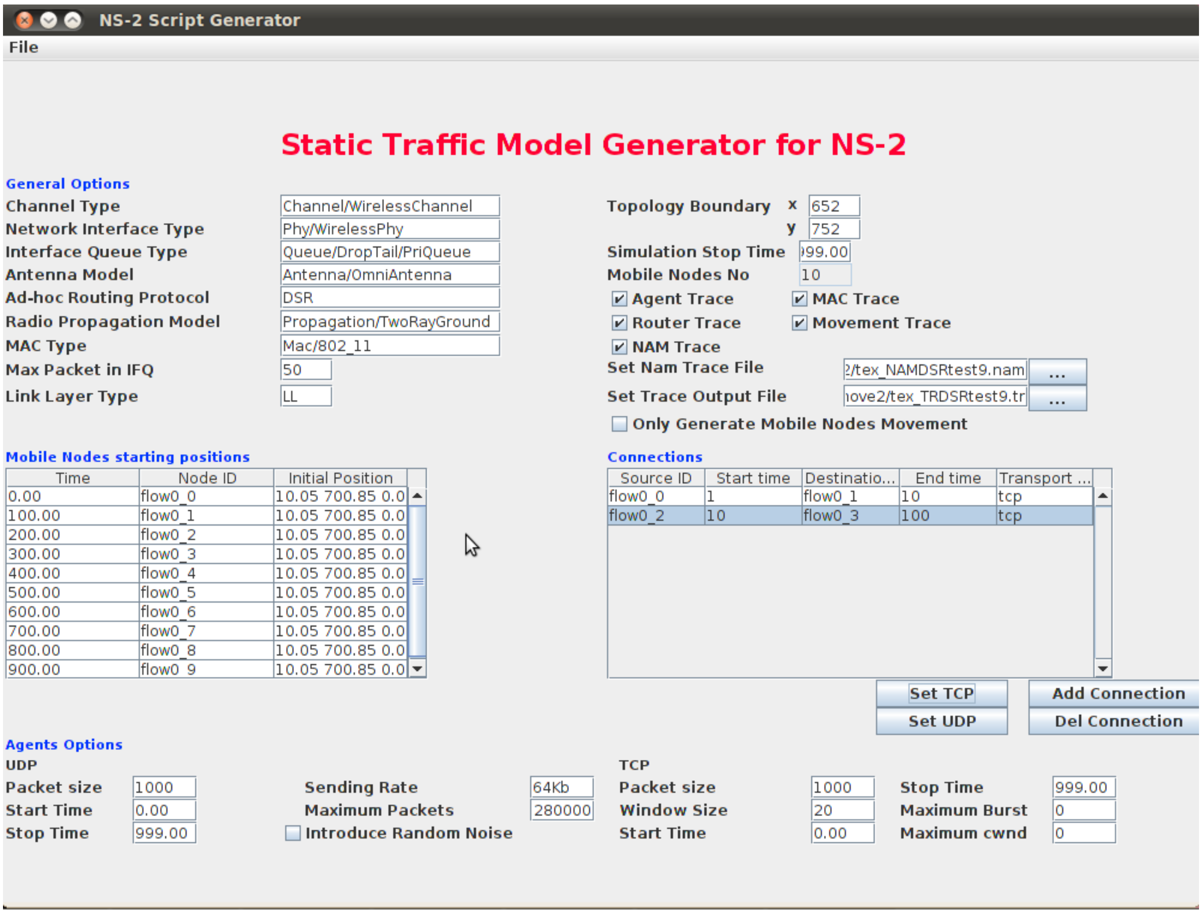


Fig. 4.12 : Static Mobility Model

After loading is done, specify the options of the TCL simulation. The trace output file is compulsory so there is need to specify it. If to generate the NAM trace file is choosen, the file location must be specifie. Then you can add the connections between each mobile nodes are added specified in the table on the left side and the transport protocol (tcp/udp) and application (http/ftp) are assigned.

After that, select File->Save or Save As and put it as <name>.tcl

4.2 SUMO

Simulation of Urban MObility" (SUMO) [49] is an open source, highly portable simulation package designed to handle large road networks. It was developed by employees of the Institute of Transportation Systems at the German Aerospace Center. As the traffic simulation “sumo” requires the representation of road networks and traffic demand to simulate in an own format, both have to be imported or generated using different sources. Here map and traffic demand both are fulfilled with the help of MOVE.

SUMO is a purely microscopic traffic simulation. Each vehicle is given explicitly, defined at least by an identifier (name), the departure time, and the vehicle’s route through the network.

If wanted, each vehicle can be described more detailed. The departure and arrival properties, such as the lane to use, the velocity, or the position can be defined. These all properties are defined when the vehicle is created and its flow definitions are set.

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
- Open source

4.2.2 Included Applications

SUMO is not only the name of the simulation application, but also the name of the complete software package which includes several applications needed for preparing the simulation. The package includes:

Table 4.1 : Included Applications of SUMO

Application Name	Short Description
SUMO	The microscopic simulation with no visualization; command line application
GUISIM	The microscopic simulation with a graphical user interface
NETCONVERT	Network importer and generator; reads road networks from different formats and converts them into the SUMO-format
NETGEN	Generates abstract networks for the SUMO-simulation
DUAROUTER	Computes fastest routes through the network, importing different types of demand description. Performs the DUA
JTRROUTER	Computes routes using junction turning percentages
DFROUTER	Computes routes from induction loop measurements
OD2TRIPS	Decomposes O/D-matrices into single vehicle trips
POLYCONVERT	Imports points of interest and polygons from different formats and translates them into a description that may be visualized by GUISIM
AdditionalTools	There are some tasks for which writing a large application is not necessary. Several solutions for different problems may be covered by these tools.

4.2.3 SUMO Visualization

It defines the actual movement of vehicles taking place over the realistic scenario of the mobility model generated which is grid mobility model.

Mobility Model used:

The simulation of DSR protocol has been carried out for the manually created map which is similar to Manhattan Mobility Model.

Run SUMO:

To run the SUMO, we need to click at 'visualization' in fig.4.3. It will run the SUMO in background and display the actual movement of vehicles in the map generated. The following screenshot fig. 4.13 shows how the vehicles seem while moving on the network.

To visualize the simulation, in order to generate the trace file simulation is run by pressing the play button. The delay time can also be adjusted for the simulation speed.

The small points over the lanes in fig. 4.13 are actually the vehicles moving over the lanes.

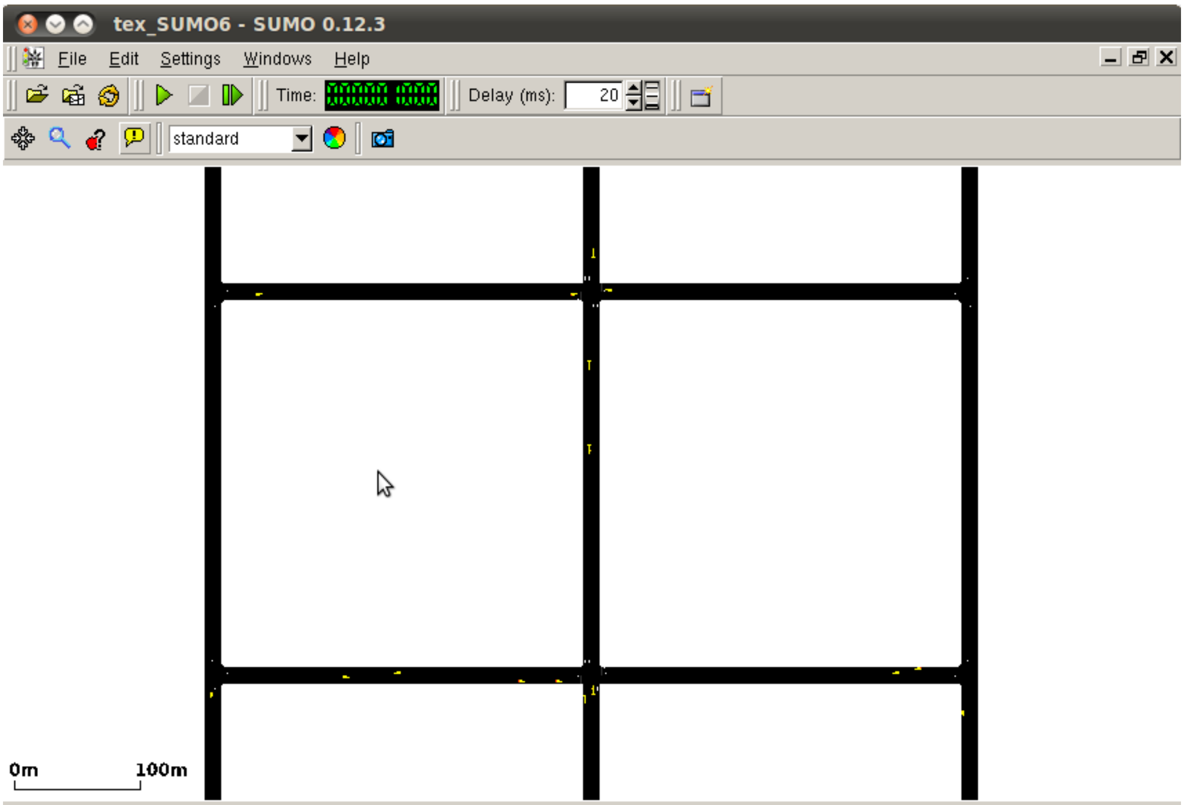


Fig. 4.13: Visualization of Traffic over SUMO

4.3 Network Simulation[1][45]

Simulation[1][45] can be defined as "Imitating or estimating how events might occur in a real situation". It can involve complex mathematical modelling, role-playing without the aid of technology, or combinations. The importance of simulation lies in the consideration of realistic conditions that change as a result of behaviour of others involved and thus we can anticipate the sequence of events or the final outcome. Different simulators such as ns2, GloMoSim, OPNET etc., are being used by researchers in order to evaluate the routing protocols. NS-2 is being used for the evaluation of the proposed routing protocol as the same is an open source, freely available and the programming languages used are C++, Tcl and OTcl.

4.3.1 NS2 Overview

The Network Simulator (NS)[27][29] is an event driven network simulator developed at UC Berkeley that simulates variety of IP networks. It implements network protocols such as Transmission Control Protocol and User Datagram Protocol, traffic source behaviour such as File Transfer Protocol, Telnet, Web, Constant Bit Rate and Variable Bit Rate, queue management mechanism, routing algorithms and more. Ns2 also implements multicasting and some of the MAC layer protocols for LAN simulations. The NS project is now a part of the VINT project that develops tools for simulation results display, analysis and converters that convert network topologies generated by well-known generators to NS formats. Currently, NS (version2) written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT) is available.

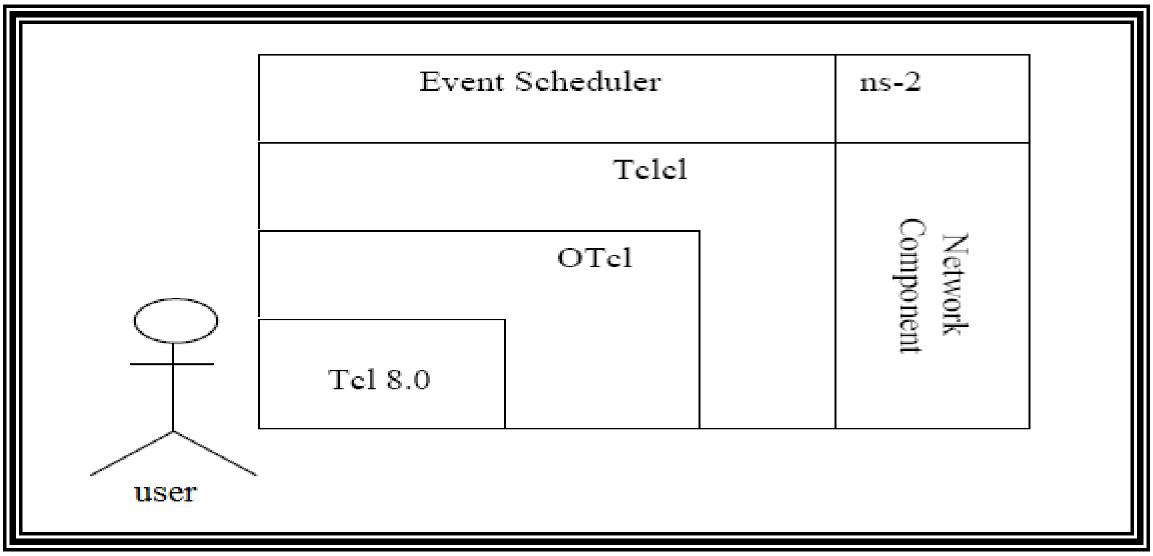


Fig. 4.14: Architectural view of NS

Fig. 4.14 shows the general architecture of NS. In this figure, a general user (not an NS developer) can be thought of standing at the left bottom corner, designing and running simulations in Tcl using the simulator objects in the OTcl library. The event schedulers and most of the network components are implemented in C++ and available to OTcl through the OTcl linkage that is implemented using tclcl. The whole thing together makes the NS, which is a OO extended Tcl interpreter with network simulator libraries.

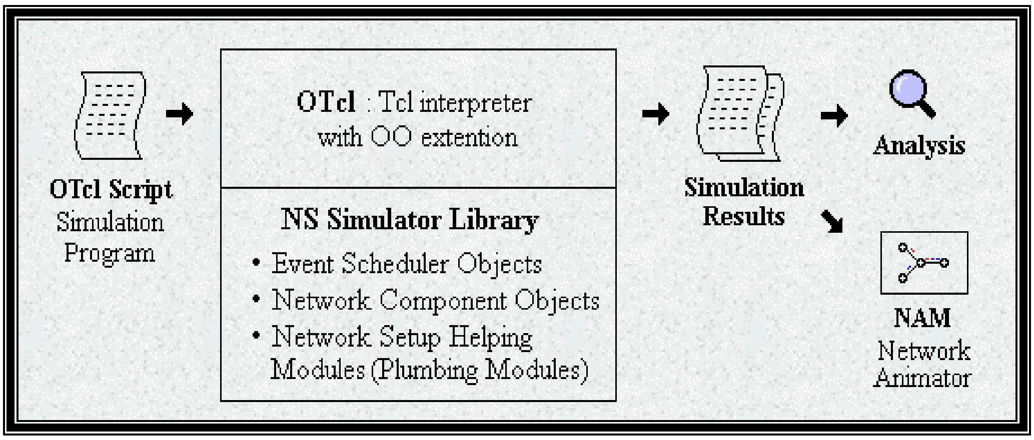


Fig 4.15: Simplified User's View of NS

Fig. 4.15, presents a simplified user's view, NS is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup module libraries.

4.3.2 Run Simulation in NS-2

To setup a simulation network, an OTcl script is written and to simulate it the script is executed which initiates an event scheduler and the network topology is setup using the network objects, controlling the traffic sources and the time to start and stop the transmitting of packets.

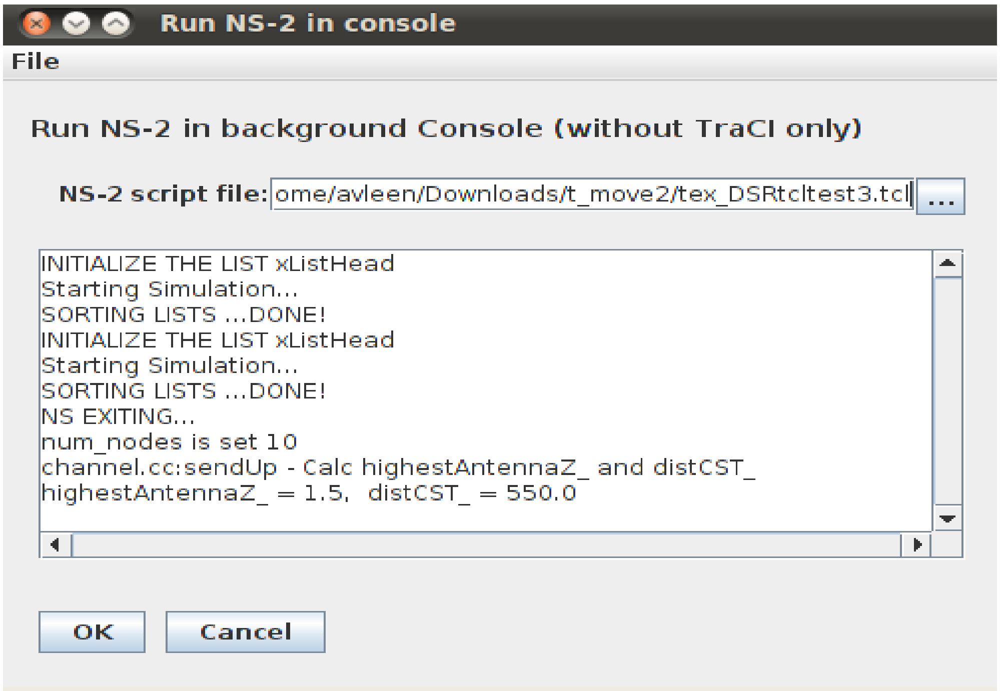


Figure 4.16 shows the tcl script being successfully executed over the simulator ns2.
The file ex_NS2.tcl is run on ns2 showing the results that the simulation has been done.

The DSRTEST6.tcl is being executed over NS-2 by virtue of which two files are being generated namely tex_NAMDSRtest6.nam and tex_TRDSRtest6.tr ..

The tex_NAMDSRtest6.nam file is used by NAM to visually show the actual movement of vehicle nodes and how the packet movement is being taking place between the nodes.

Tex_TRDSRtest6.tr is used by the tracegraph to analyse the results with different parameters. Tracegraph is described in detail in next section.

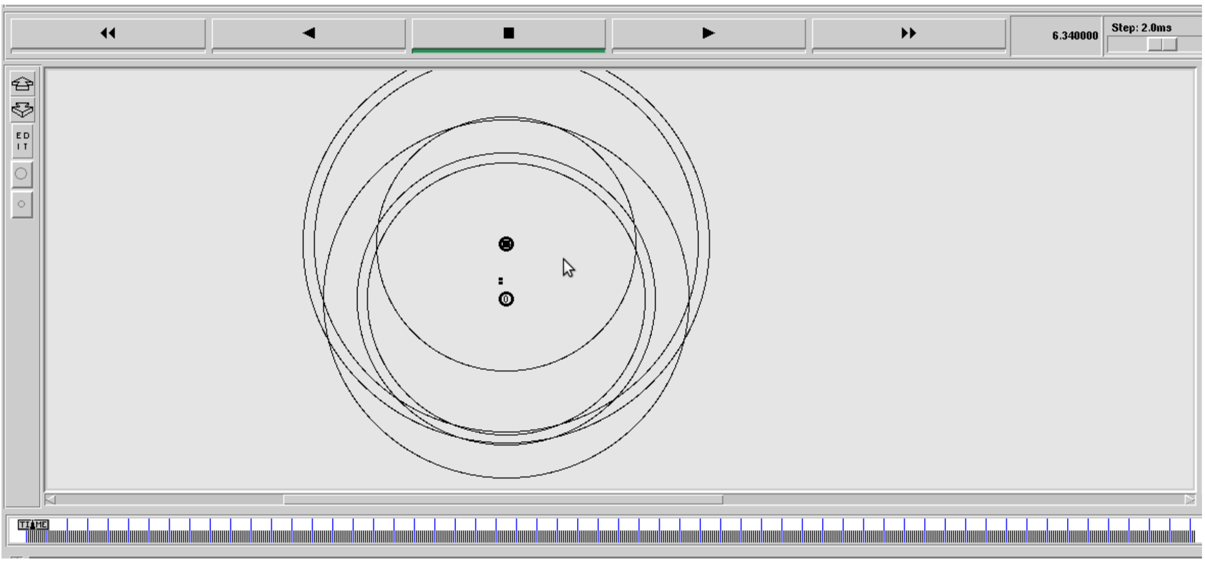


Fig. 4.17: NAM Generated for 10 nodes

Figure 4.17 shows the NAM at event time 6.34 done for 10 nodes. In this figure the packets sent between the nodes are clearly visible. The simulation time was set as 1000.

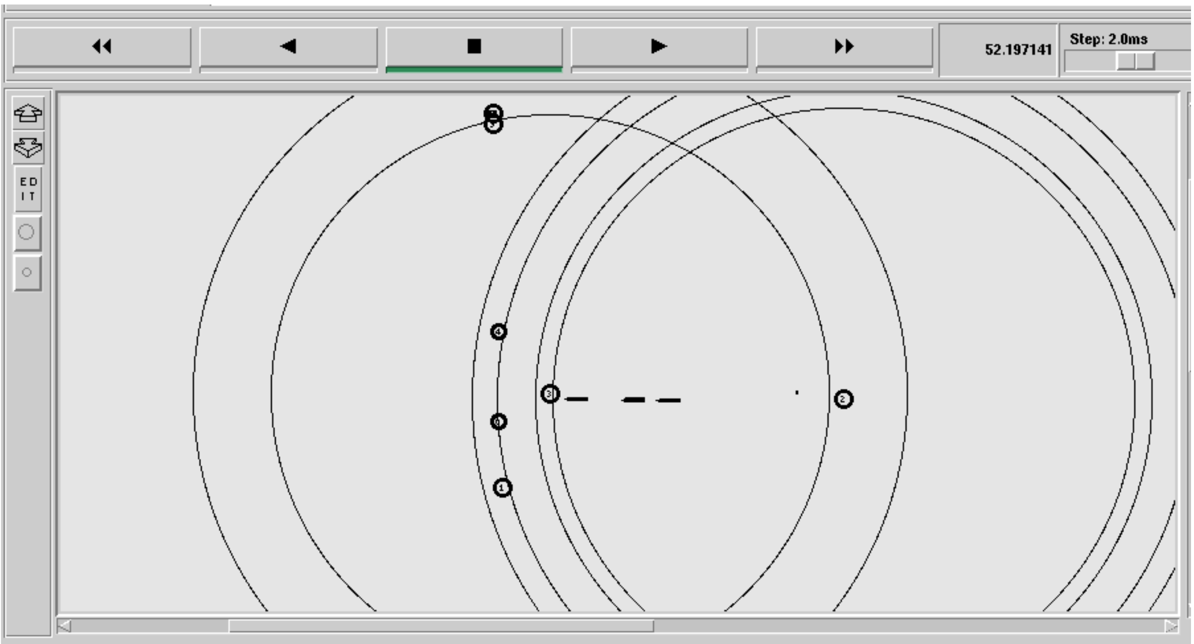


Fig. 4.18: NAM done for Simulation Time set as 100 secs

Fig. 4.18 shows the another snapshot for the NAM at event time of 52.19 for 10 nodes. The simulation time was set as 100 secs..

4.4 Tracegraph

Tracegraph202[52] 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. Trace graph runs under Windows, Linux, UNIX and MAC OS systems. Trace converter processes traces over 80x faster and is available to buy.

Supported ns-2 trace file formats:

- Wired
- Satellite
- Wireless (old and new trace)
- New trace
- Wired - Wireless

Some of the program features (version 2.05):

- delays, jitter, processing times, round trip times, throughput graphs and statistics
- whole network, link and node graphs and statistics
- all the results can be saved to text files, graphs can also be saved as jpeg and tiff
- x, y, z axes information: minimum, mean, maximum, standard deviation, median
- any graph saved in text file with 2 or 3 columns can be plotted
- script files processing to do the analysis automatically.

Tracegraph is installed in linux at */home/avleen/tracegraph202.* To run tracegraph every time, just navigate to */home/avleen/tracegraph202/* in Terminal and execute trgraph as follows: *$ ./trgraph*

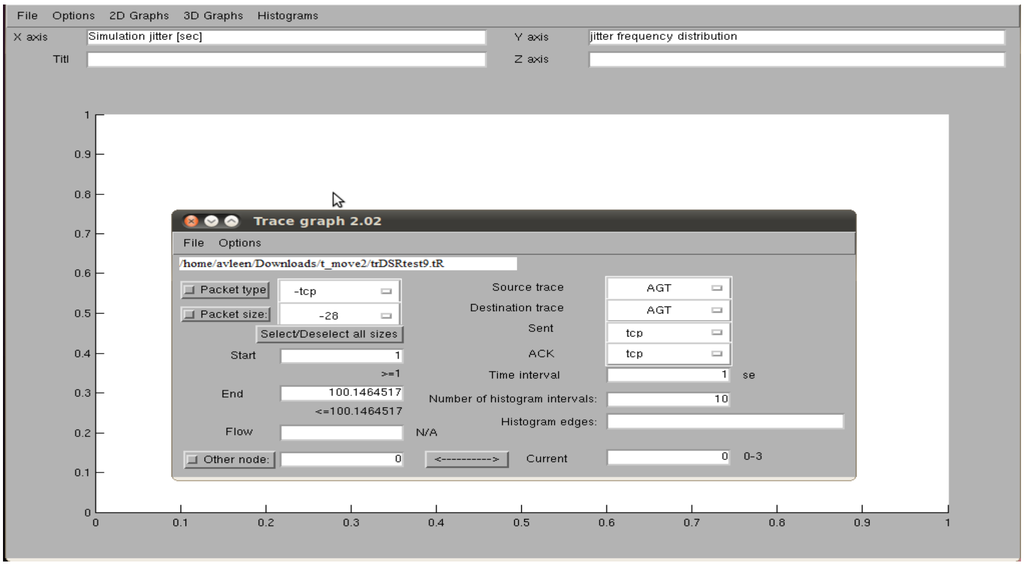


Fig. 4.19: Tracegraph Main Screen

Fig. 4.19 shows the main screen of tracegraph as the file tex_TRDSRtest9.tr is loaded by it.

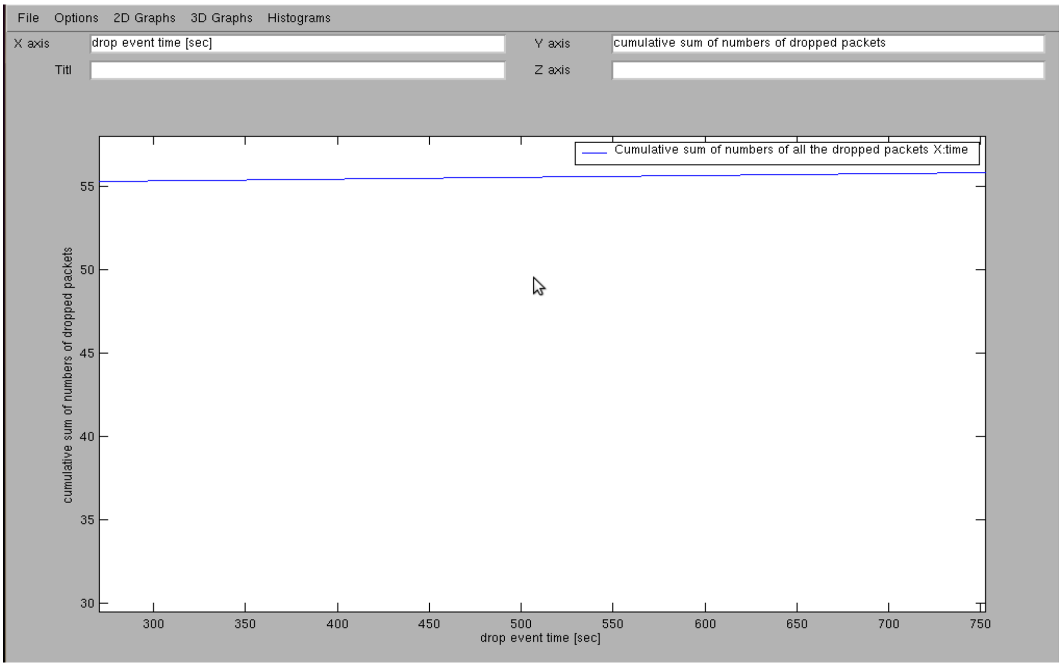


Fig. 4.20: Graph showing Cumulative Sum of Number of all Dropped Packets

Fig. 4.20 shows one sample graph of *cumulative sum of numbers of all dropped packets* made by tracegraph.

CHAPTER 5

RESULTS AND ANALYSIS

This chapter shows the results of the simulation. The analysis is being done on the basis of the results of *.tr file. With the help of 2D graphs, the simulation has been analyzed with different simulation time. When the TCL file is executed over NS-2, two files are generated, one is *.nam file which is used to visualize the mobile nodes movement and the flow of packets. The other file generated out of TCL file is *.tr which is given to tracegraph for generation of graphs over various parameters.

The .tcl files were generated for the following cases,

- 4 vehicle nodes
- 10 vehicle nodes
- 15 vehicle nodes

5.1 Network characteristics

The significance and relevance of network characteristics varies with applications. There are four basic characteristics which will be used in this work for the evaluation and comparison of results based on selecting different number of nodes.

The tracegraph is used to plot the graph by taking the following parameters of the simulation:

- Packet Size
- Throughput
- Cumulative sum of number of dropping packets

1. Packet Size

The **MTU(maximum transfer unit)** is the maximum size of a single data unit (e.g., a *frame*) of digital communications. MTU sizes are inherent properties of physical network interfaces, normally measured in bytes. The MTU for Ethernet, for instance, is 1500 bytes. Some types of networks (like Token Ring) have larger MTUs, and some types have smaller MTUs, but the values are fixed for each physical technology.

If the maximum TCP packet size is set too high, it will exceed the network's physical MTU and also degrade performance by requiring that each packet be subdivided into smaller ones (a process known as *fragmentation*). The packet size is normally set as 1400 or it can go to 1450 at max.

2. Throughput

In communication networks, such as Ethernet or packet radio, throughput or network throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time interval length(TIL).

The system throughput or aggregate throughput is the sum of the data rates that are delivered to all terminals in a network.

People are often concerned about measuring the maximum data throughput in bits per second of a communications link or network access. A typical method of performing a measurement is to transfer a 'large' file from one system to another system and measure the time required to complete the transfer or copy of the file. The throughput is then calculated by dividing the file size by the time to get the throughput in megabits, kilobits, or bits per second.

Unfortunately, the results of such an exercise will result in the goodput which is typically less than the maximum theoretical data throughput, leading to people believing that their communications link is not operating correctly. In fact, there are many overheads accounted for in goodput in addition to transmission overheads, including latency, TCP Receive Window size and system limitations, which means the calculated goodput does not reflect the maximum achievable throughput.

3. Cumulative Sum of Dropped Packets

The number of packets dropped at a given instance of time in the simulation run determines the efficiency of the protocol. It is found that during initial stages, the number of dropped packets is more since the route formation for genuine requests is delayed. During later stages, the unavailability of network resources causes the data packets to be dropped.

5.2 Simulation Setup

The table below list the details of the simulation setup used in this simulation based analysis of DSR protocol.

Table 5.1 : Simulation Setup

NS version	NS-2.33
MOVE version	2.64
SUMO version	0.12.3
Tracegraph version	2.0.2
DSR	NS-2 Default
No. Of nodes	4,10,15
Traffic type	TCP
Channel Type	Wireless Channel
Network Interface Type	Wireless physical
Antenna model	Omnidirectional
Radio Propagation Model	Two Ray Ground
Adhoc Routing Protocol	DSR
MAC type	IEEE 802.11
Simulation Time	1000 seconds
Data Type	CBR(Constant Bit Rate)
Data packet size	1000 bytes
Window Size	20
Scenario	Urban
Road Traffic Direction	Multidirectional
Queue Length	50

5.3 Performance Evaluation and Analysis

The analysis of various parameters has been done by varying the number of nodes used in the simulation. The various cases has been selected for the analysis by taking the parameters such as dropped packets, throughput and the packet size.

CASE A : Cumulative sum of dropped packets vs event time

The graph is plotted for the no of dropped packets in network against the event time. The event time is the time at which some event occurs.

1. Number of nodes: 4

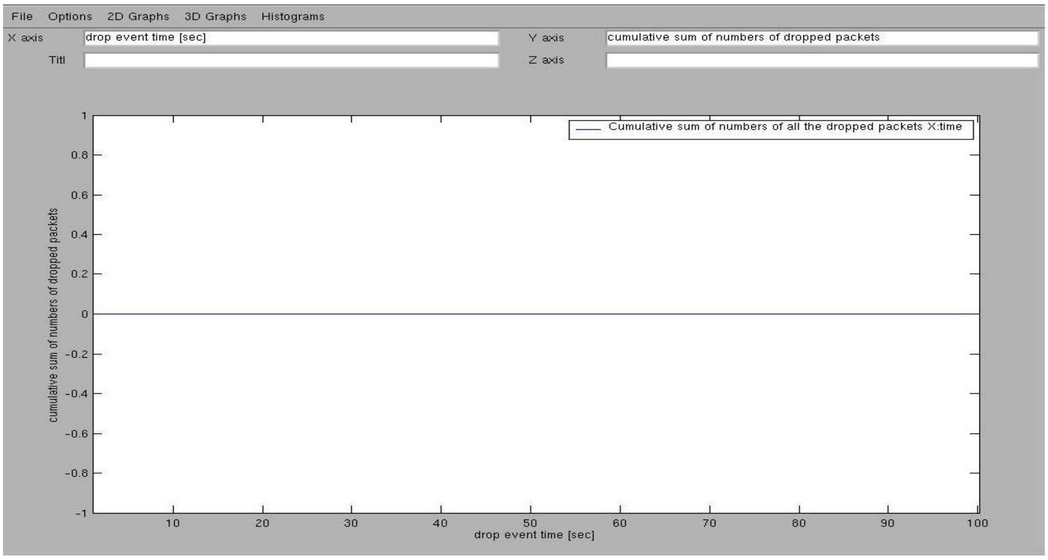


Fig. 5.1: Cumulative sum of dropped packets for 4 nodes

When the number of nodes is 4, no packet loss takes place against the event time.

2. Number of nodes: 10

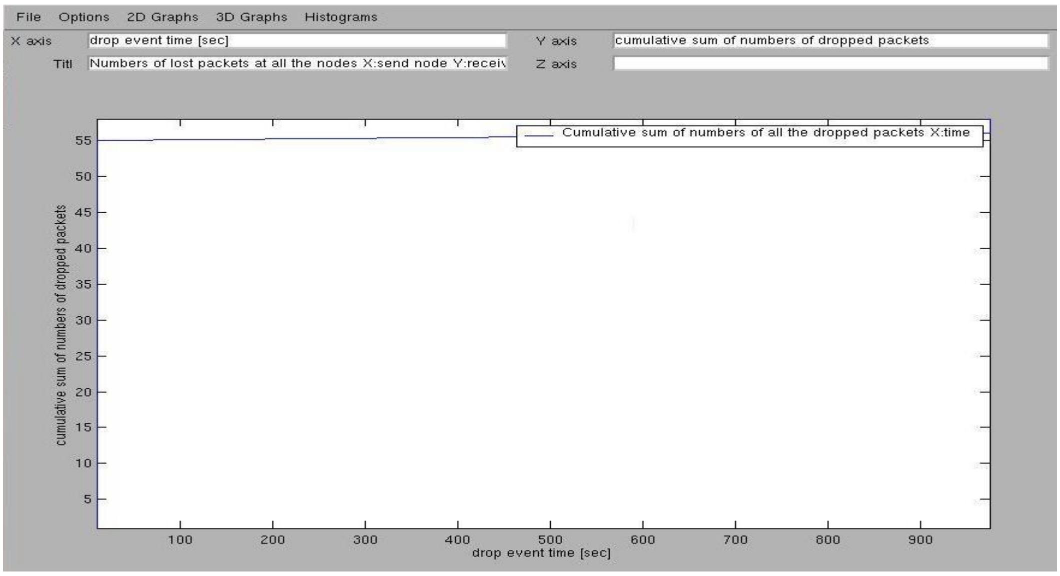


Fig. 5.2: Cumulative sum of dropped packets for 10 nodes

When the number of nodes are increased to 10, most of the packet loss takes place initially only, when the simulation starts. The number of dropped packets reaches to 55 packets in just 0 to 5 event time. The high rate of dropped packets at the initial stage is due to the number of generation of nodes at the initial step to receive the packets. The nodes are generated slowly as the time increases. So, initially there are no nodes to receive the packets leading to high rate of packet loss. After the dropped packet number reached to 55 packets, the additional dropped packets increases only by 2 in event time from 5 to 999 thus leading the total number to 57 packets.

3. Number of nodes : 15

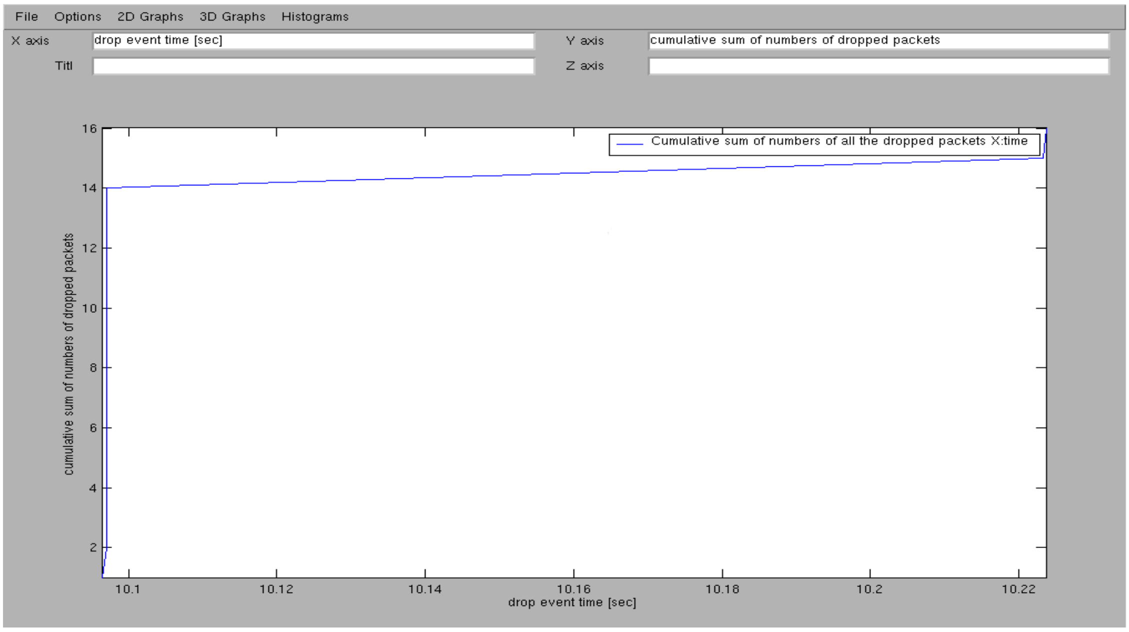


Fig 5.3: Cumulative sum of dropped packets for 15 nodes

When the number of nodes taken are 15, again the packet loss is more in the event time from 0 to 2 and then the dropped packets increases slowly with the event time. Here the interval of event time also decreases due to the increase in the number of dropped packets. The number of dropped packet between event time 10.1 and 10.22 is 1 packet.

INFERENCE: The number of dropped packets increases with the increase in number of nodes. The packet loss is always more at the initial stage and then increases slowly with the event time. The event time interval also decreases as the number of nodes increases due to the increase in dropped packets.

CASE B: Throughput of sending packets

The graph is plotted for the throughput of sending packets against the simulation time. Throughput of sending packets is the number of packets sent per time interval length(TIL).

1. Number of nodes: 4

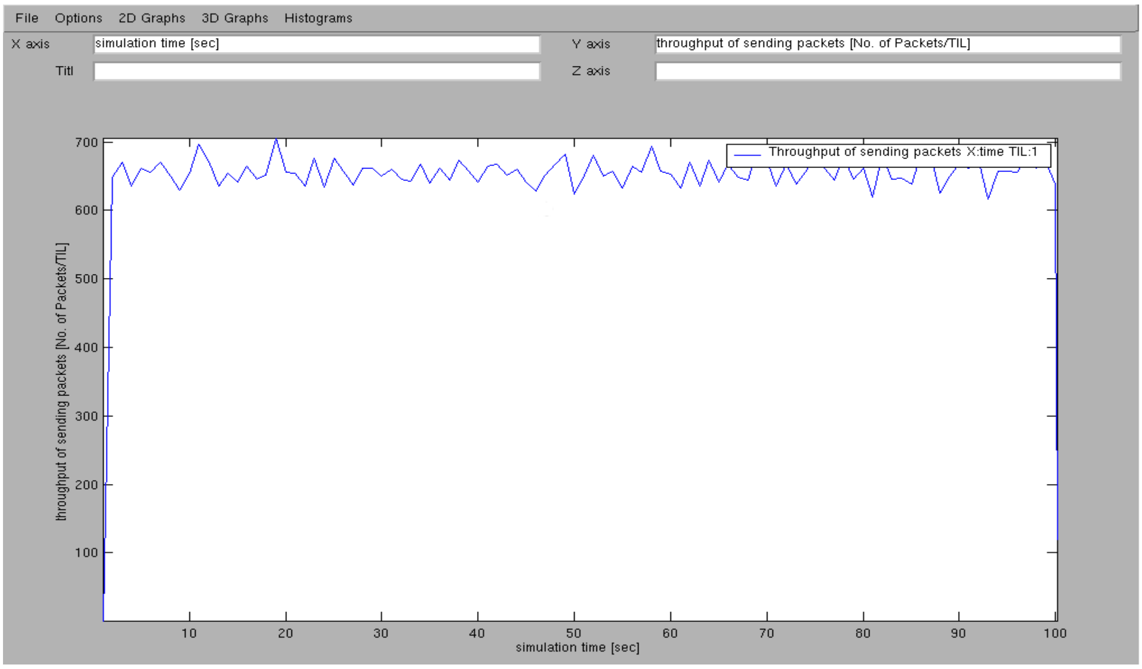


Fig 5.4: Throughput of sending packets for 4 nodes

When the number of nodes are 4, the throughput peaks to 670 packets/TIL approx. Between the simulation time 0 to 2 secs. After that, the throughput varies between 600 packets/TIL and 700 packets /TIL along with simulation time.

2. Number of nodes : 10

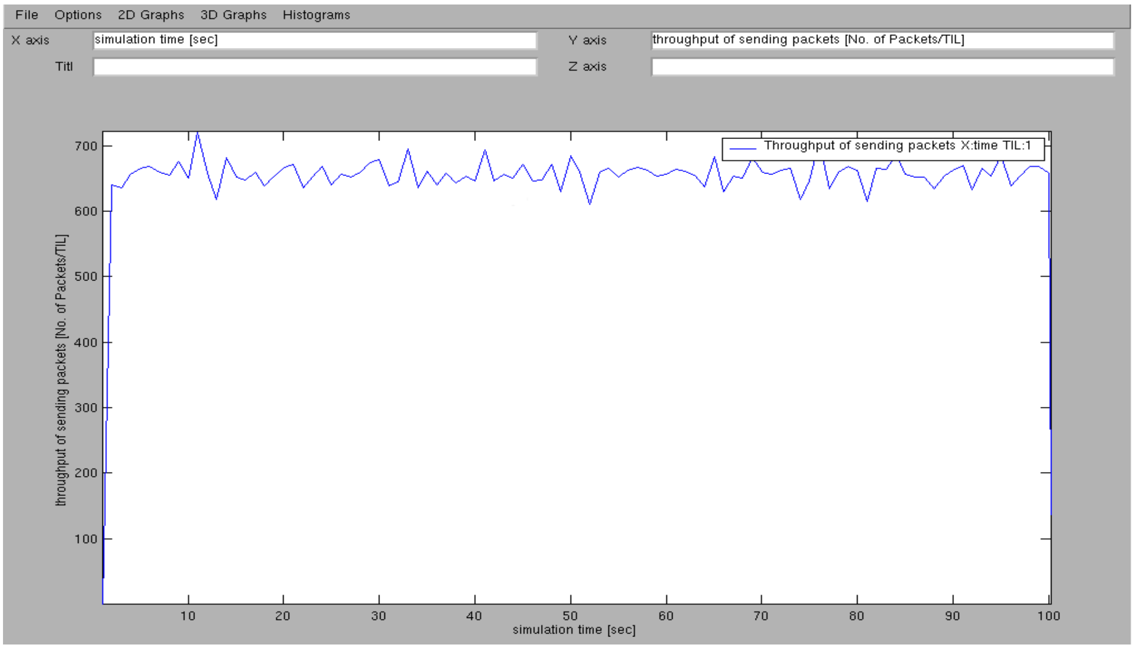


Fig 5.5: Throughput of sending packets for 10 nodes

When the number of nodes are increased to 10, initially the throughput peaks to 630 packets/TIL approx and after that it varies between 600 packets/TIL and 700 packets/TIL in the simulation time 2 to 100 secs.

3. Number of nodes: 15

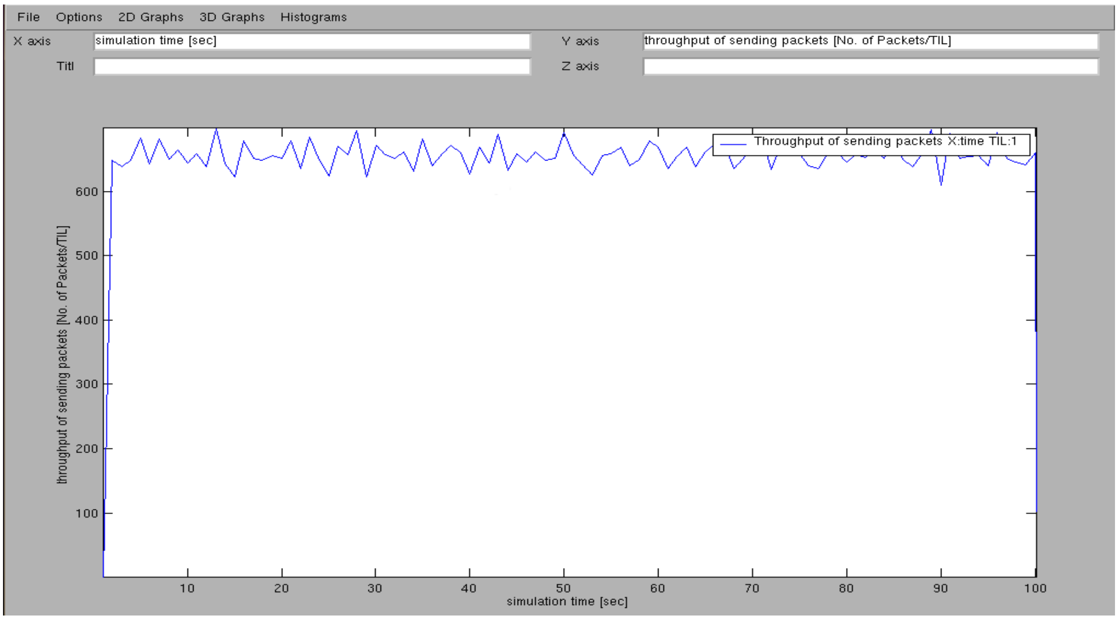


Fig 5.6: Throughput of sending packets for 15 nodes

As the nodes increases to 15, initially the throughput rises to 650packets/TIL approx. and after that it varies between 600packets/TIL and 700packets/TIL in similar fashion.

INFERENCE: The throughput always remain approximately same and lies between 600packets/TIL and 700 packets/TIL. The average throughput leading to 650 packets/TIL. As it can be concluded from the above values that the number of nodes have no effect on the throughput calculated.

CASE C : Throughput of receiving packets

The graph is plotted for the throughput of received packets and the simulation time. Throughput of received packets is number of packets received per time interval length(TIL). The simulation time is measured in seconds.

1. Number of nodes: 4

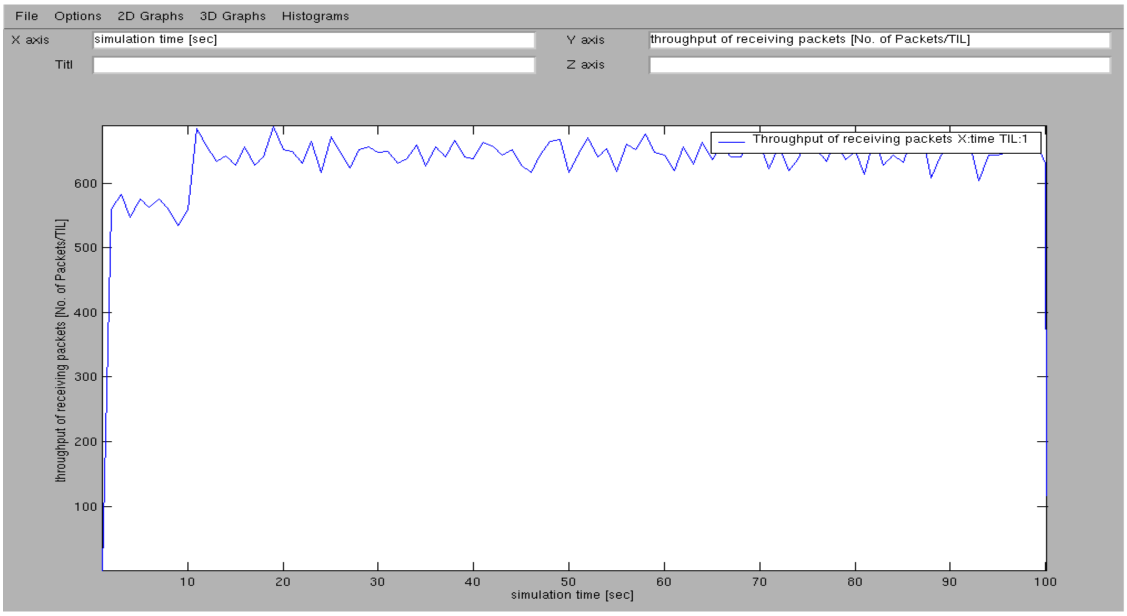


Fig 5.7: Throughput of receiving packets for 4 nodes

The throughput increases from 0 to 580packets/TIL in 2 secs as initially the packets are not received because they have not yet reached their destination nodes. There is sudden rise in throughput from 540 to 700 packets/TIL at simulation time 10. There is no continuous flow of received packets auntil the simulation time 10 and this time is known as *warm-up time*. After warm-up time, there is continuous flow of packets.

2. Number of nodes : 10

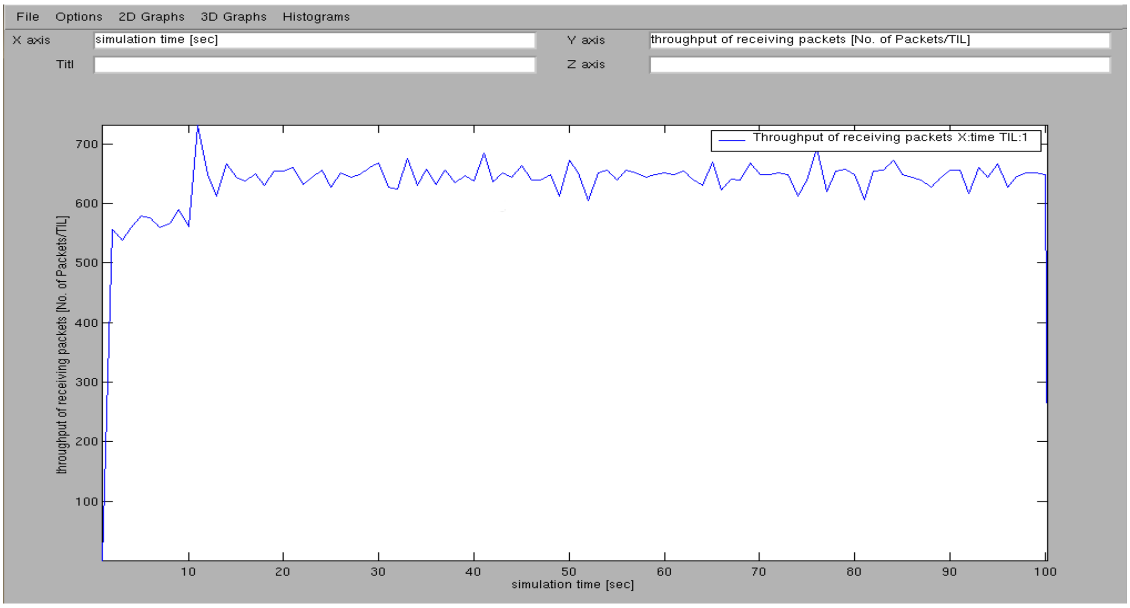


Fig 5.8: Throughput of receiving packets for 10 nodes

As the number of nodes increases to 10, throughput peaks upto 550packets/TIL in few seconds. The warm-up time in this case increases to 12 and the throughput rises from 580 to 720 packets/TIL. After that interval throughput varies between 600 to 700 packets/TIL only till the simulation ends.

3. Number of nodes : 15

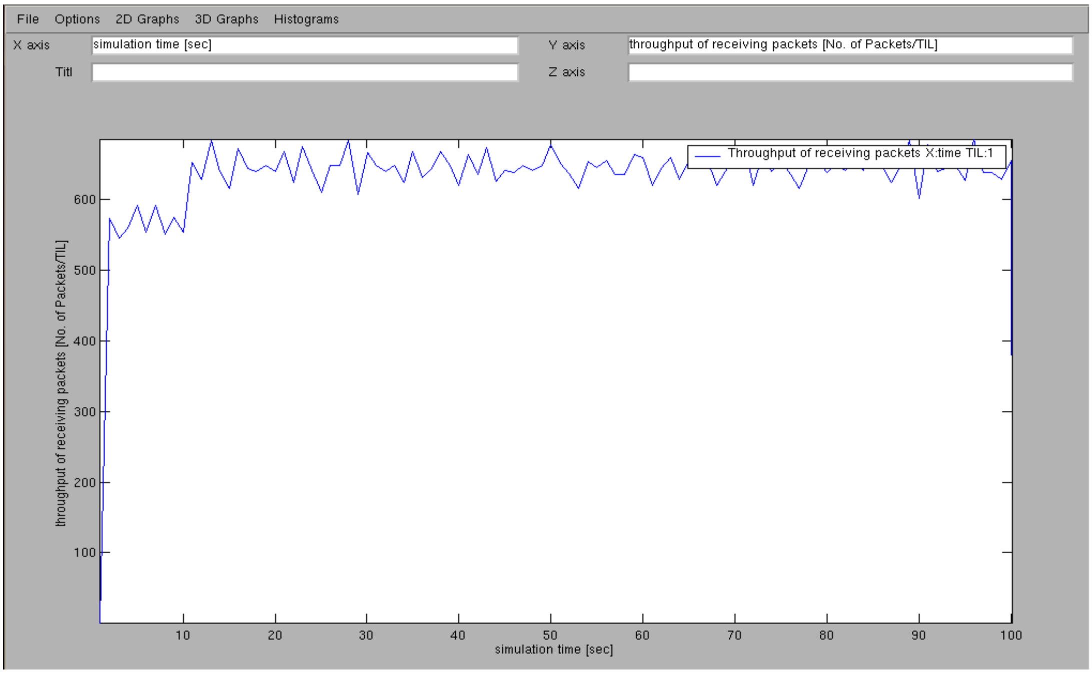


Fig 5.9: Throughput of receiving packets for 15 nodes

With 15 number of nodes, the throughput rises to 580 packets/TIL initially. The warm-up time in this case is 13 seconds and the change in throughput occurs from 550packets/TIL to 660 packets/TIL. Throughput then varies between 600 packets/TIL and 700 packets/TIL only.

INFERENCE: The average throughput of receiving packets varies between 600 packets/TIL and 700 packets/TIL after the warm-up time. The warm-up time increases as the number of nodes increases. After warm-up time, the abrupt change in throughput decreases with number of nodes as it was 160 packets/TIL in 4 nodes, 140 packets/TIL in 10 nodes and 110 packets/TIL in 15 nodes.

CASE D: Throughput of dropping packets

The graph is plotted for the throughput of dropped packets against the simulation time. The throughput of dropped packets is the number of dropped packets per time interval length(TIL). The throughput is measured in number of packets per time interval length and simulation time is measured in seconds.

1. Number of nodes : 4

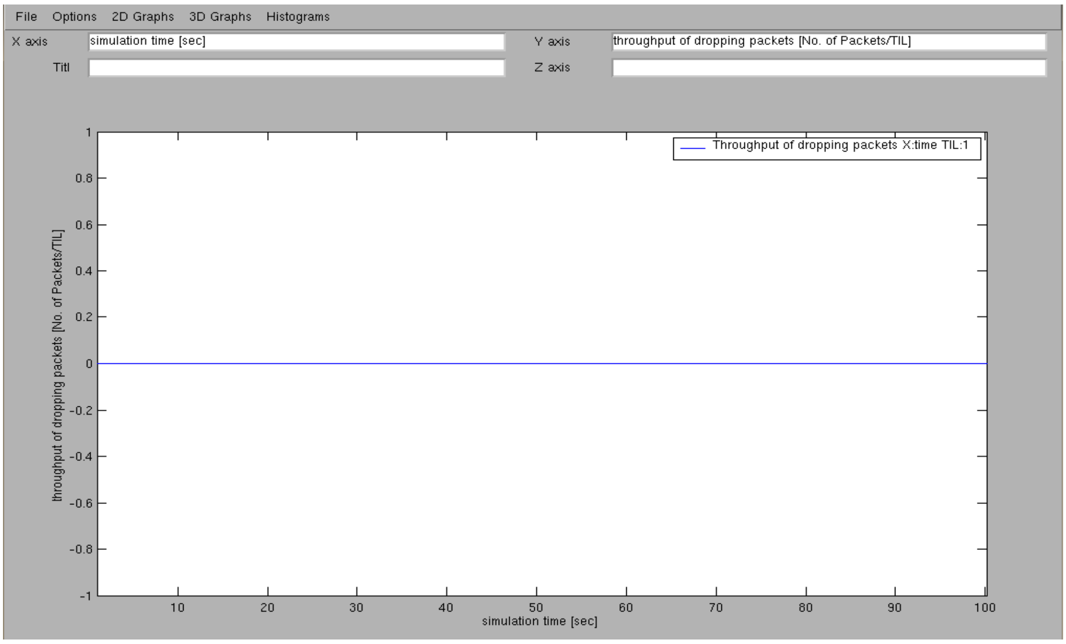


Fig 5.10: Throughput of dropping packets for 4 nodes

As the number of dropped packets in this case was 0 so the throughput will remain 0 for the entire simulation.

2. Number of nodes: 10

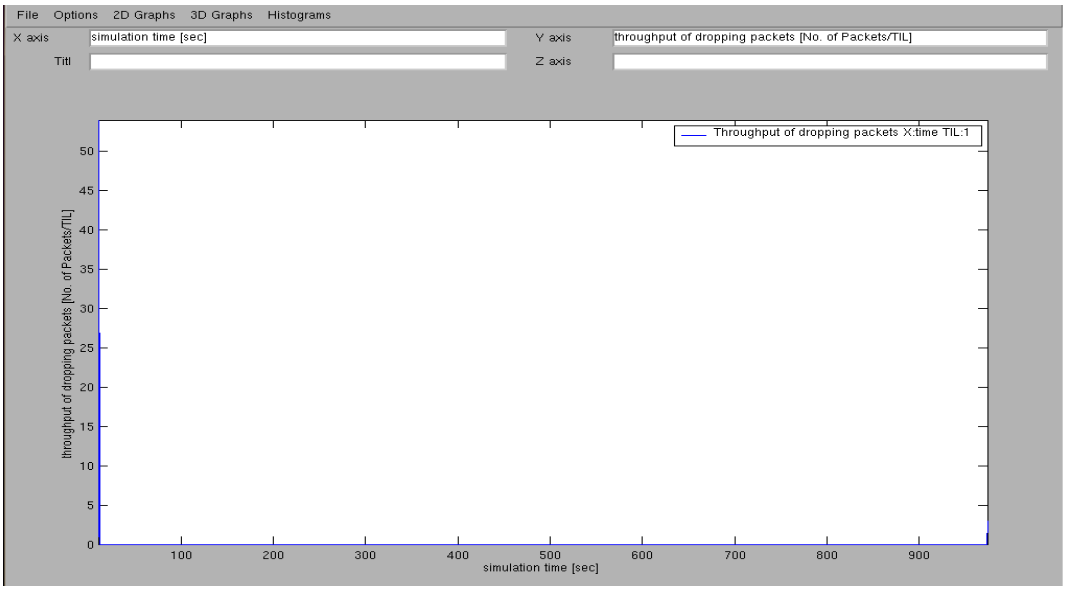


Fig 5.11: Throughput of dropping packets for 10 nodes

When the number of nodes are 10, the number of dropped packets is 55 in the smulation time from 0 to 5 secs. So the throughput rises to 55 packets/TIL initially.

3. Number of nodes : 15

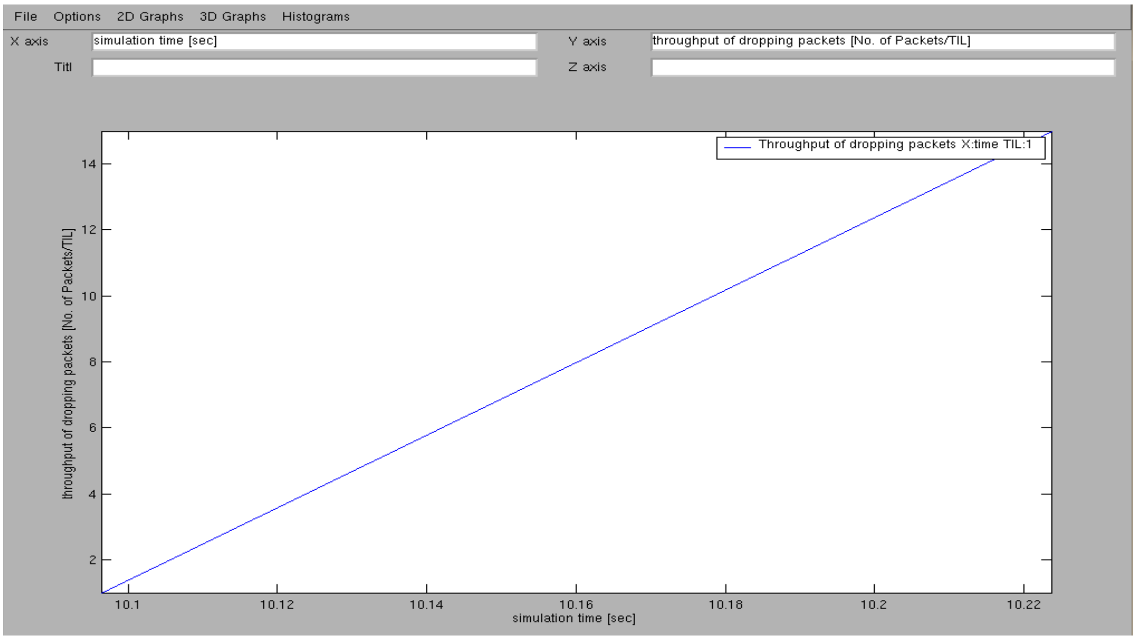


Fig 5.12: Throughput of dropping packets for 15 nodes

In this case, the number of dropped packets increases evenly as the simulation time increases, so it forms the straight line graph between the throughput of dropped packets and the simulation time.

INFERENCE: As the number of nodes increases, the steepness of graph increases as the throughput of dropping packets increases with the number of nodes.

CASE E: Packet Size vs Average Throughput of sending packets

The main purpose of this graph is to check how the average throughput is affected by the size of the packet for the sent packets. The packet size is measured in bytes and the average throughput is measured in the number of packets per time interval length.

1. Number of nodes: 4

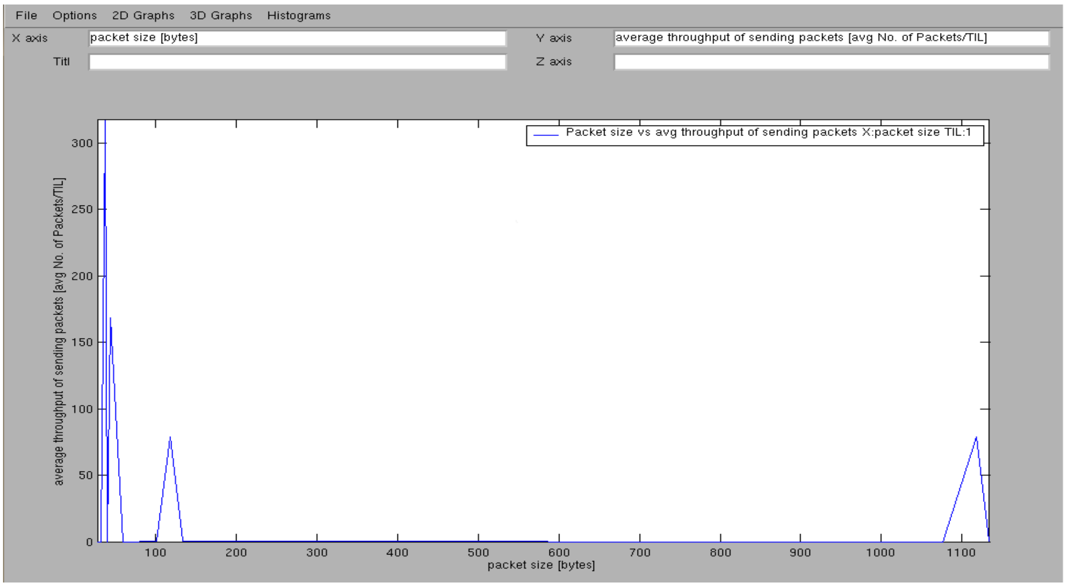


Fig 5.13: Packet Size vs Average Throughput of sending packets for 4 nodes

In this case, the average throughput of sending packets increases sharply for the packet size between 0 to 50 bytes. Then, the average throughput shows little change of 75packets/TIL when the packet size is between 100 and 130 bytes. The average throughput does not show any value for packet sizes from 130 to 1080 bytes indicating that these packet sizes are not supported by the network. Again, the packet size from 1080 to 1150 bytes are supported showing the throughput value of 75 packets/TIL.

2. Number of nodes: 10

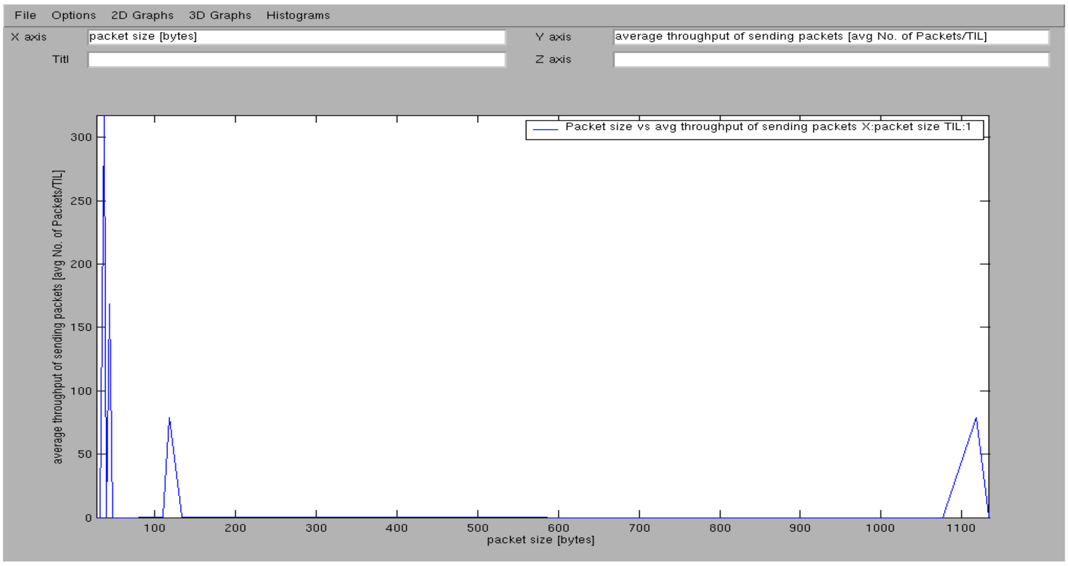


Fig 5.14: Packet Size vs Average Throughput of sending packets for 10 nodes

Here with number of nodes as 10, again the average throughput of sending packets increases abruptly between the packet size 1 to 30 bytes. After that, packet sizes supported are between 110 to 130 bytes and between 1080 to 1150 bytes. The othet packet sizes are not supported by the network.

3. Number of nodes : 15

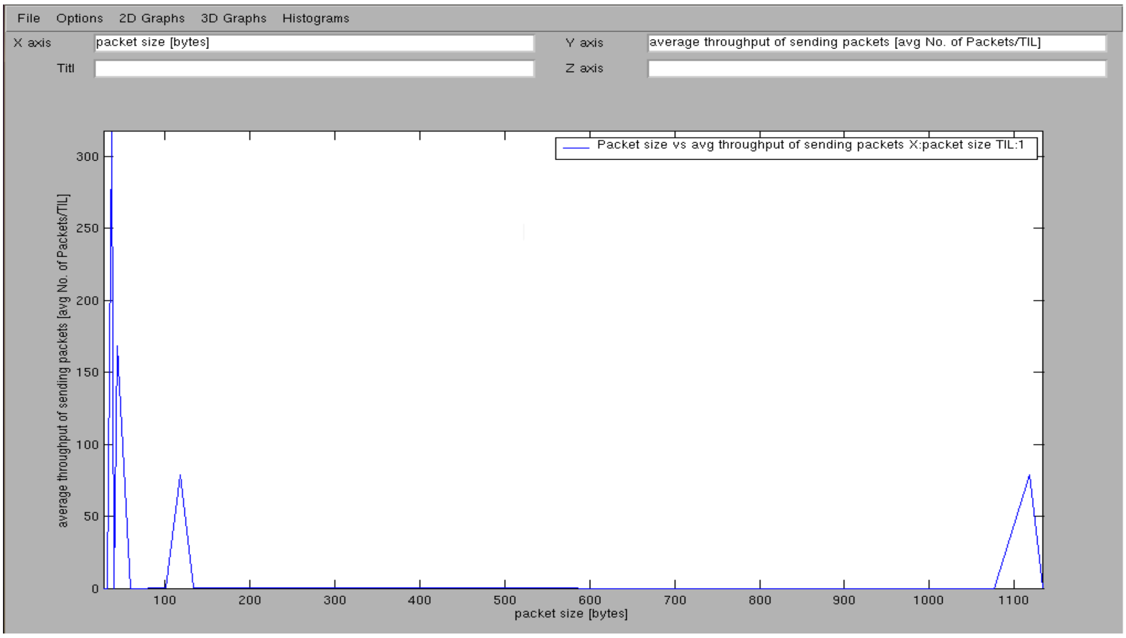


Fig 5.15: Packet Size vs Average Throughput of sending packets for 15 nodes

With number of nodes 15, similarly packet sizes supported are 0 to 50 bytes, 100 to 130 bytes and 1080 to 1150 bytes. There is sudden rise of the throughput of sending packets at lower size of packets.

INFERENCE: The packet size for which the throughput of sending packets has shown values is for smaller packet sizes below 40 bytes. The throughput of sending packets also increases showing small values between packet sizes 100 -130 bytes and 1080 -1150 bytes.

CASE F : Packet Size vs Average Throughput of receiving packets

The purpose of this graph is similarly to check how the average throughput is affected by the size of the packet, but for the received packets. Average throughput is plotted on Y-axis and the packet size is plotted on X-axis.

1. Number of nodes : 4

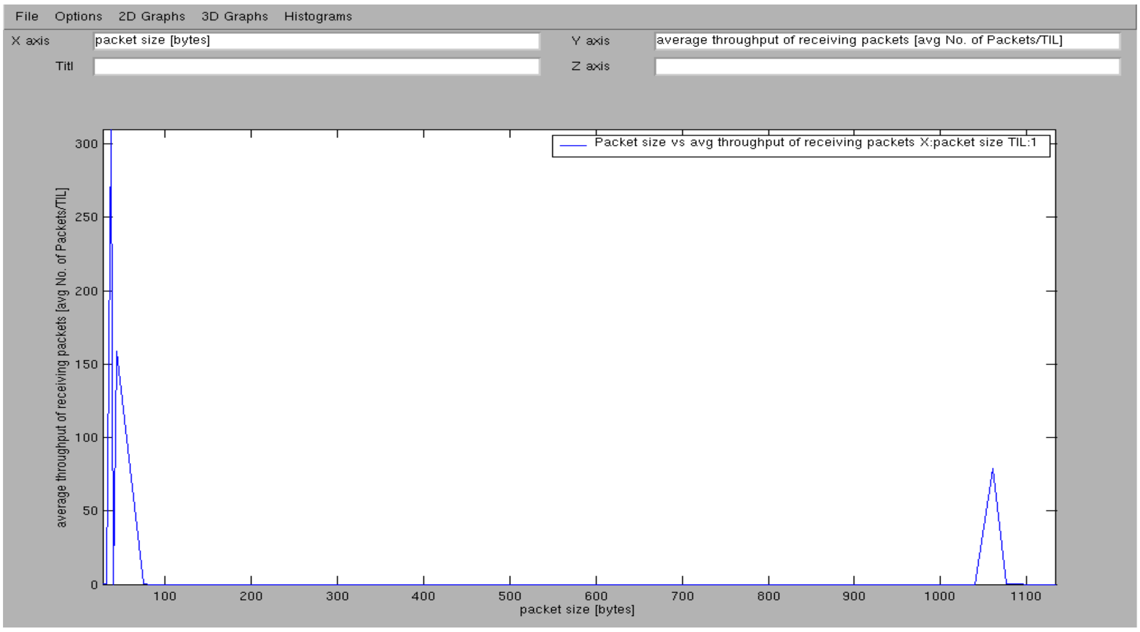


Fig 5.16: Packet Size vs Average Throughput of receiving packets for 4 nodes

With number of nodes 4, the average throughput of receiving packets increases abruptly for the 10 bytes packet size and then it increases upto 160packets/TIL for the packet size till 70 bytes. The throughput value also increases to 75packets/TIL for the packet size from 1050 to 1080 bytes.

2. Number of nodes : 10

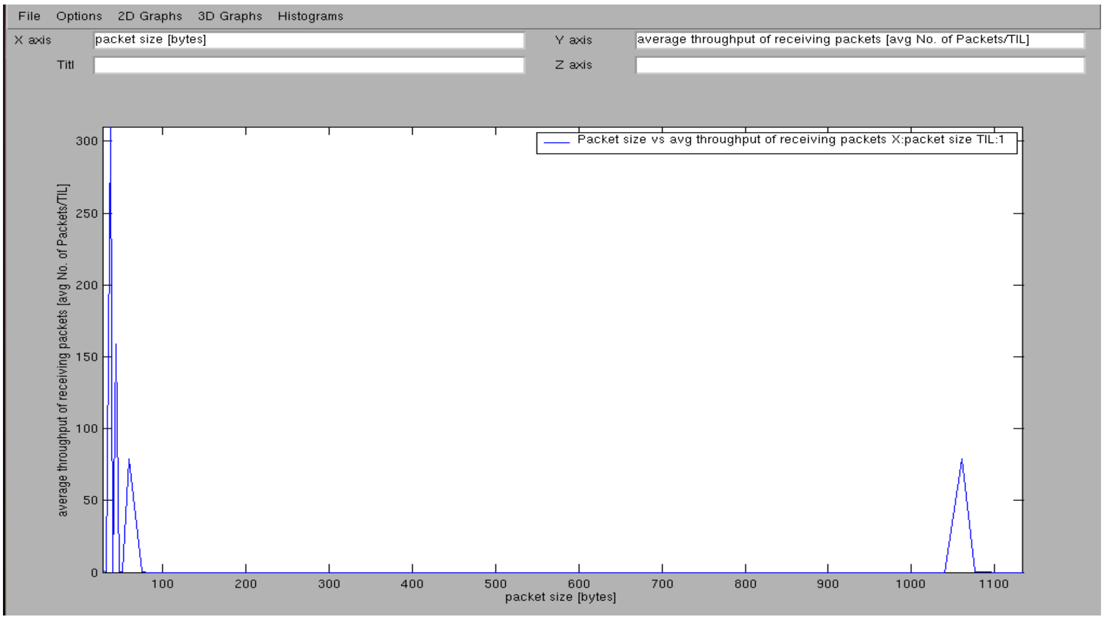


Fig 5.17: Packet Size vs Average Throughput of receiving packets for 10 nodes

Here also, the average throughput shows sharp rise for packet sizes upto 30 bytes. The average throughput also rises to value of 75 for the packet sizes from 30 to 70 bytes and 1050 to 1080 bytes.

3. Number of nodes : 15

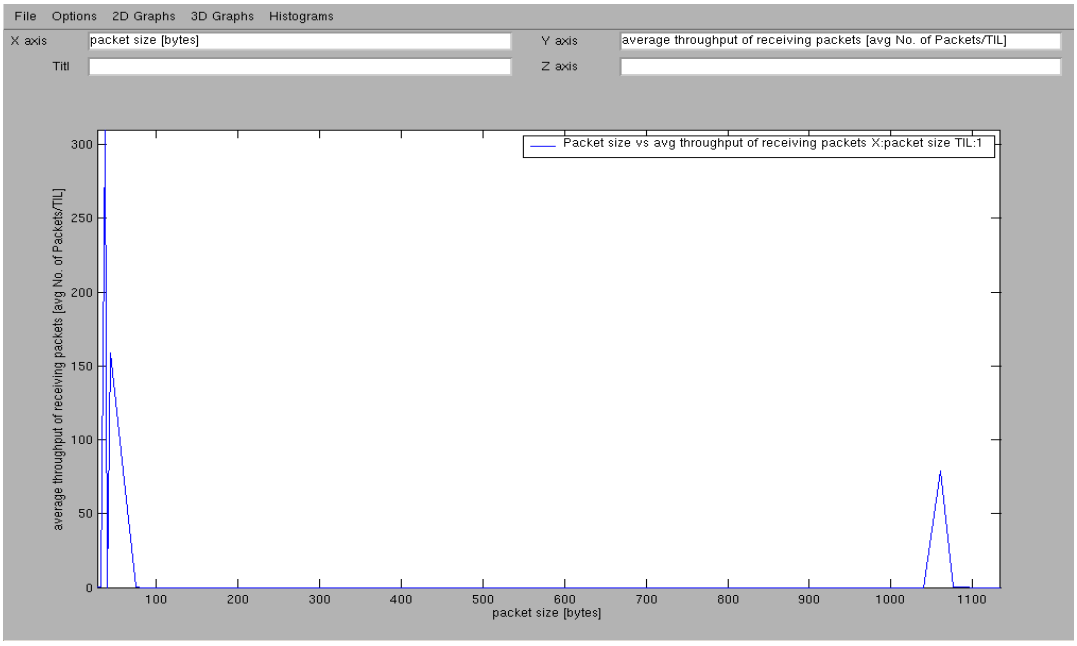


Fig 5.18: Packet Size vs Average Throughput of receiving packets for 15 nodes

With number of nodes 15, the throughput increases for packet size below 70 bytes. The average throughput also increases to value 75 packets/TIL for packet size from 1050 to 1080 bytes.

INFERENCE: The average throughput of received packets increases for packet size below 70 bytes and then it shows the value of 75 packets/TIL for packet sizes between 1050 to 1080 bytes for all mobile nodes 4, 10 and 15.

CASE G: Packet Size vs Average Throughput of dropping packets

The purpose of this graph is similarly to check how the average throughput of dropping packets is affected by the size of the packet. Average throughput is plotted on Y-axis and the packet size is plotted on X-axis.

1. Number of nodes : 4

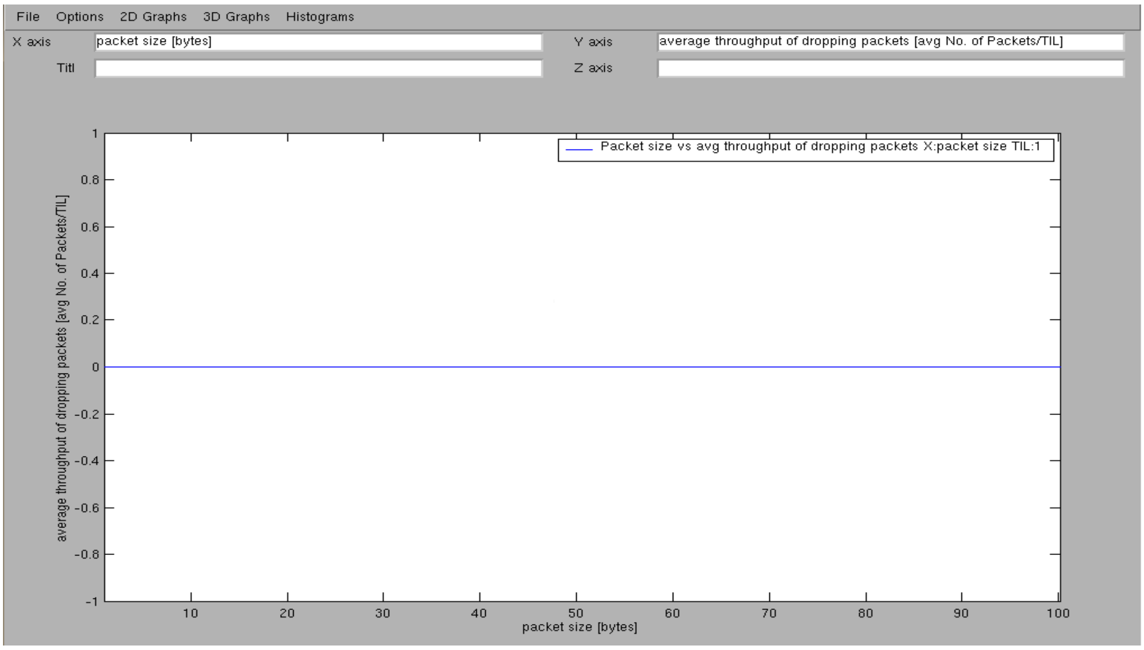


Fig 5.19: Packet Size vs Average Throughput of dropping packets for 4 nodes

With 4 number of nodes, there were no dropped packets, so the average throughput of dropped packets is 0 packets/TIL. Therefore the packet size does not have any effect on average throughput of dropping packets.

2. Number of nodes : 10

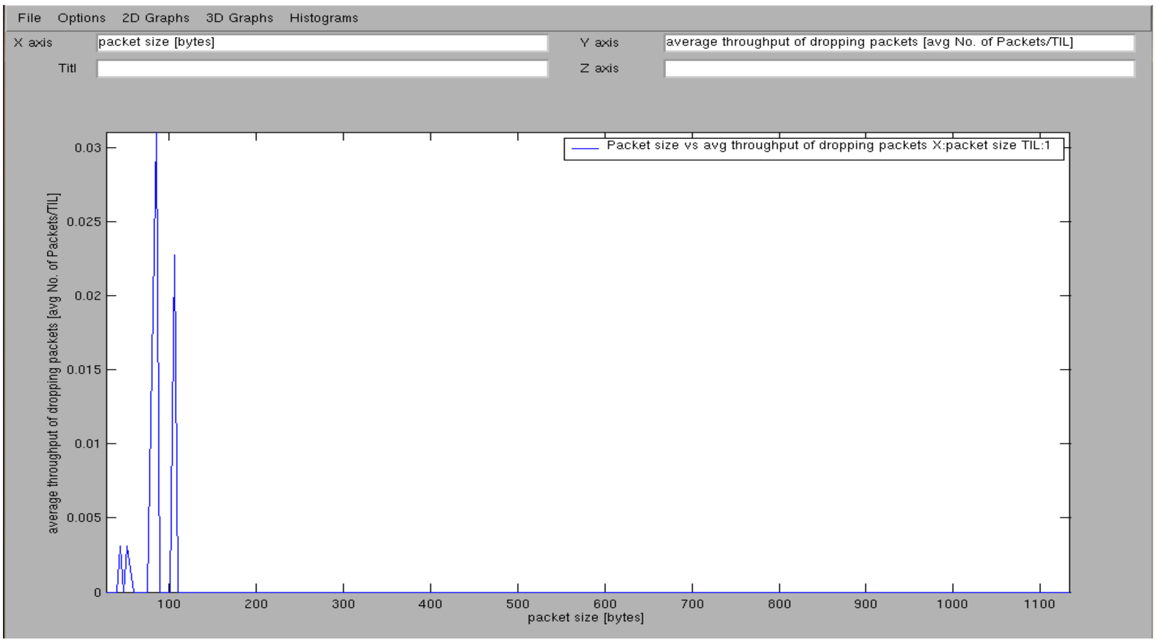


Fig 5.20: Packet Size vs Average Throughput of dropping packets for 10 nodes

As the number of nodes are increased to 10, average throughput of dropping packets is concentrated around the packet size 100 bytes. The value increases above than 0.03 packets/TIL for packet sizes 60 to 80 bytes and it increases to value 0.023 packets/TIL for packet sizes 100 to 110 bytes.

3. Number of nodes: 15

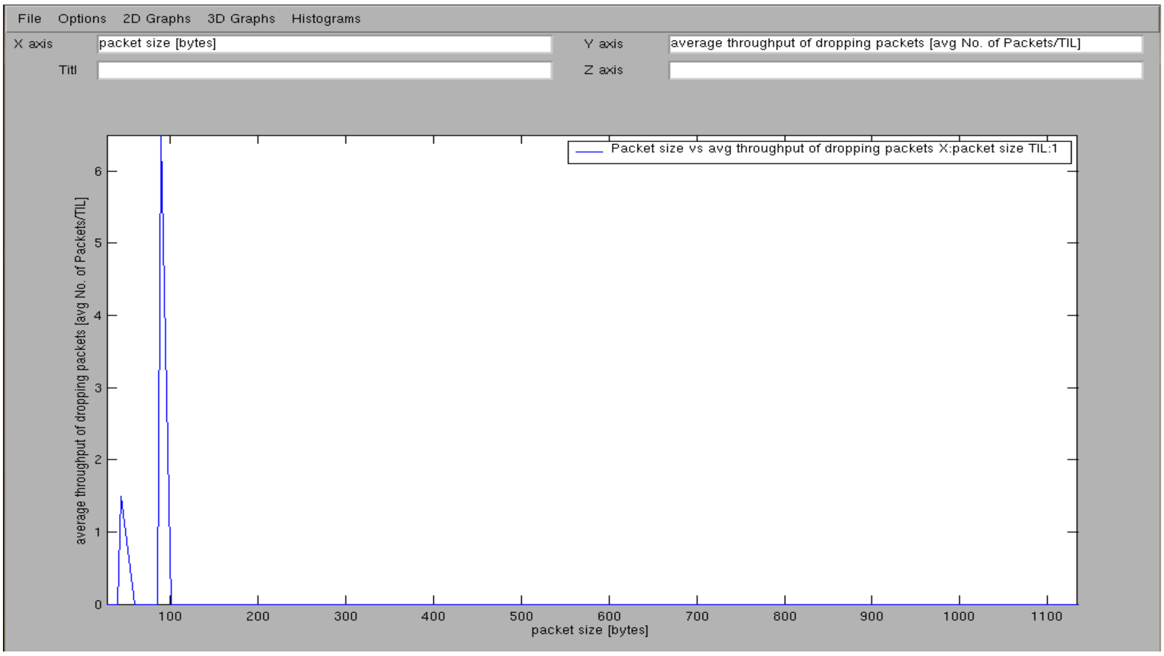


Fig 5.21: Packet Size vs Average Throughput of dropping packets for 15 nodes

With the number of nodes 15, the highest value of throughput of dropping packets is recorded for packet sizes between 80 to 100 bytes. So the highest number of dropped packets are from sizes 80 to 100 bytes.

INFERENCE: The highest ratio of dropped packets are of lower sizes around 100 bytes. The maximum number of dropped packets are of size below 110 bytes.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

VANETs are an important research field which is emerging nowadays. It defines the important technology required to support Intelligent Transportation Systems (ITS) applications. Instead of developing the all new protocols for VANET, an insight has been made to already existing MANET protocols such as DSR.

DSR is a reactive routing protocol and has been extended on VANET for simulations under MOVE and SUMO with mobility models taken from MOVE. The experiment has been conducted for 4 nodes, 10 nodes and 15 nodes. The analysis against different parameters such as sum of dropped packets, throughput and packet size has been done and following observations has been made:

1. The number of dropped packets increases with increase in number of nodes.

2. The throughput of sending packets is almost same and do not show much variation with the changing number of nodes.

3. The throughput of receiving packets is almost same but the warmup time increases as the number of nodes increase.

4. The throughput of dropping packets increases with the increase in the number of nodes from 4 to 15, as the number of dropped packets increases.

5. The throughput of sending packets is more with the packets of smaller size in all the number of mobile nodes.

6. The throughput of received packets is also more for smaller sized packets.

7. The more number of dropped packets are recorded for the packet sizes between 80 to 100 bytes for all the three scenarios of 4, 10 and 15 nodes.

The maximum nodes considered in the work are 15 due to hardware constraints of the computer. So, it would be interesting to see the behaviour of the DSR protocol for the higher number of nodes, for which the machine of higher configuration will be required. Further, we can also analyze the behaviour of DSR protocol by taking the different mobility models as now we have only used the manually created map with the help of MOVE which is similar to Manhattan Mobility Model. Furthermore, the analysis of the data collected from real time performance of VANET test bed will also be beneficial.

REFERENCES *ANNEXURE-I*

[1] Aamir Hassan, "*VANET Simulation*", Master's thesis at School of Information Science, Computer and Electrical Engineering, Halmstad University accessed on 11 May, 2012.

[2] Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "*A Survey on sensor networks*", IEEE Communication Magazine, Aug. 2002.

[3] A K Gupta, H Sadawarti, A K Verma, "*Performance analysis of MANET routing protocols in different mobility models*", Computer Science Journals 2009. *http://www.mendeley.com/research/performance-analysis-manet-routing-protocols-different-mobility-models/* accessed on 11 May, 2012.

[4] A. Shastri, R. Dadhich, Ramesh C. Poonia, "*Performance Analysis Of On-Demand Routing Protocols For Vehicular Ad-Hoc Networks*" , International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 4, August 2011.

[5] Clara, "*Wireless Networking Basics*", NETGEAR, Inc. , October 2005.

[6] C. Chong and S. P. Kumar, " S*ensor Networks: Evolution, Oppurtunities, and Challenges*", in proceedings of the IEEE, Vol. 91, No. 8, Aug. 2003.

[7] C. K. Toh, "*Ad Hoc Mobile Wireless Networks: Protocols and Systems*" ,Prentice Hall Englewood Cliff, NJ 07632, 2002.

[8] C. Murthy and B. Manoj, "*Ad hoc Wireless Networks: Architectures and Protocols*," Prentice Hall *PTR*, 2005.

[9] David B. Johnson, "*Validation of Wireless and Mobile Network Models and Simulation*", In Proc. DARPA/NIST Workshop on Validation of Large-Scale Network Models and Simulation Fairfax, VA, May 1999.

[10] David B. Johnson, David A. Maltz and Josh Broch, "*DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks*", Computer Science Department, Carnegie Mellon University ,Pittsburgh, PA 15213-3891.

[11] Divya Sharma and Ashwani Kush, "*PERFORMANCE COMPARISON OF ENERGY EFFICIENT AODV PROTOCOLS*", International Journal of Computing and Business Research (IJCBR), Volume 2, Issue 1, 2011.

[12] *Dynamic Source Routing protocol RFC* available at http*://www.ietf.org/mailarchive/web/ietf-announce/current/msg03407.html* , last accessed on 11 May, 2012.

[13] D. P. Agrawal and Qing-An Zeng, "*Introduction to wireless and Mobile Systems*," *Brooks/Cole, 2005.*

[14] F. Karnadi, Z. Mo, K.-C. Lan, "*Rapid Generation of Realistic Mobility Models for VANET*", in Proc. of the IEEE Wireless Communication and Networking Conference (WCNC‟ 07), March 2007.

[15] Gayathri Chandrasekaran, "*VANETs: The Networking Platform for Future Vehicular Applications*", Department of Computer Science, Rutgers University.

[16] Geetha Jayakumar, Gopinath Ganapathi, "*Reference Point Group Mobility and Random Waypoint Models in Performance Evaluation of MANET Routing Protocols*", in Journal of Computer Systems Networks and Communications, 2008.

[17] Harminder S Bindra, Sunil K Maakar, A L Sangal, "*Performance Evaluation of Two Reactive Routing Protocols of MANET using Group Mobility Model*", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 10, May 2010.

[18] "*How stuff works",* website available at *http://computer.howstuffworks.com/wireless-network1.htm*, last accessed on 11 May, 2012.

[19] "*Institute of Electrical and Electronics Engineer(IEEE)*" website available at *http://www.ieee.org* last accessed on 11 May, 2012.

[20] Jaap, S., Bechler, M., Wolf, L. (2005), "*Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in City Traffic Scenarios*," Proceedings of the 5th International Conference on Intelligent Transportation Systems (ITS) Telecommunications, June, 2005.

[21] Jerbi, M., Senouci, S.-M., Meraihi, R., and Ghamri-Doudane, Y. (2007), "*An improved vehicular ad hoc routing protocol for city environments*," Communications, 2007. ICC '07. IEEE International Conference, pp. 3972–3979, 24-28 June 2007.

[22] Jerome Harri, F Filali, C Bonnet, "*Mobility models for vehicular ad hoc networks: a survey and taxonomy*" in IEEE Communications Surveys Tutorials, 2009.

[23] Jesus Gabriel Balderas Lopez, "*A Survey Of Geographic Routing Protocols for Vehicular Ad Hoc Networks (VANETs)*" , New Mexico State University, 2010.

[24] Josiane Nzouonta, Neeraj Rajgure, Guiling (Grace) Wang and Cristian Borcea, "VANET *Routing on City Roads Using Real-Time Vehicular Traffic Information*", IEEE Transactions On Vehicular Technology, Vol. 58, No. 7, September 2009.

[25] Kapang Lego, Pranav Kumar Singh and Dipankar Sutradhar, "*Comparative Study of Adhoc Routing Protocol AODV, DSR and DSDV in Mobile Adhoc Network*" , Indian Journal of Computer Science and Engineering , Vol. 1 No. 4, January 2011.

[26] Karp, B. and Kung, H. T, "*GPSR: greedy perimeter stateless routing for wireless networks*", In Mobile Computing and Networking, pages 243-254, 2000.

[27] Ke Liu "*Network Simulator 2: Introduction*", Dept. Of Computer Science, SUNY Binghamton, available at *http://www.cs.binghamton.edu/~kliu/cs580t/ns2pre.pdf*.

[28] Liu Tie-Yuan, Chang Liang and Gu Tian-Long, "*Analyzing the Impact of Entity Mobility Models on the Performance of Routing Protocols in the MANET*", in Proc 3rd Int Conf Genetic and Evolutionary Computing WGEC 2009.

[29] Mahmood Hasanlou *"Introduction to mobility and network simulator 2".*

[30] Maria Fazio, Claudio Enrico Palazzi, Shirshanka Das and Mario Gerla, *" Facilitating Real-time Applications in VANETs through Fast Address Auto-configuration*," In Proc. 3rd IEEE CCNC International Workshop on Networking Issues in Multimedia Entertainment, (CCNC/NIME 2007), IEEE Communications Society, Las Vegas, USA.

[31] *Mobile Ad-hoc Networks (MANET)* available at *http://www.ietf.org/html.charters/manetcharter.html* last accessed on 11 May, 2012.

[32] Mohd Izuan and Mohd Saad , "*Performance Analysis of Random-Based Mobility Models in MANET Routing Protocol*" in European Journal of Scientific Research, 2009.

[33] M K Jeya Kumar, R S Rajesh, "*A Survey of MANET Routing Protocols in Mobility Models*" in International Journal Of Soft Computing, 2009.

[34] Naumov, V., Gross and T.R. , "*Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks*," INFOCOM 2007. 26th IEEE International Conference on Computer Communications. pp.1919-1927, 6-12 May, 2007.

[35] Pei, G., Gerla, M., and Chen, T.-W. , "*Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks*", Proc. ICC 2000, New Orleans, LA, June 2000.

[36] Potnis, N., & Mahajan, A., "*Mobility Models for Vehicular Ad Hoc Network Simulations*", In Proceedings of the 44th ACM Southeast regional conference (pp 746-747). Melbourne, Florida, 2005.

[37] Pranav Kumar Singh, Kapang Lego and Dr. Themrichon Tuithung, "*Simulation based Analysis of Adhoc Routing Protocol in Urban and Highway Scenario of VANET*", International Journal of Computer Applications (0975 – 8887) Volume 12– No.10, January 2011.

[38] Priti Garg and Asma Tuteja, "*Comparative Performance Analysis of Two Ad-hoc Routing Protocols*",International Conference on Network and Electronics Engineering IPCSIT, vol.11, 2011.

[39] P. Nain, D. Towsley, B. Liu, and Z. Liu, "*Properties of Random Direction Models*," INRIA technical report RR-5284, July 2004.

[40] Rainer Baumann, "*Engineering and Simulation of Mobile Adhoc Routing Protocols for VANET on highways and in cities*", ETH Zurich 2004.

[41] Rajeshwar Singh, Dharmendra K Singh and Lalan Kumar, "*Performance Evaluation of DSR and DSDV Routing Protocols for Wireless Ad Hoc Networks*", International Journal of Advanced Networking and Applications, Volume: 02, Issue: 04, Pages: 732-737, 2011.

[42] Rakesh Kumar Jha, Suresh V. Limkar and Dr. Upena D. Dala, "*A Performance Comparison of Routing Protocols(DSR and TORA) for Security Issue In MANET(Mobile Ad Hoc Networks)*", IJCA International Journal Of Computer Applications, Special Issue on "Mobile Ad-hoc Networks" MANETs, 2010.

[43] Rakesh Kumar, Mayank Dave, " *A Comparative Study of Various Routing Protocols in VANET*", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011.

[44] Ram Shringar Raw and Sanjoy Das, "*Performance Comparison of Position-based Routing Protocols in Vehicle-to-Vehicle (V2V) Communication*", in IJEST International Journal Of Engineering Science And Technology, Jan 2011.

[45] Ricki G. Ingalls, "*Introduction To Simulation*", Proceedings of the 2002 Winter Simulation Conference School of Industrial Engineering and Management, 322 Engineering North ,Oklahoma State University ,Stillwater, OK 74078, U.S.A. Available at *http://www.imit.kth.se/courses/2G1126/vt03/Pmodule-vt03/papers/ingalls.pdf*

[46] R. Mangharam, D. Weller, R. Rajkumar, P. Mudalige, and F. Bai, "*Groovenet: A hybrid simulator for vehicle-to-vehicle networks*," in Mobile and Ubiquitous Systems - Workshops, 3rd Annual International Conference , 2006.

[47] R Manoharan, P Ajit Jose Sylvester and M Ramprasath, "*A Robust Routing Protocol for VANETs using Cellular Automata*", International Journal of Recent Trends in Engineering and Technology, Vol. 3, No. 2, May 2010.

[48] Seet, B.-C., Liu, G., Lee, B.-S., Foh, C. H., Wong, K. J., Lee, K.-K. (2004), "*A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications*" Networking 2004.

[49] SUMO (2012). "*Simulation of Urban Mobility".* Accessed on 11 May, 2012 from *http://sumo. sourceforge.net/docs/bibliography.shtml.*

[50] Sunil Taneja and Ashwani Kush, "Performance Evaluation Of DSR and OADV over UDP and TCP Connections", International Journal of Computing and Business Research (IJCBR) Volume 1, No. 1 December – 2010.

[51] Thomas D.C. Little and Ashish Agarwal, "*An Information Propagation Scheme for VANETs*", Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, September 13-16, 2005.

[52] "Trace graph - Network Simulator NS-2 trace files analyser", which can be accessed at http://ns-2.blogspot.in/2008/10/trace-graph-network-simulator-ns-2.html.

[53] Uma Nagaraj, Dr. M. U. Kharat, Poonam Dhamal, "*Study of Various Routing Protocols in VANET*", IJCST International Journal Of Computer Science And Technology, Vol. 2, Issue 4, Oct . - Dec. 2011.

[54] Vasanthi.V, Romen Kumar.M, Ajith Singh.N and M.hemalatha, "*A Detailed Study Of Mobility Models in WirelessSensor Networks*", Journal of Theoretical and Applied Information Technology, Vol. 33, No.1, 15th November 2011.

[55] "*Vehicle Safety Communications Project*" Task 3 Final Report. Technical report, The CAMP Vehicle Safety Communications Consortium, Mar 2005. Sponsored by U.S. Department of Transportation (USDOT). Available through National Technical Information Service, Springfield, Virginia 22161.

[56] Xiaoyan Hong, Mario Gerla and Guangy upei " *A Group Mobility Model for wireless Ad Hoc Networks*", IEEE November 2001.

[57] Yun-Wei Lin, Yuh-Shyan Chen and Sing-Ling Lee, "*Routing Protocols in Vehicular Ad Hoc Networks: A Survey and Future Perspectives*", Journal Of Information Science And Engineering 26, 913-932 ,2010.

PUBLISHED

- Avleen Kaur Malhi and A. K. Verma, " *A Review of Routing Protocols in VANETs*", International Conference On Recent Advances In Electronics And Computer Engineering, Dec 17-18, 2011, Baru Sahib, HP.

COMMUNICATED

- Avleen Kaur Malhi and A. K. Verma, "*Simulation and Analysis of DSR Protocol in VANETs",* International Journal of Computer Science Issues (IJCSI), Vol. 9, Issue 4, July 2012.
- Avleen Kaur Malhi, Tajinder Kaur and A. K. Verma, "*Simulation and Comparison of AODV and DSR Routing Protocols in VANETs*", International Conference on Issues and Challenges in Networking, Intelligence and Computing Technologies(ICNICT-2012), Sep 7-8, 2012, Ghaziabad, India.