CLASSIFICATION AND SIMULATION OF ROUTING PROTOCOLS IN WIRELESS AD-HOC NETWORK USING NS2

A DISSERTATION
SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE AWARD OF THE DEGREE OF

MASTER OF ENGINEERING
(COMPUTER TECHNOLOGY & APPLICATIONS)

BY
SANJEEV KUMAR UPADHYAYA
College Roll No. 08/CTA/03
Delhi University Roll No. 3008

Under the guidance of
Dr. S. K. SAXENA

DEPARTMENT OF COMPUTER ENGINEERING
DELHI COLLEGE OF ENGINEERING
UNIVERSITY OF DELHI

JULY-2005

CERTIFICATE

It is to certify that the work that is being presented in this dissertation entitled ***"Classification And Simulation Of Routing Protocols in Wireless Ad-hoc Network Using NS2"***, in partial fulfillment of the requirement for the award of the degree of **Master of Engineering** in Computer Technology and Application submitted by **Sanjeev Kumar Upadhyaya** (08/CTA/03) to the Department of Computer Engineering, Delhi College of Engineering, is an authentic record of the student's own work carried out under the supervision and guidance of **Dr. S. K. Saxena**, in the Department of Computer Engineering.

The work embodied in this dissertation has not been submitted for the award of any other degree to the best of my knowledge.

Dr. D. Roy Choudhury
Professor and Head
Dept. of Computer Engg.
Delhi College of Engg.
Delhi

Dr. S. K. Saxena
Dept. of Computer Engg.
Delhi College of Engg.
Delhi

ACKNOWLEDGMENTS

I would first like to thank my supervisor, **Dr. S. K. Saxena**, Department of Computer Engineering, for his guidance during my whole project work. I also give extra special thanks to him for dedicating his valuable time whenever I needed to discuss project related works without any delay.

I would like to thank **Dr. D. Roy Choudhury**, Professor and Head, department of Computer Engineering, for providing facilities for this dissertation.

I am also thankful to all my friends who continuously helped and motivated me during the course of this dissertation.

Lastly, I express my respect and regards to my parents, who have been a constant source of inspiration to me.

Sanjeev Kumar Upadhyaya
08/CTA/03
M.E. (CTA)

ABSTRACT

An ad hoc network is a collection of wireless nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration, characterized by node mobility, dynamic topology structure, scarce bandwidth, unreliable media and limited power supply. Nodes in an ad hoc network must cooperate and carry out a distributed routing protocol in order to make multi-hop communications possible.

An ad hoc routing protocol has to carry out in such an environment. In recent years, a vast study has been conducted on the ad hoc routing protocols. Hundreds of different routing protocols have been proposed and published.

In this thesis, we identify various techniques used to classify routing protocols for wireless Ad Hoc Networks which will help in understanding of current protocols and designing of new protocols. This thesis also presents results of packet level simulation with respect to Fraction of packets delivered, End-to-end delay and routing load for a given traffic and mobility model for various routing protocols (DSDV, AODV, DSR, TORA).

The protocols are simulated with ns-2, a widely available experimental network simulator. The ns-2 core component is implemented in C++ and the interfacing is provided through Otcl/Tcl/Tk.

CONTENTS

List of Figures

The history of wireless networks started in the 1970s and the interest has been growing ever since. During the last decade, and especially at its end, the interest has almost exploded probably because of the fast growing Internet. The tremendous growth of personal computers and the handy usage of mobile computers necessitate the need to share information between computers. At present, this sharing of information is difficult, as the users need to perform administrative tasks and set up static, bi-directional links between the computers. This motivates the construction of temporary networks with no wires, no communication infrastructure and no administrative intervention required. Such interconnection between mobile computers is called an *Ad hoc Network*. In such environment, it may be necessary for the mobile computers to take help of other computers in forwarding a packet to the destination due to the limited range of each mobile host's wireless transmission.

In this chapter, we will navigate deeply in Mobile Ad hoc Networks (MANETs) and give a detailed overview about many different points in MANET, such as, classification, some different definitions and applications.

1.1 Wireless Networks

Today, we see two kinds of wireless networks but the difference between them is not as obvious as it seems. The first kind and most used today is a wireless network built on top of a "wired" network and thus creates a *reliable infrastructured wireless network*. The wireless nodes are able to act as bridges in a wired network. This kind of wireless nodes are called base-stations. An example of this wireless network is the cellular-phone networks where a phone connects to the base-station with the best signal quality. When the phone moves out of range of a base-station, it does a "hand-off" and switches to a new base-station within reach. The "hand-off" should be fast enough to be seamless for the user of the network. Other more recent networks are wireless networks for offices that are usually called Wireless Local Area Networks (WLAN). In this kind, there is no infrastructure at all except the participating mobile nodes. This is called an *infrastructure less network* or more commonly an ad hoc network. The word "ad hoc" can be translated

as “improvised” or “not organized” which often has a negative meaning, but the sense in this context is not negative but only describing the dynamic network situation.

All or some nodes within an ad hoc network are expected to be able to route data-packets for other nodes in the network beyond their own transmission-range. This is called *peer-level multi-hopping* and is the base for ad hoc networks that constructs the interconnecting structure for the mobile nodes.

Another classification introduced in [1] classified wireless networks into three different types according to relative mobility of hosts and routers:

1. Fixed wireless network. Fixed hosts and routers use wireless channels to communicate with each other and form a fixed wireless network. An example is a wireless network formed by fixed network devices using directed antennas, as shown in Figure 1-1.

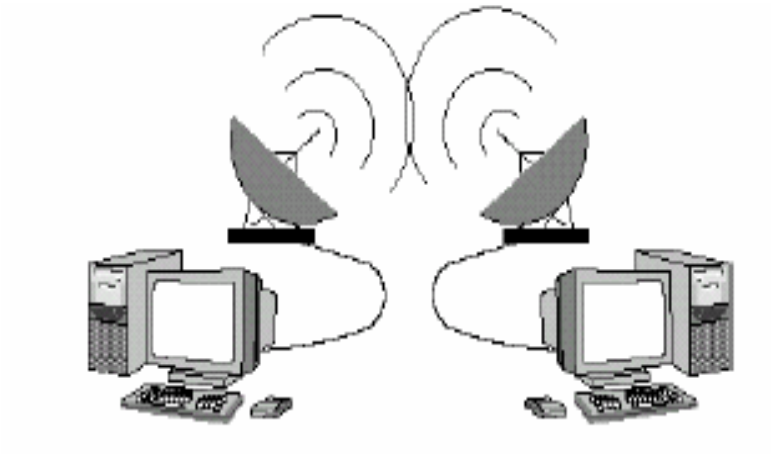

Figure 1.1 An example of a fixed wireless network

2. Wireless network with fixed access points. Mobile hosts use wireless channels to communicate with fixed access points, which may act as routers for those mobile hosts, to form a mobile network with fixed access points. An example is a number of mobile laptop users in a building that access fixed access points, as illustrated in Figure 1-2.

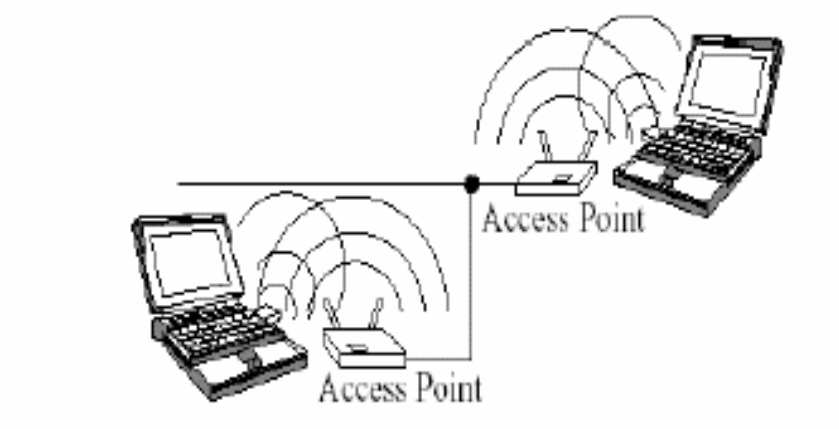

Figure 1.2 An example of a wireless network with access points

3. Mobile ad hoc network. A mobile ad hoc network is formed by mobile hosts. Some of these mobile hosts are willing to forward packets for neighbors. These networks have no fixed routers, every node could be router. All nodes are capable of moving and can be connected dynamically in an arbitrary manner. The responsibilities for organizing and controlling the network are distributed among the terminals themselves. The entire network is mobile, and the individual terminals are allowed to move freely. In this type of networks, some pairs of terminals may not be able to communicate directly with each other and have to rely on some other terminals so that the messages are delivered to their destinations. Such networks are often referred to as *multi-hop* or *store-and-forward* networks. The nodes of these networks function as routers, which discover and maintain routes to other nodes in the networks. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices. Figure 1-3 shows an example for vehicle-to-vehicle network communicating with each other by relying on peer-to-peer routings.

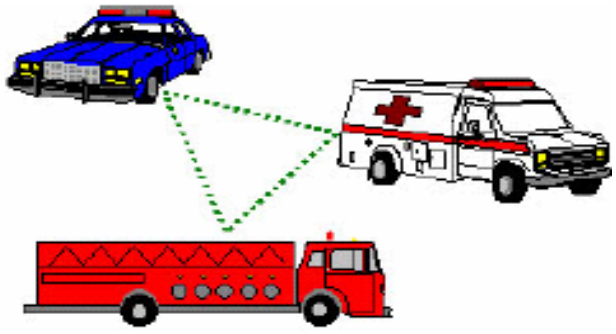

Figure 1.3 An example of a vehicle-to-vehicle network

1.2 Day-to-Day Life Scenario for Wireless Network

Let us imagine here some of the possible use-cases involving mobile terminals and what the future could look like when some technology related problems are solved. Today *Rajeev* is going in a class trip to the Telecom Museum. Like every day, he takes his mobile phone, which has wireless capabilities. On the way to his class, he goes through heterogeneous wireless networks. When he takes the bus, his terminal enters an ad hoc mode. He meets a friend and wants to exchange files in order to use electronic tools for comparing the results of their math homework. As they are rapidly bored with that, they would like to play a game. In order to play with other players, they check if there are other people that already play a network game within the bus or in the cars around. *Rajeev* runs a peer-to-peer application and makes a search for a good game that could amuse him during the short bus trip. Fortunately, someone in the bus has made his terminal available for others to download programs, and he has a great collection of games. *Rajeev* chooses to try out one of the games made available. The game is automatically downloaded, installed and configured on his terminal. He can start playing right away. His friend may join also the game easily on his terminal. Later, at the train station *Rajeev's* terminal automatically connects to the wireless station gateway and enters an access-point mode. As the station gateway offers connectivity to the Internet, *Rajeev* can check his E-mails. When he goes outside the station in order to wait for his friends, he can keep his connectivity to the Internet through the environment management modules of the people walking or waiting in the train station. These modules were downloaded from the station gateway or from some privileged user terminals and they implement an algorithm optimized for the specific wireless station environment. He checks the train departure time in order to make sure that he is not late. Unfortunately, he gets lost, so he establishes a vision call with his classmate that has the possibility to send him a map of the station. When all his classmates and the teachers arrive, they go into the train and *Rajeev's* terminal then connects to the wireless access-point located in the coach he is in. This gateway has a lot of different applications available for download (such as chat or file-exchange peer-to-peer application). As *Rajeev* is feeling hungry, he orders a snack by initiating a VoIP (Voice over Internet Protocol) with the "coach service", the order is then processed by the coach gateway and transmitted to the bar coach. One of the

accompanying teachers has brought his laptop and he uploads the program of the day onto the gateway collaborative-work application. The program is a very big document since it contains pictures, sounds and videos. The students cannot download it on their mobile phones due to the limited memory of their terminals, but they can read it from the gateway. Then a discussion between the students and the teachers starts and they decide to change the program of the day, the teacher makes her document editable and the students start editing it thanks to their PDA. Of course, the collaborative-work application keeps track of all the suggestions issued by the class thanks to a sophisticated yet simple version management system. Once arrived at the museum, the terminals of the students get from the museum wireless gateway all the software components needed to operate. These components include: authentication, resource discovery, naming and routing. The PDA also downloads the terminal part of the museum applications. This way, the young visitors will be able to access additional documentation in relation with their current location in the museum just by downloading a video, or play with a quiz especially designed to have the students focus on their visit. When the students leave the museum, the results of the quiz and the data they have collected are synchronized with their home data.

1.2.1 The Wireless Network Environments

In the previous scenario, the little boy's terminal enters four different wireless networks, each one having specific characteristics.

The bus environment as shown in Figure 1-4 is a pure mobile ad hoc sub-network, i.e. there is no wired infrastructure, and the terminals themselves have to configure the network without centralized administration.

Figure 1.4 Pure ad hoc network (bus environment)

This network itself may be split into several sub-networks, a private network between

Rajeev and his friend, and a "public" network initiated by someone who makes publicly available games. The previous scenario shows that the two networks may be merged dynamically. The train station as shown in Figure 1-5 is an ad hoc sub-network operating in access-point mode, i.e. the terminals use the access-points in the station as bridges to communicate with each other. At the MAC layer, in access-point mode, terminals send and receive messages only from the access point, whereas in ad hoc mode every terminal sends messages to every other. The wireless network at the train station also offers Internet connectivity and multi-hop routing (to allow people walking a small distance from the train station to keep their sessions open).

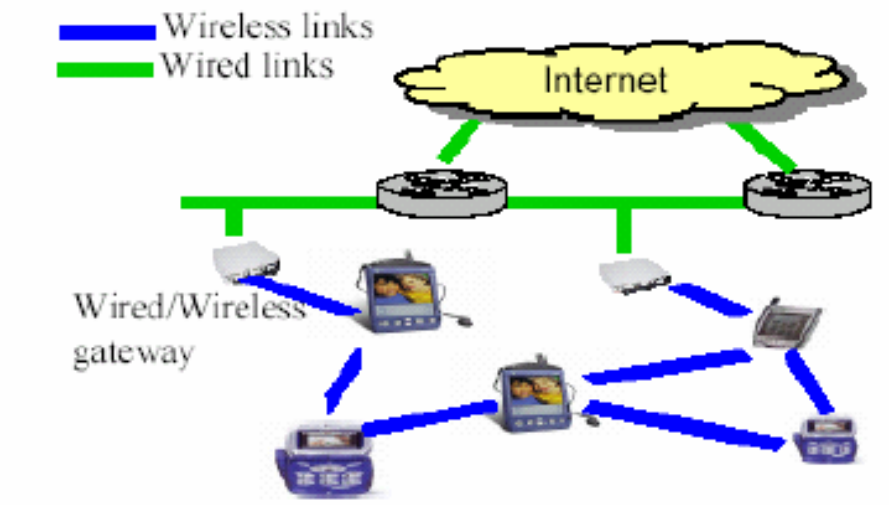

Figure 1.5 The train station environment

Inside the train, wireless terminals can operate in access-point mode (which is more efficient than ad hoc mode) because there is one access point per coach. All the terminals switch automatically from the network in the station to the network in the coach: more precisely the network split into two parts; each part has to reconfigure itself automatically. Each coach also carries a gateway that offers applications to the train company customers as shown in figure 1.6

Figure 1.6 The coach environment

The museum is the fourth wireless environment described here. It is very similar to the station environment except it offers different services and applications. Also, one is more likely to trust the people he meets in the museum than in the train station (regarding the personal information disclosed for instance), so may be the services offered could take advantage of this fact.

1.3 Mobile Ad hoc Network

Ad hoc networks are emerging as the next generation of networks and defined as a collection of mobile nodes forming a temporary (spontaneous) network without the aid of any centralized administration or standard support services. In Latin, *ad hoc* literally means "for this," further meaning "for this purpose only," and thus usually temporary [2]. An ad hoc network is usually thought of as a network with nodes that are relatively mobile compared to a wired network. Hence the topology of the network is much more dynamic and the changes are often unpredictable oppose to the Internet which is a wired network. This fact creates many challenging research issues, since the objectives of how routing should take place is often unclear because of the different resources like bandwidth, battery power and demands like latency. The routing protocols used in ordinary wired networks are not well suited for this kind of dynamic environment.

1.3.1 Mobile Ad hoc Networks Definitions

A clear picture of exactly what is meant by an ad hoc network is difficult to pinpoint. In today's scientific literature, the term is used in many different ways. There are many different definitions, which describe ad hoc networks, but we just present two of them. The first one is given by *Internet Engineering Task Force* (IETF) group and the other one

is given by Murphy [3]. After that, we describe an ad hoc network from a graph theoretical point of view.

1.3.1.1 IETF Definition of Ad hoc Network

A Mobile Ad hoc Network (MANET) is a self-configuring (autonomous) system of mobile routers (and associated hosts) connected by wireless links -the union of which form an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet operating as a hybrid fixed/ad hoc network.

This is in contrast to the well-known single hop cellular network model that supports the needs of wireless communication by installing base stations as access points. In these cellular networks, communication between two mobile nodes completely relies on the wired backbone and the fixed base stations. In a MANET, no such infrastructure exists and the network topology may dynamically change in an unpredictable manner since nodes are free to move as shown in Figure 1-7.

As for the mode of operation, ad hoc networks are basically peer-to-peer multi-hop mobile wireless networks where information packets are transmitted in a store-and-forward manner from a source to an arbitrary destination, via intermediate nodes.

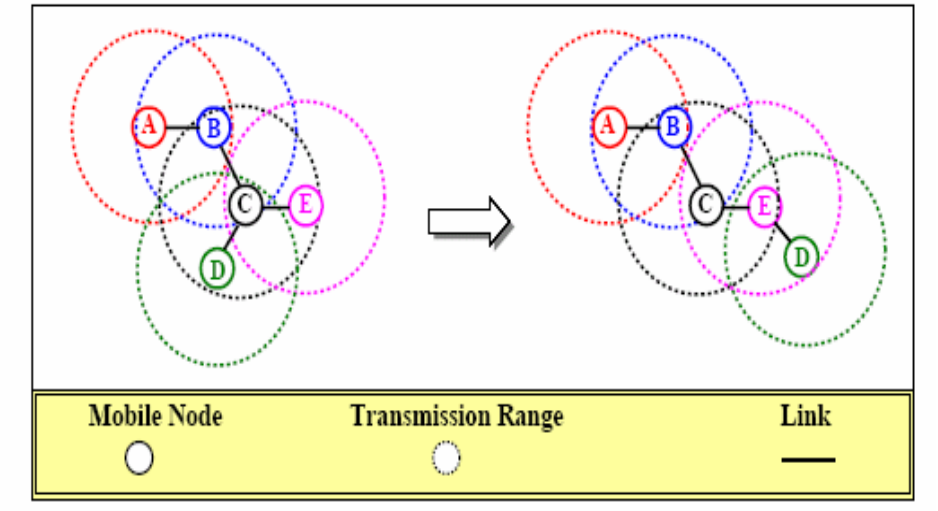

Figure 1.7 Dynamic topology in ad hoc networks

1.3.1.2 Murphy Definition of Ad hoc Network

According to [3], an ad hoc network is "a transitory association of mobile nodes which do not depend upon any fixed support infrastructure. Participants at a conference and disaster relief workers may find it necessary to interact with each other in this manner when the static support infrastructure is not available. An ad hoc network can be visualized as a continuously changing graph. Connection and disconnection is controlled by the distance among nodes and by willingness to collaborate in the formation of cohesive, albeit transitory community".

Distance among nodes, With ad hoc networks, no additional infrastructure is required beside the network nodes themselves. It is the distance among nodes, or rather their adjacency, that defines the boundaries of the network. That means, the only arranging of two or more mobile nodes within a certain boundary defines a new network in an ad hoc manner. Now, if the nodes were not mobile, an ad hoc network would not be different from LAN (Local Area Network). So, it is also the mobility of nodes, causing variations in their distance that gives such networks their ad hoc nature.

The actual boundary defining an ad hoc network really depends on the technology being used. Some ad hoc network solutions are limited to Personal Area Network (PAN). The Bluetooth technology, for instance, is allowed only for PAN (up to about 10 meters) [4].

Willingness to collaborate, The collocation of several nodes within a certain distance is a necessary but not sufficient condition to form an ad hoc network. In addition, collocated nodes need to be willing to collaborate. By definition of ad hoc networks, this willingness is expressed at the network level. The decision to collaborate or not is expressed by going online or offline.

Transitory peer-to-peer communities, Intuitively, the above features of ad hoc networks characterize their somehow "here and now" nature. That means, at any point in time (now), the network is defined by all nodes that are both within a certain distance and online (here). As consequence, nodes tend to appear and disappear in an ad hoc network much more often than in other types of networks, leading to so-called transitory community. People can join and withdraw from the community at any time.

1.3.2 Mobile Ad hoc Network Graph

A MANET topology can also be defined as a dynamic (arbitrary) multi-hop graph G = (N, L), where N is a finite set of mobile nodes MNs and L is a set of edges which represent wireless links. A link (i, j) □ L exists if *and* only if the distance between two mobile nodes is less or equal than a fixed radius r as shown in Figure 1-8. This r represents the radio transmission range that depends on wireless channel characteristics including transmission power. Accordingly, the neighborhood of a node x is defined by the set of nodes that are inside a circle (assume that MNs are moving in a two-dimensional plane) with center at x and radius r, and it is denoted by

$N_r(x) = N_x = \{n_j | d(x,n_j) \le r, x \ne n_j, \forall j \in N, j \le |N|\}$,where x is an arbitrary node in graph G and d is a distance function [5].

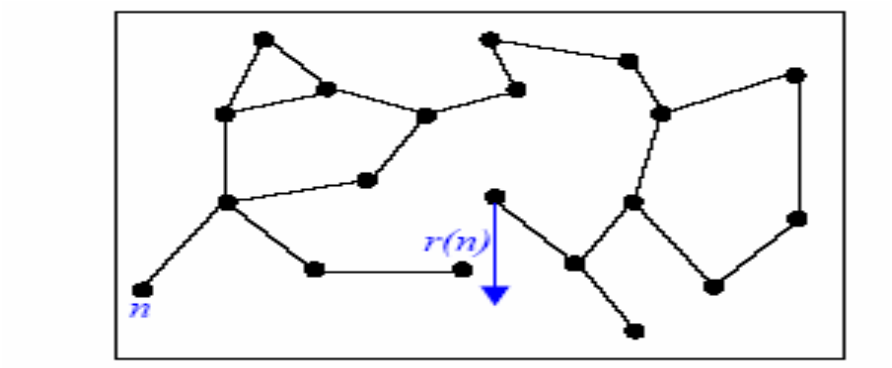

Figure 1.8 The graph of a wireless ad hoc network

A *path* (*route*) from node i to node j, denoted by Rij is a sequence of nodes Rij=(i,$n1$, $n2$,…,nk, j) where (i,$n1$), (nk,j) and (ny,ny+1) for 1≤ y ≤ k-1 are links. A *simple path* from i to j is a sequence of nodes with no node being repeated more than once. Due to the mobility of the nodes, the set of paths (wireless links) between any pair of nodes and distances is changing over time. New links can be established and existing links can vanish.

1.3.3 Example of An Ad hoc Network

Imagine a scenario as in *Tsunami disaster* relief operations wherein timely communication is a very important factor, the relief workers come in the area and without the need of any existing infrastructure, just switch on their handsets and start communicating with each other while moving and carrying out rescue work. In this case

of rescue problems, for example in scenes of natural disasters, an ad hoc network could be formed by communication devices in fire brigades, helicopters, ambulances, policies and also people with laptop computers or mobile phones in hospitals, pharmacies and so on, all together work in a collaborative way to provide effective solutions to the problem. Figure 1-9 shows an example of an ad hoc network which has different communication devices and some connection amongst them (When devices are in the same transmission range).

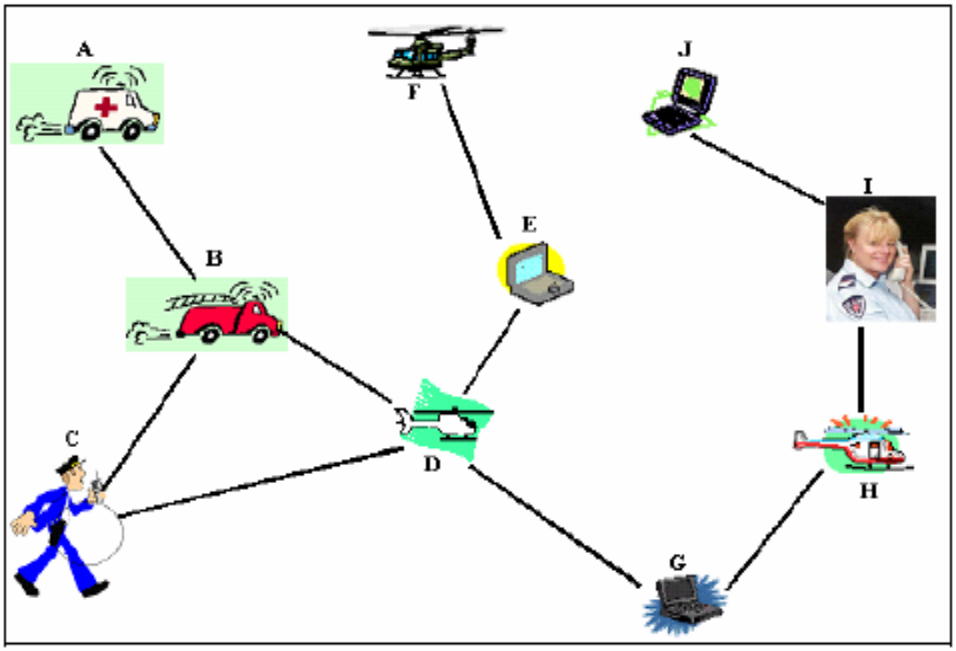

Figure 1.9 Representation of a mobile ad hoc network to solve rescue problems

It is obvious that they need to communicate with each other; however the high degree of mobility in this kind of network makes them change quickly. Some devices could be out of range with respect to others and therefore each device (node) must be able to act as a router to relay packets generated by other nodes. In Figure 1-10 when node *G* needs to communicate with node *I*, node *H* has to act as a router and transmit its information. But, what happen when node *H* is out of range with respect to node *I* ?. The possibility that node *E* is now in range of node *I* and the topology of the network has changed to different one may exist as it is shown in Figure 1-10. In this new topology, node *G* can communicate

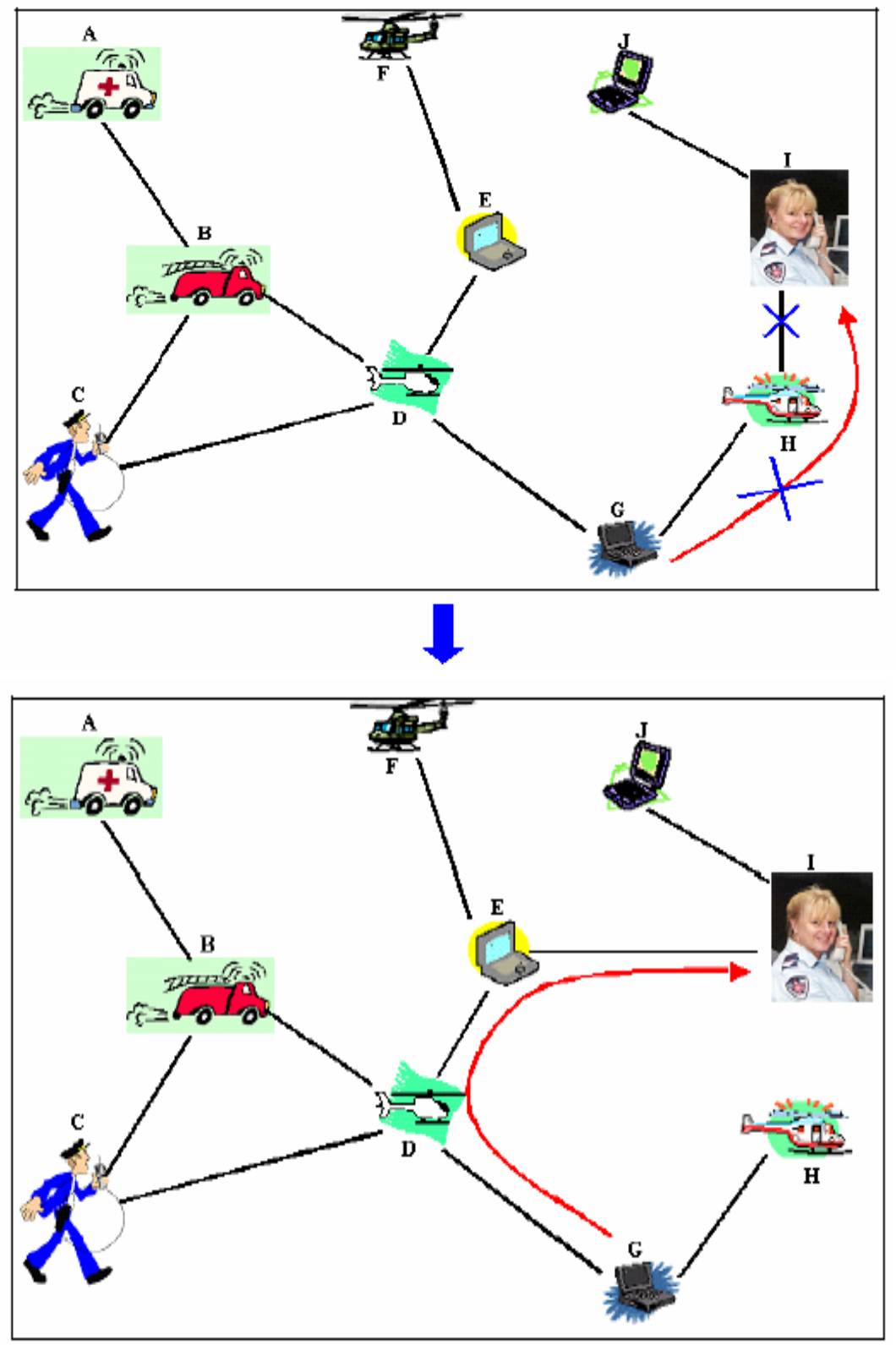

Figure 1.10 Representation of a new network topology in a case of rescue problems

Then, we can see how difficult the tasks of routing and maintaining paths in MANETs are and how much they are affected because of the node mobility, transmission signal, limitations on the battery power of mobile nodes and moreover limitations on the resources in terms of bandwidth of the wireless medium and the fact of sharing this medium (most commonly controlled by the IEEE 802.11 Medium Access Control (MAC) protocol).

1.4 Statement of the Problem

The goal set for the work, which is being presented in this dissertation, can be stated as follows:

1. To Identify various techniques used to classify routing protocols for wireless Ad Hoc Networks which will help in understanding of current protocols and designing of new protocols.
2. To perform the simulation of various routing protocol in wireless Ad-hoc network using Network Simulator (NS2) and also perform comparison of (DSDV , DSR and AODV routing protocols) with respect to Fraction of packets delivered, End-to-end delay and routing load for a given traffic and mobility model.

1.5 Organization of the Dissertation

The remaining chapters are organized as follows. Chapter 2 presents description of various routing protocols that will be compared while Chapter 3 presents various techniques used for classifying the routing protocols in wireless Ad - hoc network and Chapter 4 describes various mobility models used in our simulation. Chapter 5 gives brief description about Network Simulator (NS2) and our simulation methodology and the metrics for the performance analysis. Chapter 6 presents the simulation results and comparison of protocols and finally conclusions are presented in Chapter 7.

In a network, because of the fact that it may be necessary to hop several hops (multi hops) before a packet reaches the destination, a routing protocol is needed. The routing protocol has two main functions, selection of routes for various source-destination pairs and the delivery of messages to their correct destination .This chapter contains description of following routing protocols

2.1 Destination-Sequenced Distance-Vector Routing (DSDV)
2.2 Ad Hoc On-Demand Distance Vector Routing (AODV)
2.3 Dynamic Source Routing (DSR)
2.4 Temporally Ordered Routing Algorithm (TORA)
2.5 Zone Routing Protocol (ZRP)

2.1 ***Destination-Sequenced Distance-Vector Routing*** — The Destination-Sequenced Distance-Vector Routing protocol (DSDV) described in [6] is a table-driven algorithm based on the classical Bellman-Ford routing mechanism [7]. The improvements made to the Bellman-Ford algorithm include freedom from loops in routing tables.

Every mobile node in the network maintains a routing table in which all of the possible destinations within the net-work and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. To help alleviate the potentially large amount of network traffic that such updates can generate, route updates can employ two possible types of packets. The first is known as a *full dump*. This type of packet carries all available routing information and can require multiple network protocol data units (NPDUs). During periods of occasional movement, these packets are transmitted infrequently. Smaller *incremental* packets are used to relay only that information which has changed since the last full dump. Each of these

broadcasts should fit into a standard-size NPDU, thereby decreasing the amount of traffic generated. The mobile nodes maintain an additional table where they store the data sent in the incremental routing information packets.

New route broadcasts contain the address of the destination, the number of hops to reach the destination, the sequence number of the information received regarding the destination, as well as a new sequence number unique to the broadcast [6]. The route labeled with the most recent sequence number is always used. In the event that two updates have the same sequence number, the route with the smaller metric is used in order to optimize (shorten) the path. Mobiles also keep track of the settling time of routes, or the weighted average time that routes to a destination will fluctuate before the route with the best metric is received (see [6]). By delaying the broadcast of a routing update by the length of the settling time, mobiles can reduce network traffic and optimize routes by eliminating those broadcasts that would occur if a better route was discovered in the very near future.

2.2 *Ad Hoc On-Demand Distance Vector Routing* — The Ad Hoc On-Demand Distance Vector (AODV) routing protocol described in [8] builds on the DSDV algorithm previously described. AODV is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes on a demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. The authors of AODV classify it as a *pure on-demand route acquisition* system, since nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges [8].

When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a *path discovery* process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a "fresh enough" route to the destination is located. Figure 2a illustrates the propagation of the broadcast RREQs across the network. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its own sequence number, as well as a

broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies an RREQ. Along with its own sequence number and the broadcast ID, the source node includes in the RREQ the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destination/intermediate node responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ (Fig.2.1b). As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the RREP came. These forward route entries indicate the active forward route. Associated with each route entry is a route timer which will cause the deletion of the entry if it is not used within the specified lifetime. Because the RREP is forwarded along the path established by the RREQ, AODV only supports the use of symmetric links.

Routes are maintained as follows. If a source node moves, it is able to reinitiate the route discovery protocol to find a new route to the destination. If a node along the route moves, its upstream neighbor notices the move and propagates a *link failure notification* message (an RREP with infinite metric) to each of its active upstream neighbors to inform them of the erasure of that part of the route [8]. These nodes in turn propagate the *link failure notification* to their upstream neighbors, and so on until the source node is reached. The source node may then choose to reinitiate route discovery for that destination if a route is still desired.

An additional aspect of the protocol is the use of *hello* messages, periodic local broadcasts by a node to inform each mobile node of other nodes in its neighborhood. Hello messages can be used to maintain the local connectivity of a node. However, the

use of hello messages is not required. Nodes listen for retransmission of data packets to ensure that the next hop is still within reach.

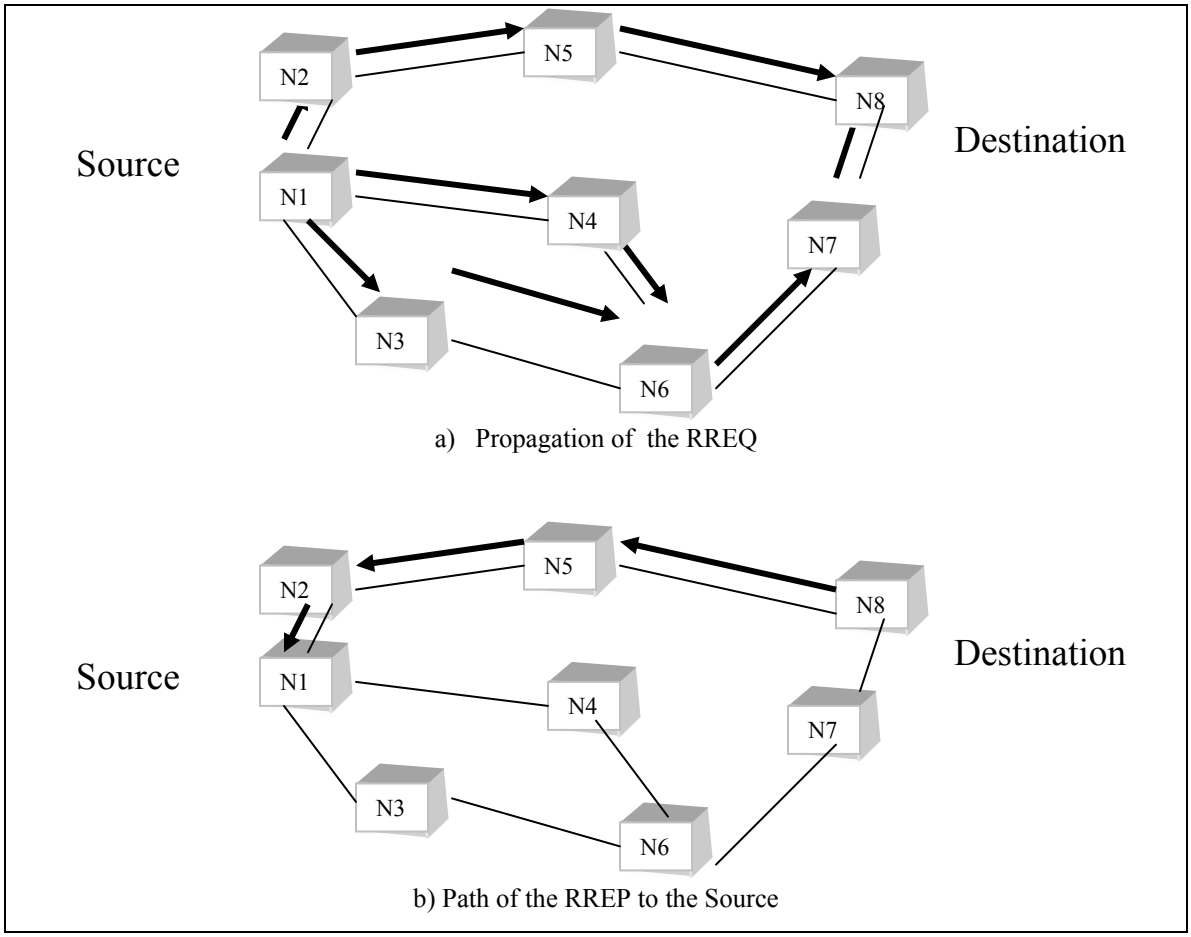

Figure 2.1. AODV route discovery

If such a retransmission is not heard, the node may use any one of a number of techniques, including the reception of hello messages, to determine whether the next hop is within communication range. The hello messages may list the other nodes from which a mobile has heard, thereby yielding greater knowledge of network connectivity.

2.3 ***Dynamic Source Routing*** — The Dynamic Source Routing (DSR) protocol presented in [9] is an on-demand routing protocol that is based on the concept of source routing. Mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learned.

The protocol consists of two major phases: route discovery and route maintenance. When a mobile node has a packet to send to some destination, it first consults its route cache to determine whether it already has a route to the destination. If it has an unexpired route to the destination, it will use this route to send the packet. On the other hand, if the node does not have such a route, it initiates route discovery by broadcasting a *route request* packet. This route request contains the address of the destination, along with the source node's address and a unique identification number. Each node receiving the packet checks whether it knows of a route to the destination. If it does not, it adds its own address to the *route record* of the packet and then forwards the packet along its outgoing links. To limit the number of route requests propagated on the outgoing links of a node, a mobile only forwards the route request if the request has not yet been seen by the mobile and if the mobile's address does not already appear in the route record.

A *route reply* is generated when the route request reaches either the destination itself, or an intermediate node which contains in its route cache an unexpired route to the destination [10]. By the time the packet reaches either the destination or such an intermediate node, it contains a route record yielding the sequence of hops taken. Figure 2.2a illustrates the formation of the route record as the route request propagates through the network. If the node generating the route reply is the destination, it places the route record contained in the route request into the route reply. If the responding node is an intermediate node, it will append its cached route to the route record and then generate the route reply. To return the route reply, the responding node must have a route to the initiator. If it has a route to the initiator in its route cache, it may use that route. Otherwise, if symmetric links are supported, the node may reverse the route in the route record. If symmetric links are not supported, the node may initiate its own route discovery and piggyback the route reply on the new route request. Figure 2.2b shows the transmission of the route reply with its associated route record back to the source node.

Route maintenance is accomplished through the use of route error packets and acknowledgments. *Route error* packets are generated at a node when the data link layer encounters a fatal transmission problem. When a route error packet is received, the hop in error is removed from the node's route cache and all routes containing the hop are truncated at that point. In addition to route error messages, acknowledgments are used to

verify the correct operation of the route links. Such acknowledgments include passive acknowledgments, where a mobile is able to hear the next hop forwarding the packet along the route.

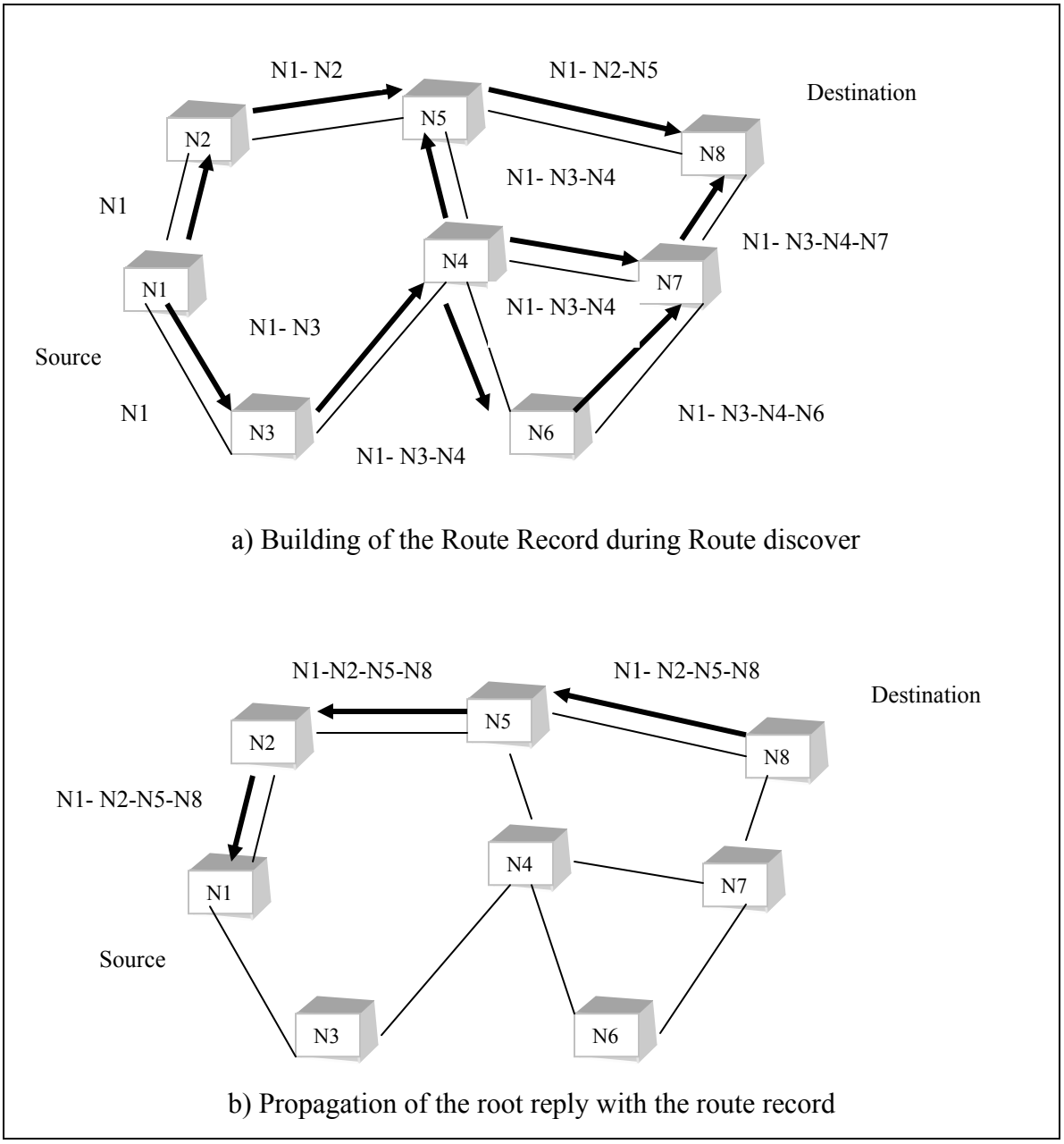

Figure 2.2. Creation of the route record in DSR.

2.4 ***Temporally Ordered Routing Algorithm*** — The Temporally Ordered Routing Algorithm (TORA) is a highly adaptive loop-free distributed routing algorithm based on the concept of link reversal [11]. TORA is proposed to operate in a highly dynamic

mobile networking environment. It is source-initiated and provides multiple routes for any desired source/destination pair. The key design concept of TORA is the localization of control messages to a very small set of nodes near the occurrence of a topological change. To accomplish this, nodes need to maintain routing information about adjacent (one-hop) nodes. The protocol performs three basic functions :

- Route creation
- Route maintenance
- Route erasure

During the route creation and maintenance phases, nodes use a "height" metric to establish a directed acyclic graph (DAG) rooted at the destination. Thereafter, links are assigned a direction (upstream or downstream) based on the relative height metric of neighboring nodes, as shown in Fig. 2.3a. This process of establishing a DAG is similar to the query/reply process proposed in Lightweight Mobile Routing (LMR) [12]. In times of node mobility the DAG route is broken, and route maintenance is necessary to reestablish a DAG rooted at the same destination. As shown in Fig. 2.3b, upon failure of the last downstream link, a node generates a new reference level which results in the propagation of that reference level by neighboring nodes, effectively coordinating a structured reaction to the failure. Links are reversed to reflect the change in adapting to the new reference level. This has the same effect as reversing the direction of one or more links when a node has no downstream links.

Timing is an important factor for TORA because the "height" metric is dependent on the logical time of a link failure; TORA assumes that all nodes have synchronized clocks TORA's metric is a quintuple comprising five elements, namely:

- Logical time of a link failure
- The unique ID of the node that defined the new reference level
- A reflection indicator bit
- A propagation ordering parameter
- The unique ID of the node

The first three elements collectively represent the reference level. A new reference level is defined each time a node loses its last downstream link due to a link failure.

TORA's route erasure phase essentially involves flooding a broadcast *clear packet* (CLR)

throughout the network to erase invalid routes.

In TORA there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other. Because TORA uses inter nodal coordination, its instability problem is similar to the "count-to-infinity" problem in distance-vector routing protocols, except that such oscillations are temporary and route convergence will ultimately occur.

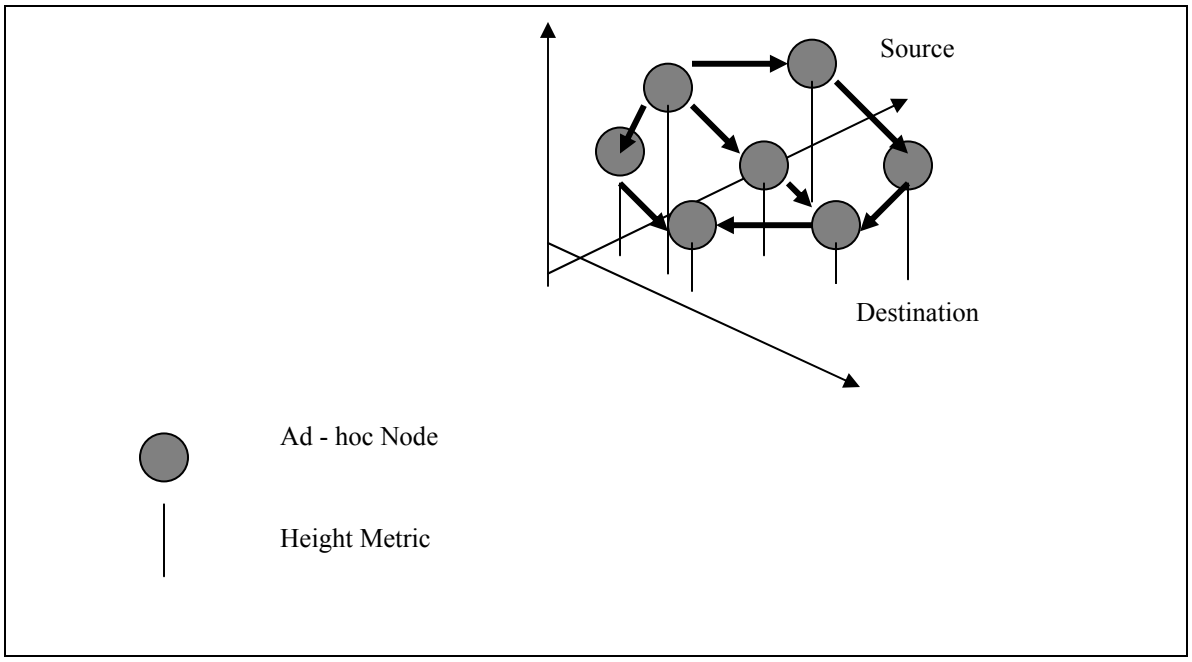

Figure 2.3. a) Route creation (showing link direction assignment)

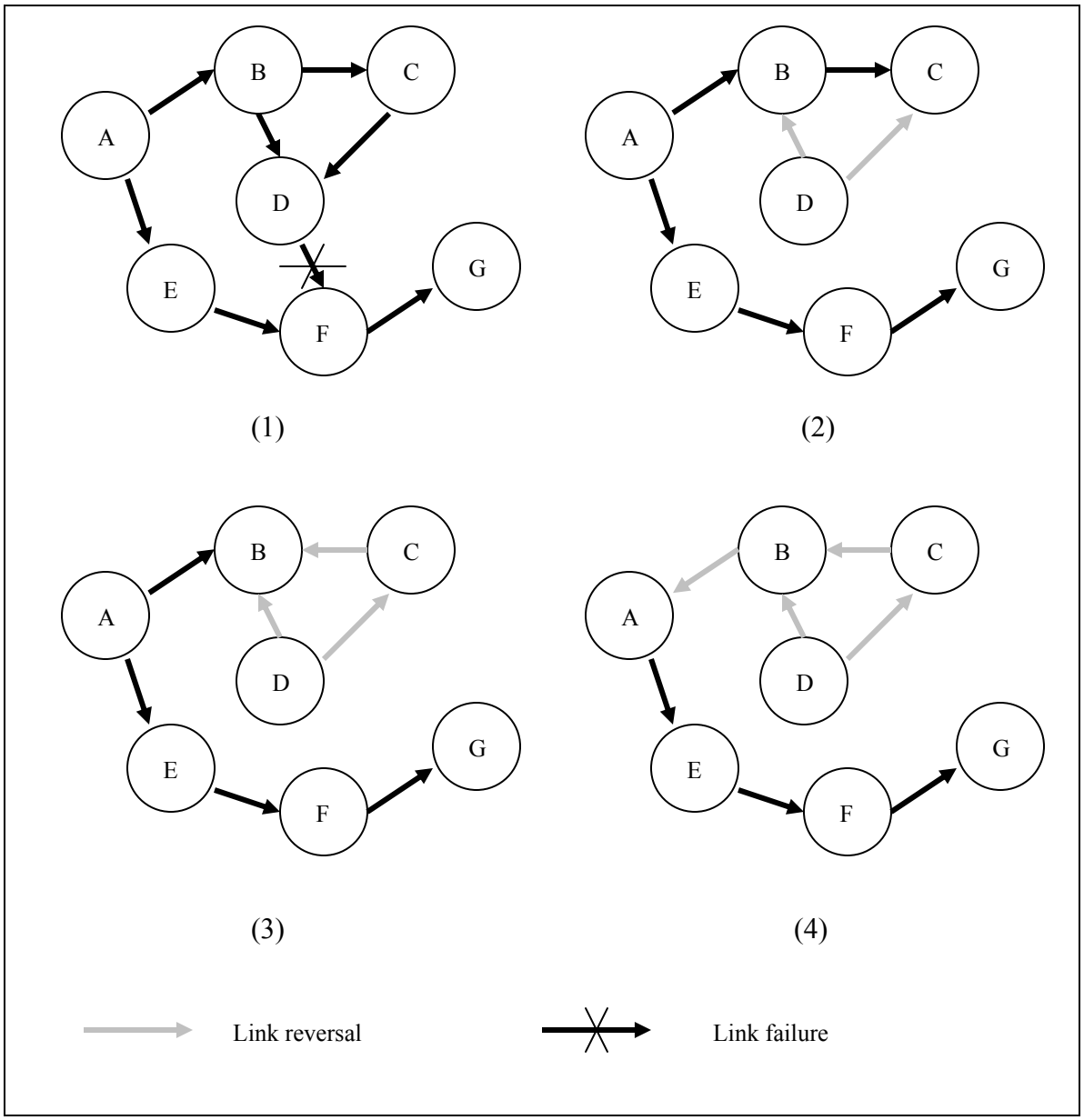

Figure 2.3. b) route maintenance (showing the link reversal phenomenon) in TORA.

2.5 ***Zone Routing Protocol*** — Zone Routing Protocol (ZRP) is a hybrid of a reactive and a proactive routing protocol. It divides the network into several routing zones and specifies two totally detached protocols that operate inside and between the routing zones. A zone is a local region defined by a single parameter called the zone radius, which is measured in hops. Nodes proactively maintain routing information for nodes within their zones and reactively discover routes for nodes outside their zones. The two

routing mechanisms are referred as the Intrazone Routing Protocol (IARP) and Interzone Routing Protocol (IERP) respectively.

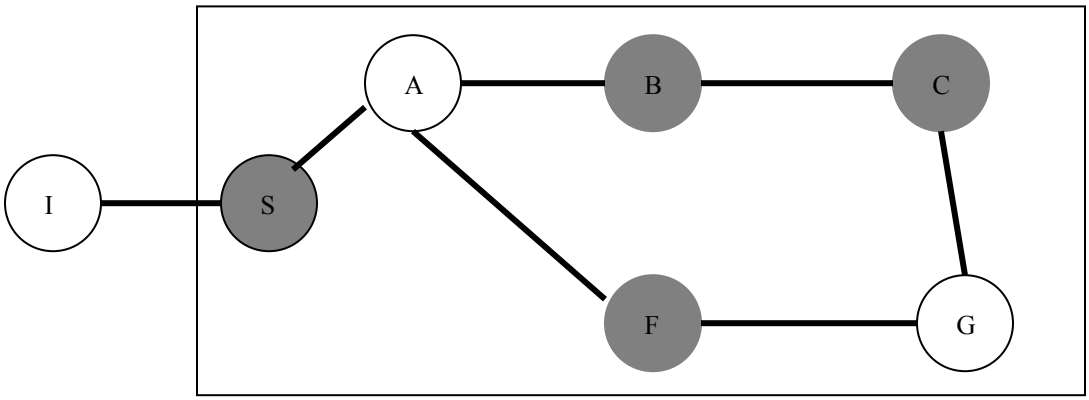

Figure 2.4 Zone of node F of radius 2

Figure 2.4 – illustrates the zone of node F where the zone radius is two. The only node outside of F's zone is I. Although one path to it is of length three. C is in the zone because there is a two-hop path from F.

Peripheral nodes refer to nodes whose minimum distance from the node in question is the zone radius. In Figure 2.4: S, B and C are F's peripheral nodes. Nodes use their peripheral nodes where issuing a network wide search for a route. By targeting queries to peripheral nodes, route requests tend to diverge away from the source, thereby reducing redundancy.

The strategy of using peripheral nodes for reactive route discoveries is called *bordercasting.* As the zone radius approaches the radius of the network, ZRP behaves like a proactive routing protocol. At the other extreme, where the zone radius is one, ZRP behaves like a reactive protocol. When the zone radius lies in between these two extremes, ZRP leverages the benefits of each to provide a robust, efficient, hybrid protocol.

- **Intrazone Routing Protocol (IARP)**

IARP is the proactive component of ZRP. It operates inside the routing zone and learns the minimum distance and routes to all the nodes within zone. The protocol is not defined and can include any number of proactive protocols, such as Distance Vector or link-state

routing. Different zones may operate with different intrazone protocols as long as the protocols are restricted to those zones. A change in topology means that update information only propagates within the affected routing zones as opposed to affecting the entire network.

- **Interzone Routing Protocol (IERP)**

IERP is the reactive part and is used for finding routes between different routing zones. IERP can be implemented as a reactive, distance vector algorithm, where nodes cache route information and packet routing is performed on a hop-by-hop basis. Alternatively, it can require no caching, in which case it relies on source routing. This is useful if the destination node does not lie within the routing zone.

When a node needs to send packets to a destination, it first checks to see if the destination is in the same zone. If so, the path to the destination is known (through IARP) and delivers them according. If the destination is not within the source's routing zone, the source bordercasts a route query to all of its peripheral nodes, which in turn forwards the request if the destination node is not found within their routing zone. This procedure is repeated until the request node is found and a route reply is sent back to the source indicating the route. IERP uses a Bordercast Resolution Protocol (BRP) that is included in ZRP, BRP provides bordercasting services, which do not exist in IP. Bordercasting is the process of sending IP datagrams from one node to all its peripheral nodes. BRP keeps track of the peripheral nodes and resolves a bordercast address to the individual IP addresses of the peripheral nodes. The message that was bordercasted is then encapsulated into BRP packet and sent to each peripheral node.

An example of this route discovery procedure is shown in Figure 2.5. Nodes A, F, B, C, G and H all are in zone F. Even though node B also has a distance of 3 from node F, it is included in the zone since the shortest distance is only 2 hops. In Figure 2.5 nodes S, B, C and H are border nodes to F. Node S want to send a packet to node D. since node d is not in the routing zone of S, a route request is sent to the border node B and F. Each border node checks to see if D is in their routing zone. Neither B nor F finds the requested node in their routing zone; thus the request is forwarded to the respectively border nodes. F sends the request to S, B, C and H while B sends the request to S, F, E

and G. Now the requested node D is found within the routing zone of both C and E thus a reply is generated and sent back towards the source node S.

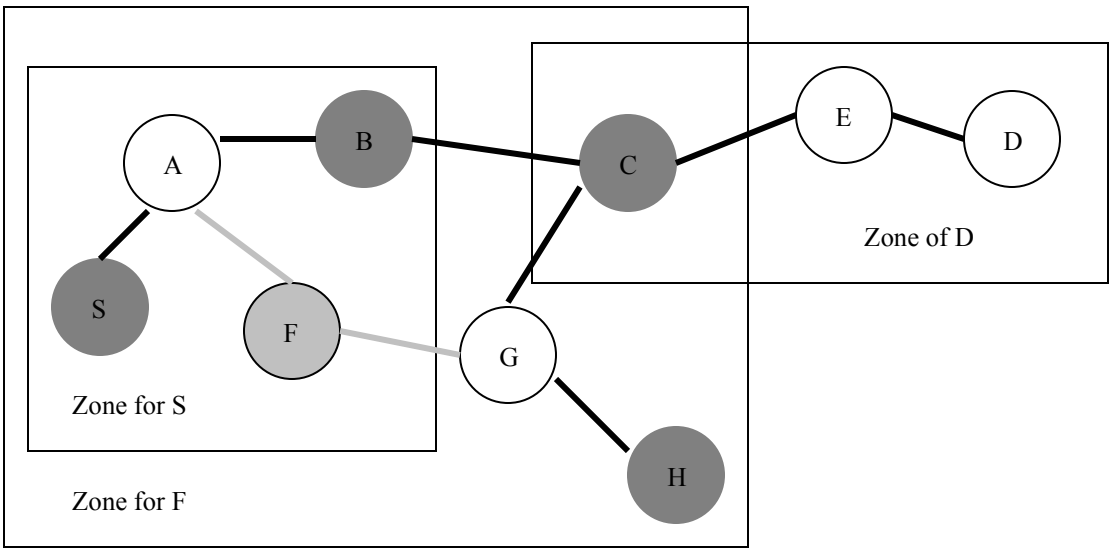

Figure 2.5 Network using ZRP.

The three different colors shows different zones F, S and D respectively.

IERP must maintain route information during the propagation of route requests in order that the discovering node may transmit the route reply message back to the source node. The route reply message must also contain the newly discovered route fro the source to the destination. Each node may place its IP address into the header of the route request packet to the original source. If nodes do not cache any route information, strict source routing must be used. If some route information is cached, then loose source may be used. If all nodes cache route information, then next-hop routing may be used.

To reduce redundant route request propagation through zones that have already been queried, ZRP may incorporate a query detection and early termination mechanism. In such a scheme, nodes keep track of the queries they hear on the network. They detect the queries either by relaying the to peripheral nodes or by operating in promiscuous mode. The query source IP address and unique identifier (i.e. sequence number) are recorded in a Detected Queries table. If a redundant query enters a region, then the detecting node may drop the packet and discontinue its propagation in that area. ZRP offers two distinct

methods of query detection for redundant queries and to reduce overhead. Query Detection 1 (QDI) allows the intermediate nodes to detect a redundant query and terminate the thread. Query Detection 2 (QD2) allows all nodes to detect a redundant query and terminate the request.

Figure 2.6 illustrates both levels of advanced query detection. In this example, node S broadcasts to two peripheral nodes B and D. The intermediate nodes A and C can detect passing route request packets and records that S's routing zone has been queried. In single channel networks, node E may also be able to receive A's transmission and record the query information as well.

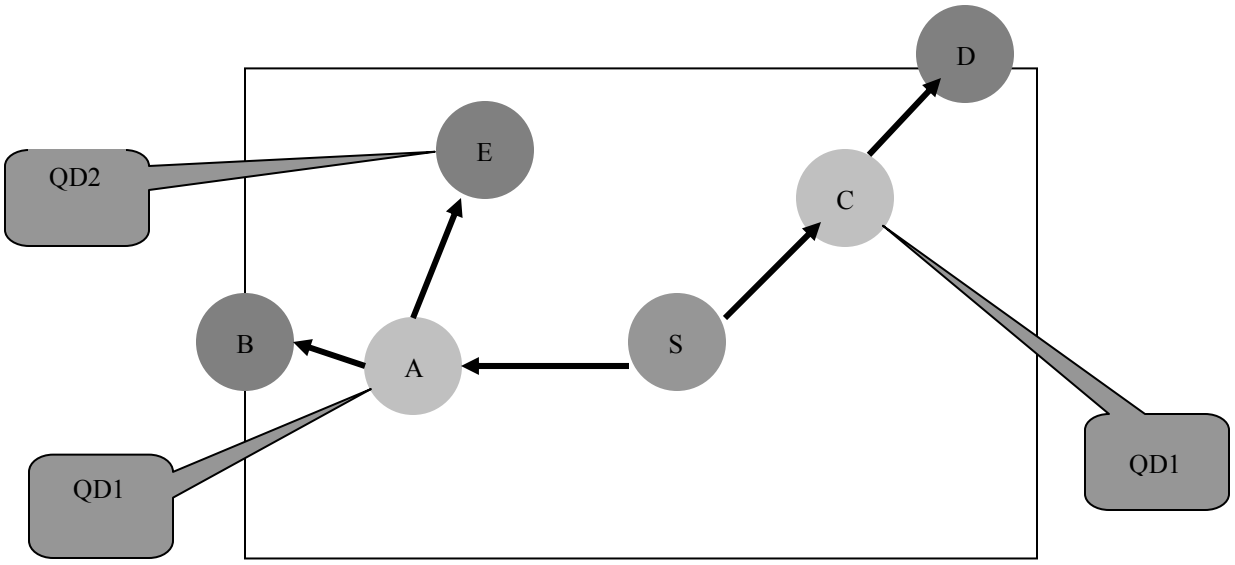

Figure 2.6 Advanced Query Detection (QD1/QD2)

A node will not relay a query packet to a targeted bordercast recipient either if that recipient lies inside the routing zone of a previously bordercast node or if this node has already relayed the query to this recipient. This scheme, which we refer to as early termination (ET), relies on query detection to identify which local nodes have already bordercast the query. To identify the nodes that lie within the routing zone of these bordercast nodes, the topology of an extended zone of radius 2ρ-1 hops (where ρ is the radius of the "basic" routing zone) must be maintained. Conveniently. IARP already maintains this extended information in support of multicast based border casting.

Figure 2.7 illustrates the operation of ET. Node B first detects (and relays) a route request packet broadcast by node T. Later, B receives a route request packet to be relayed

to the broadcast recipient node C, B recognizes that node C belongs to the previously queried routing zone of node T and therefore withholds transmission.

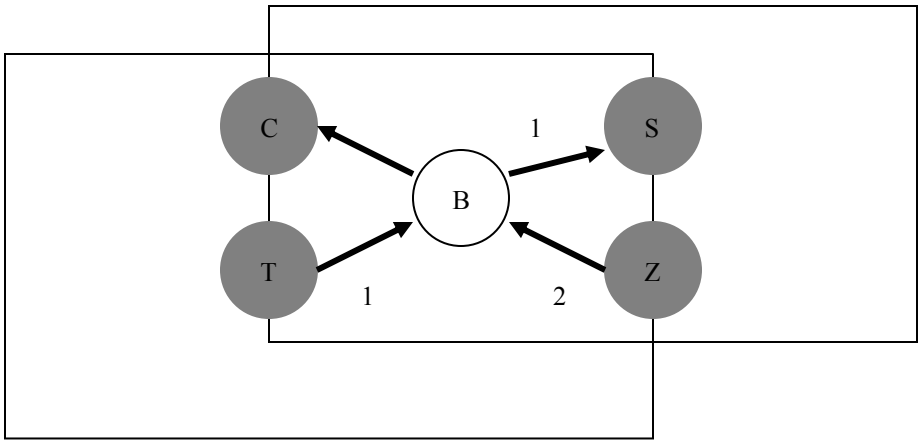

Figure 2.7 Early Termination

IERP also provides route maintenance. Route failures can be detected and reported proactively by IARP when a node leaves a zone. They can also be reported by IP when the next-hop for a datagram is determined to be unreachable. Nodes detecting a route failure may choose to notify the source and / or attempt to repair the route.

There are different criteria for designing and classifying routing protocols for wireless ad hoc networks. Classification is done on the basis of the network types. In the following, ad hoc networks are classified according to three different aspects.

3.1 Classification According to Communication

This simple classification is based on the formation and type of communication.

3.1.1 Single-hop Ad hoc Network

Nodes are in their reachable area and can communicate directly as shown in Figure 3-1. Single-hop ad hoc networks are the simplest type of ad hoc networks where all nodes are in their mutual range, that means the individual nodes can communicate directly together, without any help of other intermediate nodes. One could call this type of networks also plug and play networks, since it concerns mainly the simple and fast structure from temporary connections.

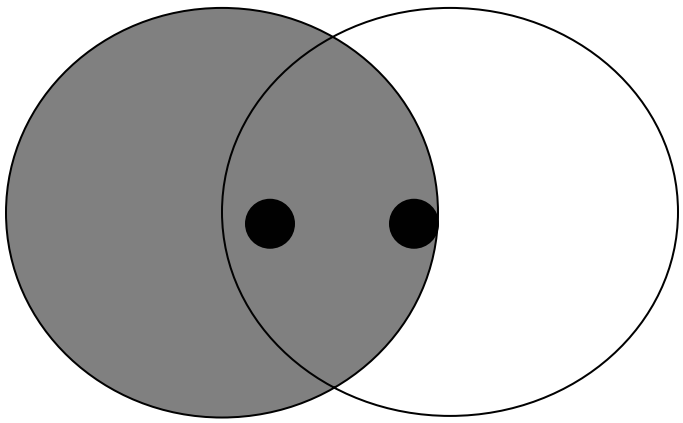

Figure 3.1 Single-hop ad hoc network

With this class, the mobility does not play a role. The individual nodes do not have to be static, they must remain however within the range of all nodes, that means the entire network could move as group, which would not modify in the communication relations anything.

3.1.2 Multi-hop Ad hoc Network

This class in the literature is the most examined type of ad hoc networks. It differs from the first class in that, some nodes are far and cannot communicate directly. Therefore, the traffic of these communication end-points has to be forwarded by other intermediate nodes. Figure 3-2 shows the communication path of far nodes as black lines. With this class also, one assumes that the nodes are mobile. The basic difficulty of the networks of this class is the node mobility, whereby the network topology is subjected to continuous modifications

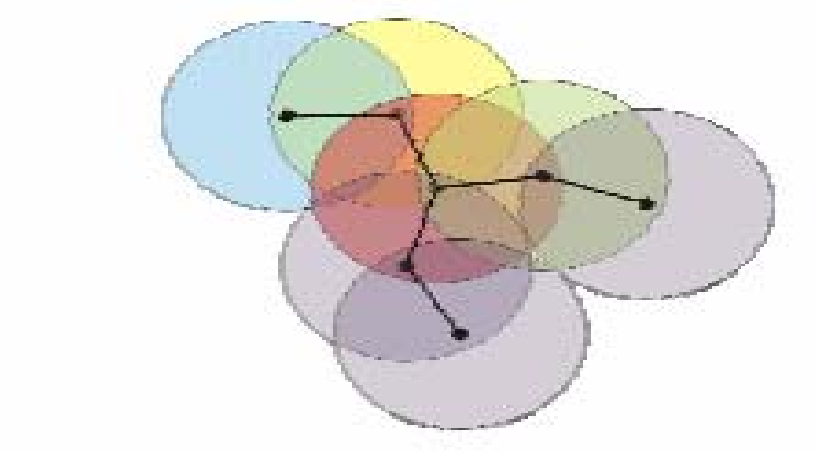

Figure 3.2 Multi-hop ad hoc network

The general problem in networks of this class is the assignment of a routing protocol. High performance routing protocols must be adaptive to the fast topology modification.

3.2 Classification According to Topology

Ad hoc networks can be classified according to the network topology. The individual nodes in an ad hoc network are divided into different types with special functions. There are three different classes: *flat*, *hierarchical* and a*ggregate ad hoc networks*.

3.2.1 Flat Ad hoc Networks

In *flat ad hoc networks*, all nodes carry the same responsibility and there is no distinction between the individual nodes as indicated in Figure 3-3. All nodes are equivalent and can transfer all functions in the ad hoc network.

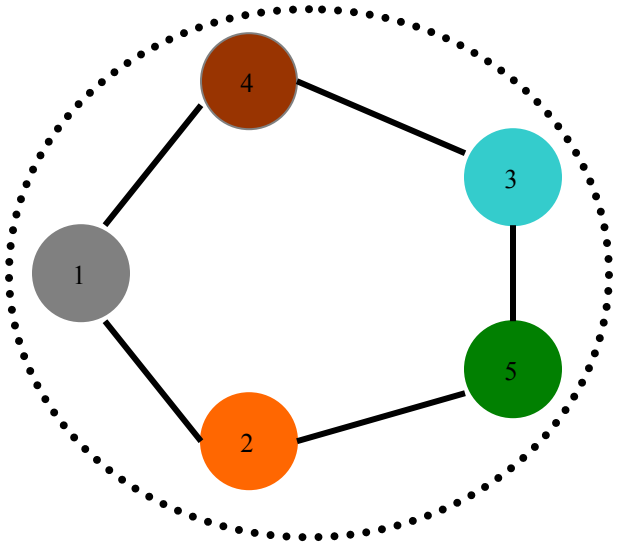

Figure 3.3 Flat ad hoc network

Control messages have to be transmitted globally throughout the network, but they are appropriate for highly dynamic network topology. The scalability decreases when the number of nodes increases significantly.

3.2.2 Hierarchical Ad hoc Networks

Hierarchical ad hoc networks consist in this case of several clusters, each one represents a network and all are linked together as indicated in Figure 3-4. The nodes in hierarchical ad hoc networks can be differentiated into two types:

- Master nodes: administer the cluster and are responsible for passing the data on to other cluster.
- Normal nodes: Communicate within the cluster directly together and with nodes in other clusters with the help of the master node. Normal nodes are called also *slave* nodes.

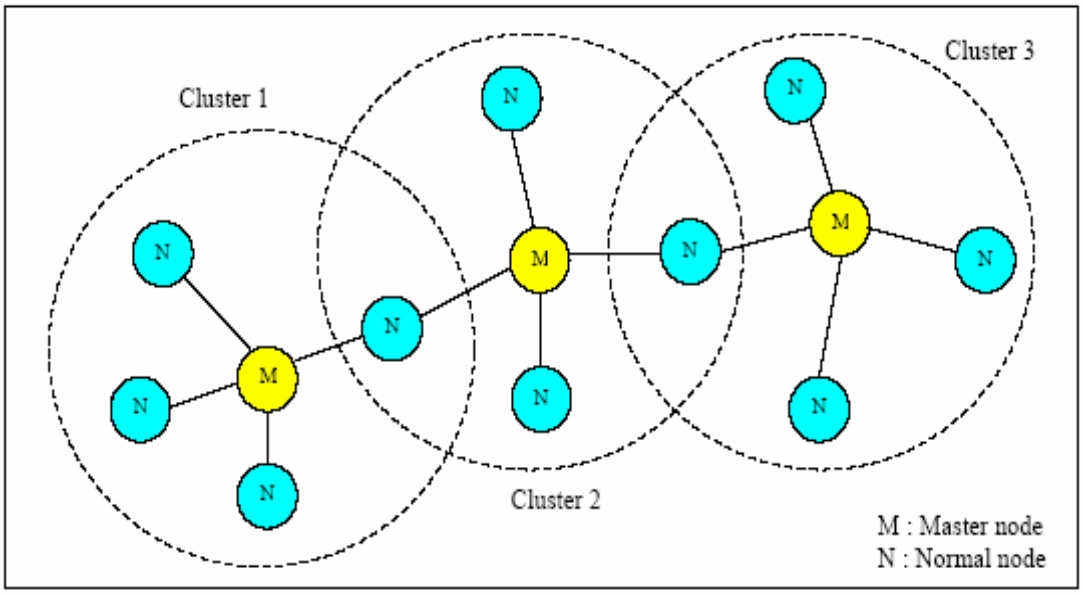

Figure 3.4 Hierarchical ad hoc network

One assumes, that the majority of communication (control messages) takes place within the cluster and only a fraction between different clusters. During communication within a cluster no forwarding of communication traffic is necessary. The master node is responsible for the switching of a connection between nodes in different clusters.

The flat architecture has the following advantages over the hierarchical:

- Increased reliability and survivability.
- No single point of failure.
- Alternative routes in the network.
- More optimal routing.
- Better coverage, i.e. reduced use of the wireless resources.
- Route diversity, i.e. better load balancing property.
- All nodes have one type of equipment.

The *no single point of failure* is of great importance for a message to reach its destination. This means that if one node goes down, the rest of the network will still function properly. In the hierarchical approach this is altogether another matter. If one of the cluster heads goes down, that section of the network won't be able to send or receive messages to other sections for the duration of the downtime of the cluster head.

Hierarchical architectures are more suitable for low mobility case. Although flat architectures are more flexible and simpler than hierarchical ones, hierarchical architectures provide a more scalable approach.

3.2.3 Aggregate Ad hoc Networks

Aggregate ad hoc networks bring together a set of nodes into zones. Therefore, the network is partitioned into a set of zones as shown in Figure 3-5. Each node belongs to two levels topology: low level (node level) topology and high level (zone level) topology. Also, each node may be characterized by two ID numbers: node ID number and zone ID number. Normally, aggregate architectures are related to the notion of *zone*. In aggregate architectures, we find both intra-zone and inter-zone architectures which in turn can either support flat or hierarchical architectures.

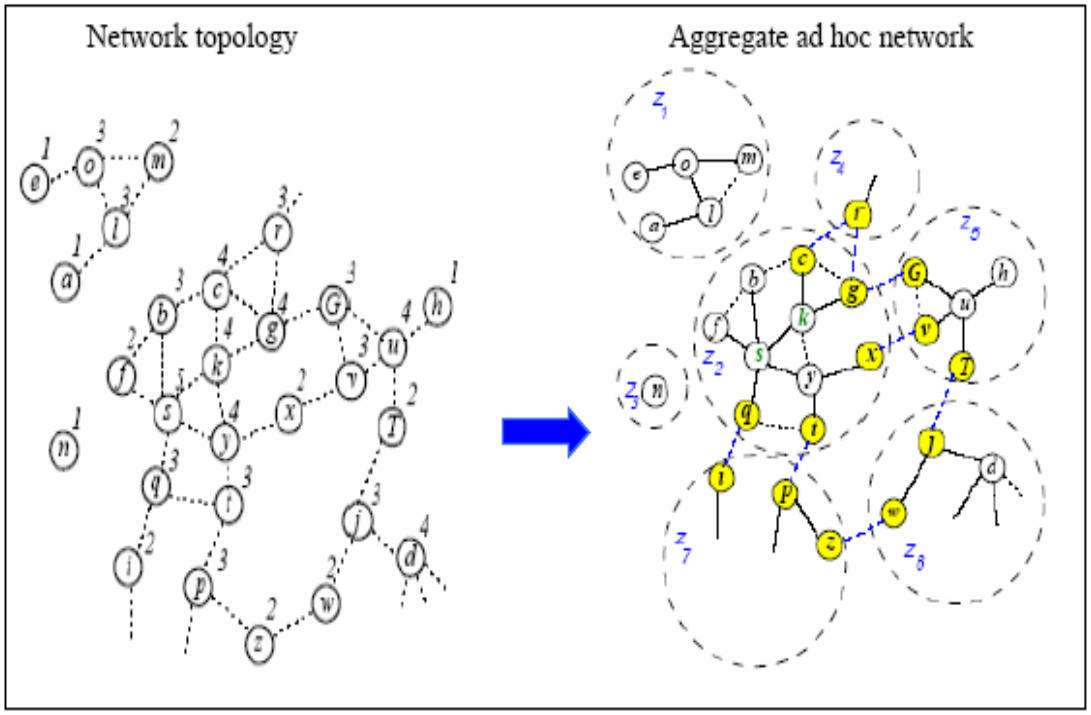

Figure 3.5 Aggregate network architecture

3.3 Classification According to Node Configuration

Further classification of ad hoc networks can be executed on the basis of the hardware configuration of the nodes [13]. The configuration of the nodes in a mobile ad hoc network is important and can depend very strongly on the actual application.

3.3.1 Homogeneous Ad hoc Networks

In homogeneous ad hoc networks, all nodes possess the same characteristics regarding the hardware configuration as processor, memory, display and peripheral devices. Most well known representatives of homogeneous ad hoc networks are wireless sensor networks. In homogeneous ad hoc networks, applications can proceed from certain prerequisites; for example, the localization is considerably facilitated by the presence of control components in each node.

3.3.2 Heterogeneous Ad hoc Networks

In heterogeneous ad hoc networks, the nodes differ according to the hardware configuration. Each node has different characteristics, resources and policies. In ad hoc networks of this class, all nodes cannot provide the same services.

3.4 Routing Protocols Classification

Several routing protocols have been proposed for mobile ad hoc network regarding application requirements and network properties. The routing protocols for mobile ad hoc network are classified according to six criteria as indicated in Figure 3-7:

1. Routing Philosophy: *Table-driven* versus *on-demand* versus *hybrid approach*,

2. Routing Architecture: *Flat* versus *hierarchical* versus *aggregate architecture*,

3. Routing Information: *Global Position* versus *Global Position-Less based protocols*,

4. Routing Generation: *First generation* versus *second generation* versus *third generation*.

5. Routing Updates : Periodical updates versus event-driven update

6. Route Computation : Decentralized computation versus Distributed Computation

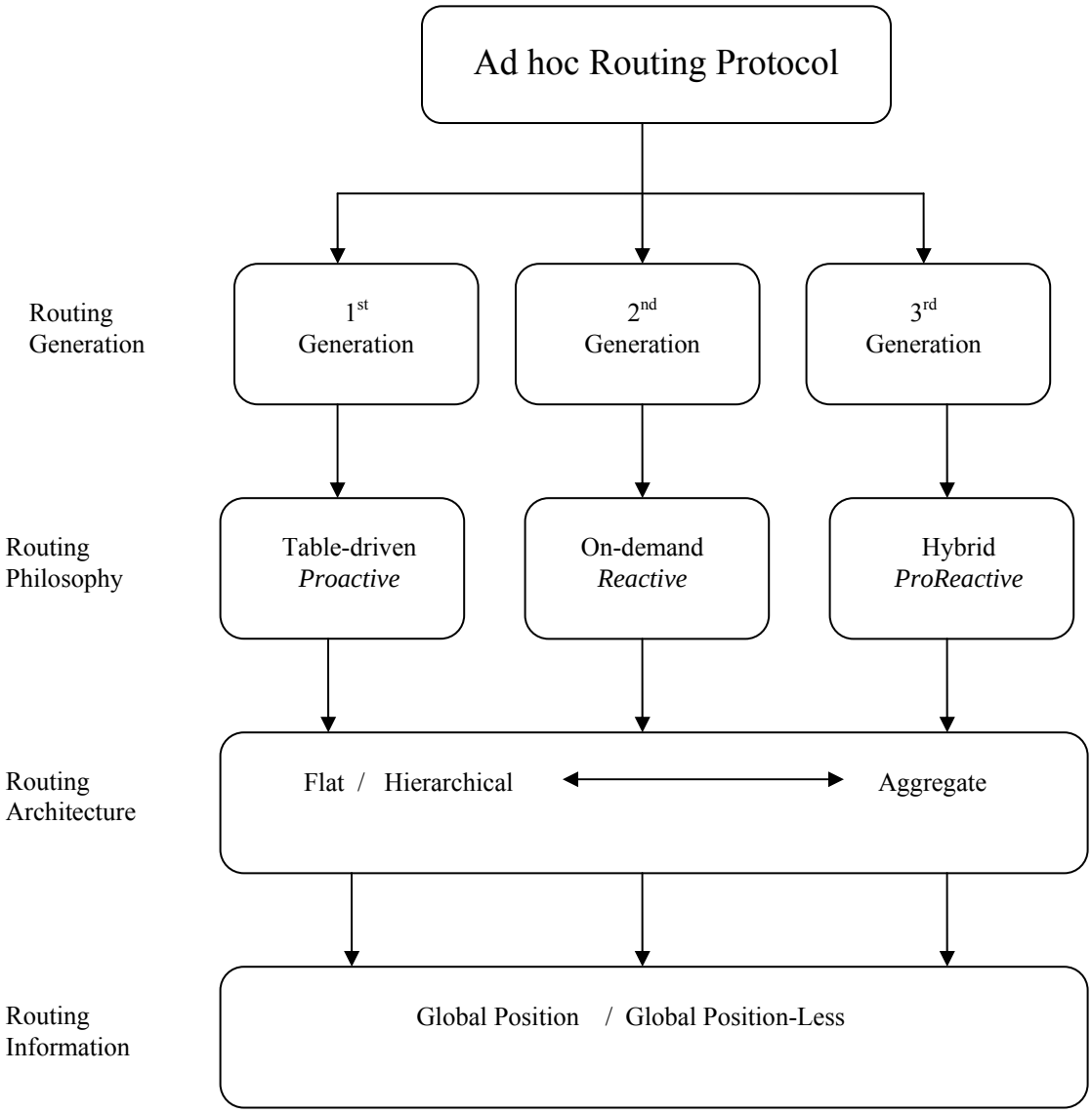

Figure 3.7 Classification of ad hoc network routing protocols

3.4.1 Routing Philosophy

- **Table Driven Routing Protocols (Proactive)**

  In proactive or table-driven routing protocols, each node continuously maintains up-to-date routes to every other node in the network. Routing information is periodically transmitted throughout the network in order to maintain routing table consistency.

Thus, if a route has already existed before traffic arrives, transmission occurs without delay. Otherwise, traffic packets should wait in queue until the node receives routing information corresponding to its destination. However, for highly dynamic network topology, the proactive schemes require a significant amount of resources to keep routing information up-to-date and reliable.

Proactive protocols suffer the disadvantage of additional control traffic that is needed to continually update stale route entries. Since the network topology is dynamic, when a link goes down, all paths that use that link are broken and have to be repaired. If no application is using these paths, then the effort gone in to repair may be considered wasted. This wasted effort can cause scarce bandwidth resources to be wasted and can cause further congestion at intermediate network points. Proactive protocols are scalable in the number of flows and the number of nodes but are not scalable in the frequency of topology change. Thus, this strategy is appropriate for a network with low mobility.

Certain proactive routing protocols are Destination-Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR) and Clusterhead Gateway Switch Routing (CGSR).

- **On-Demand Routing Protocols (Reactive)**

  In contrast to proactive approach, in reactive or on demand protocols, a node initiates a route discovery throughout the network, only when it wants to send packets to its destination. For this purpose, a node initiates a *route discovery* process through the network. This process is completed once a route is determined or all possible permutations have been examined. Once a route has been established, it is maintained by a *route maintenance* process until either the destination becomes inaccessible along every path from the source or until the route is no longer desired. In reactive schemes, nodes maintain the routes to active destinations. A route search is needed for every unknown destination. Therefore, theoretically the communication overhead is reduced at expense of delay due to route research. Furthermore, the rapidly changing topology may break an active route and cause subsequent route searches [14]. Reactive protocols may not be optimal in terms of bandwidth utilization because of flooding of the route discovery request, but they remain scalable in the frequency

of topology change. Such protocols are not scalable in the number of nodes, however, they can be made scalable if a hierarchical architecture is used. Further reactive protocols are not scalable in the number of flows. Thus, reactive strategies are suitable for networks with high mobility and relatively small number of flows.

Some reactive protocols are Cluster Based Routing Protocol (CBRP), Ad hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), Temporally Ordered Routing Algorithm (TORA), Associativity-Based Routing (ABR), Signal Stability Routing (SSR) and Location Aided Routing (LAR).

- **Hybrid Protocols**

  Finally in hybrid protocols, each node maintains both the topology information within its zone and the information regarding neighboring zones, that means, proactive behavior within a zone and reactive behavior among zones. Thus, a route to each destination within a zone is established without delay, while a route discovery and a route maintenance procedure is required for destinations that are in other zones.

  The zone routing protocol (ZRP), zone-based hierarchical link state (ZHLS) routing protocol and distributed dynamic routing algorithm (DDR) are three hybrid routing approaches. The hybrid protocols can provide a better trade-off between communication overhead and delay, but this trade-off is subjected to the size of a zone and the dynamics of a zone. Furthermore, hybrid approaches provide a compromise on scalability issue in relation to the frequency of end-to-end connection, the total number of nodes and the frequency of topology change. Thus, the hybrid approach is an appropriate candidate for routing in a large network.

3.4.2 Routing Architecture

- **Flat Architecture**

  In flat architecture, all nodes carry the same responsibility. Flat architectures do not optimize bandwidth resource utilization in large networks because control messages have to be transmitted globally throughout the network, but they are appropriate for highly dynamic network topology. The scalability decreases when the number of nodes increases significantly.

- **Hierarchical Architecture**

  On the contrary, in hierarchical architecture, aggregated nodes into clusters and clusters into super-clusters conceal the details of the network topology. Some nodes, such as clusterheads and gateway nodes have a higher computation communication load than other nodes. Hence, the mobility management becomes complex. The network reliability may also be affected due to single points of failure associated with the defined critical nodes. However, control messages may only have to be propagated within a cluster. Thus, the multilevel hierarchy reduces the storage requirement and the communication overhead of large wireless networks by providing a mechanism for localizing each node. In addition, hierarchical architectures are more suitable for low mobility case. Although flat architectures are more flexible and simpler than hierarchical one, hierarchical architectures provide more scalable approach.

- **Aggregate Architecture**

  Finally, aggregate architecture aggregates a set of nodes into zones. Therefore, the network is partitioned into a set of zones. Each node belongs to two levels topology: low level (node level) topology and high level (zone level) topology. Also, each node is characterized by two ID numbers: node ID number and zone ID number. Normally, aggregate architecture is related to the notion of zone. In aggregate architecture, we find both intra-zone and inter-zone architectures which in turn can either support flat or hierarchical architecture.

3.4.3 Routing Information

- **Global Position (GP) Based Protocols**

  In global position (GP) based protocols, the network relies on another system which can provide the physical information of the current position of MNs. Such physical locations can be obtained by using the Global Position System (GPS). This involves increasing in energy consumption, cost of the network maintenance and hardware requirements. Generally, satellites are used to deliver this physical information. Any problem in one of the used satellites will surely affect the efficiency of the network and in some cases can easily make this latter blocked.

- **Global Position-Less (GPL) Based Protocols**

  In global position-less (GPL) based protocols, the network is stand-alone in the sense that it operates independently of any infrastructure. However, there are some situations where the physical location remains useful such as emergency disaster relief after a hurricane or earthquake.

3.4.4 Routing Generation

- **First generation** of routing protocols in mobile ad hoc networks is ”*table-driven*” approaches which are mainly influenced by *Internet* routing protocols. They attempt to maintain consistent and up-to-date routing information from each node to every other node in the network.
- **Second generation** of routing protocols is ”*on-demand*” approaches that are designed in order to decrease high communication overhead in table-driven approach due to maintaining up-to-date routing information. While on-demand routing protocols decrease communication overhead, there are doubts about its scalability and delay. On-demand routing protocols are different in the way they construct and maintain a route to the destination and the metrics they use to differentiate the discovered routes, as well as the mechanism to avoid loop formation. Path finding differs from reactive to proactive approach in the sense that reactive approach applies an explicit route request that is followed by an explicit route reply while proactive approach uses implicit route reply.
- **Third generation** of routing protocols called *hybrid approach* which is introduced to provide a better compromise between communication overhead and delay as well as better scalability.

3.4.5 Routing Updates

Routing information needs to be disseminated to network nodes in order to ensure that the knowledge of link state and network topology remains up-to-date. Based on when the routing information will be disseminated, we can classify routing protocols as periodical update and event-driven update protocols.

- **Periodical Update** protocols disseminate routing information periodically. Periodical updates will simplify protocols and maintain network stability, and most importantly, enable (new) nodes to learn about topology and the state of network. However if the periods between updates is large, the protocol may not keep the information up-to-date. On the other hand, if the period is small, to many routing packets will be disseminated which consumes the precious bandwidth of a wireless network.

  - **Event-Driven Update ,** In this case, when events occur, (such as when a link fails or a new link appears), an update packet will be broadcast and the up-to-date status can be disseminated over the network soon. The problem might be that if the topology of networks changes rapidly, a lot of update packets will generated and disseminated over the network which will use a lot of precious bandwidth, and furthermore, may cause too much fluctuation of routes. One solution is to use some threshold.

  Periodical update and event-driven update mechanism can be used together, forming what is called a hybrid mechanism.

3.4.6 Routing Computation

Based on how (or where) a route is computed, there are two categories of routing protocols : decentralized computation and distributed computation.

- **Decentralized Computation**, In decentralized computation - based protocol, every node in the network maintains global and complete information about the network topology such that the node can compute the route to a destination itself when desired. The route computation in Link state routing is a typical example of decentralized computation.
- **Distributed Computation**, In distributed-based protocol, every node in the network only maintains partial and local information about the network topology. When a route needs to be compared, many nodes collaborate to compute the route. The route computation in Distance vector routing and the route discovery in on-demand routing belongs to this category.

Table 1: Comparison of Routing Protocols

Protocols	Route Computation	Structures	#Routes	Source Routing
DVR	Proactive/ Distributed	Flat	Single	No
DSDV	Proactive/ Distributed	Flat	Single	No
DSR	Reactive / Broadcast QUERY	Flat	Multiple	Yes
AODV	Reactive / Broadcast QUERY	Flat	Multiple	No
TORA	Reactive / Broadcast QUERY	Flat	Multiple (DAG)	No
ZRP	Proactive(intra)/ Reactive(inter)	Flat	Single/ Multiple	Yes for interzone
CSGR	Proactive/ Distributed	Hierarchy	Single	No
CEDAR	Proactive / Core Broadcast QUERY	Hierarchy	Single	Yes

Table 2: Comparison of Routing Protocols

Protocols	Stored Information	Update Period	Update Information	Update Destination	Method
DVR	Distance-vector	Periodical	Distance vector	Neighbors	Broadcast
DSDV	Distance-vector	Hybrid	Distance vector	Neighbors	Broadcast
DSR	Routes to desired Destination	Event-driven	Route-Error	Source	Unicast
AODV	Next hops for to desired Destination	Event-driven	Route-Error	Source	Unicast
TORA	Neighbor's heights	Event-driven	Nodes height	Neighbor's	Broadcast
ZRP	Local(within zone), topology	Periodical	Link state of nodes in the Zone	Neighbor's	Broadcast
CSGR	Clus. mem. Table, Dist. Vect.	Periodical	Clus. mem. Table, Dist. Vect.	Neigh.& Clus. head	Broadcast
CEDAR	Core/other nodes: global/local	Period. / Event -driven	Dynamic/ Stable link state	Neigh./Core nodes	Bro./Core bro.

To evaluate the performance of a protocol for an ad hoc network, it is necessary to test the protocol under realistic conditions, especially including the movement of the mobile nodes. A survey of different mobility models follows. This includes the Random Waypoint Model that is used during the simulation of protocols.

4.1 Random Walk Mobility Model

This model is based on random directions and speeds. By randomly choosing a direction between *0* and 2π and a speed between 0 and Vmax, the mobile node moves from its current position. A recalculation of speed and direction occurs after a given time or a given distance walked. The random walk mobility model is memory less. Future directions and speeds are independent of the past speeds and directions. This can cause unrealistic movement such as sharp turns or sudden stops. If the specified time or distance is short, the nodes are only walking on a very restricted area on the simulation area.

4.2 Random Waypoint Mobility Model

A mobile 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 mobile node waits again pause time seconds before choosing a new way point and speed.

The mobile 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.

The Random Waypoint Mobility Model is very widely used in simulation studies of MANET. As described in [16] the performance measures in mobile ad hoc networks are affected by the mobility model used. One of the most important parameters in mobile ad - hoc simulations is the nodal speed. The users want to adjust the average speed to be stabilized around a certain value and not to change over time. They also want to be able

to compare the performance of the mobile ad hoc routing protocols under different nodal speeds. For the Random Waypoint Mobility Model a common expectation is that the average is about half of the maximum, because the speeds in a Random Waypoint Model are chosen uniformly between 0 m/s and Vmax. But is this the average speed really reached in simulations? Not at all, the studies in [16] show that the average speed is decreasing over time and will approach 0. This could lead to wrong simulation results.

This phenomenon can be intuitively explained as follows. In the Random Waypoint Mobility Model a node selects its destination and its speed. The node keeps moving until it reaches its destination at that speed. If it selects a far destination and a low speed around 0 m/s, it travels for a long time with low speed. If it selects a speed near Vmax the time traveling with this high speed will be short. After a certain time the node has traveled much more time at low speed than at high speed. The average speed will approach 0 m/s. The suggestion in [16] to prevent this problem is choosing, e.g. 1 m/s instead of 0 m/s as Vmin. With this approach the average speed stabilizes after a certain at a value ½ * Vmax.

4.3 Random Direction Mobility Model

To reduce *density waves* in the average number of neighbors by the Random Waypoint Model the Random Direction Mobility Model was created. *Density waves* are the clustering of nodes in one part of the simulation area. For the Random Waypoint Mobility Model the probability of choosing a location near the center or a way point which requires traveling through the center of the area is high. The Random Direction Mobility Model was invented to prevent this behavior and to promote a semi constant number of neighbors. The mobile node selects a direction and travels to the border of the simulation area. If the boundary is reached, the node pauses for a specific time and then chooses a new direction and the process goes on. Because of pausing on the border of the area, the hop count for this mobility model is much higher than for most other mobility models.

The Network Simulator 2 (ns2) is a discrete event driven simulator developed at UC Berkeley [17, 18]. It is part of the VINT project. The goal of ns2 is to support networking research and education. It is suitable for designing new protocols, comparing different protocols and traffic evaluations. Ns2 is developed as a collaborative environment. It is distributed freely and open source. A large amount of institutes and people in development and research use, maintain and develop ns2. This increases the confidence in it. Versions are available for FreeBSD, Linux, Solaris, Windows and Mac OS X.

5.1 Structure of ns2

ns2 is built using object oriented methods in C++ and OTcl (object oriented variant of Tcl). As we

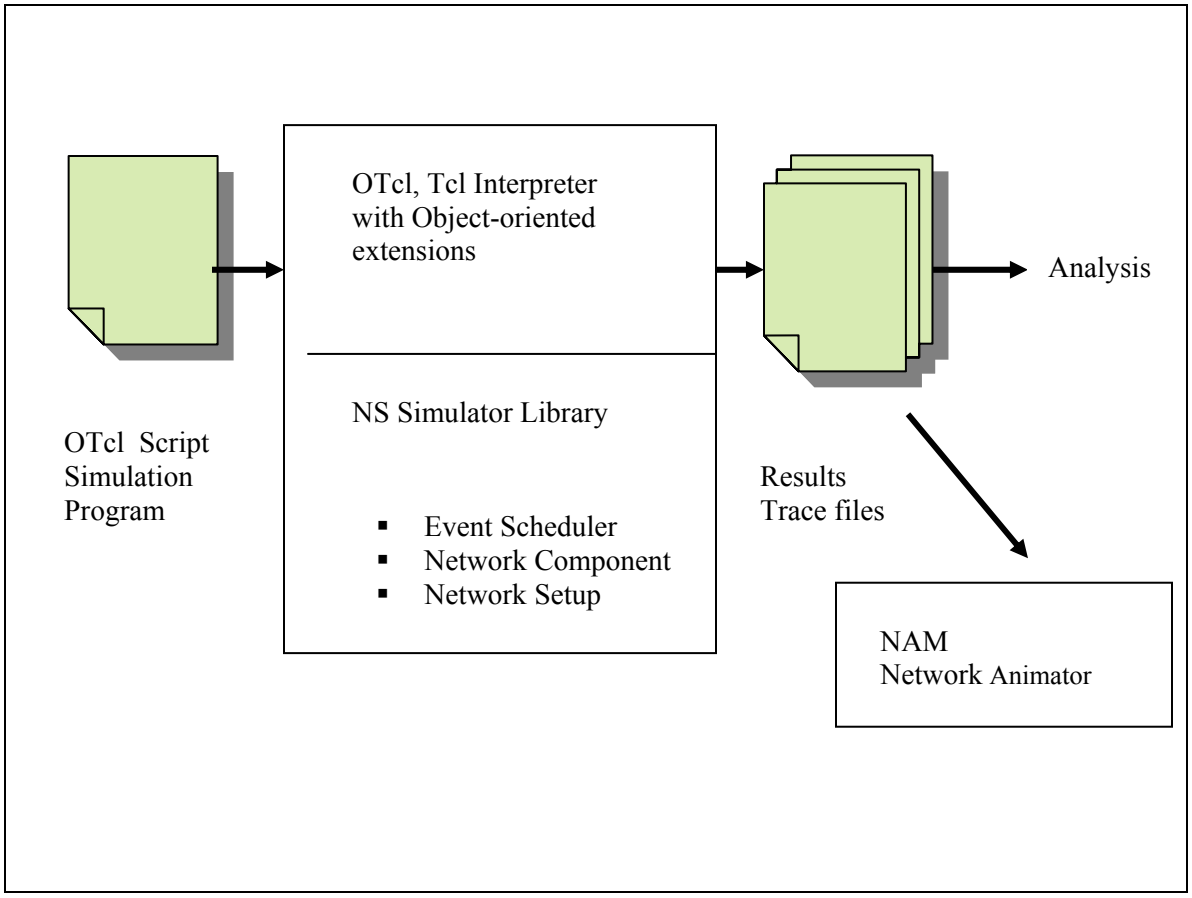

Figure 5.1 Simplified User's View of NS

can see in Fig. 5.1, ns2 interprets the simulation scripts written in OTcl. A user has to set the different components (e.g. event scheduler objects, network components libraries and setup module libraries) up in the simulation environment. The user writes his simulation as a OTcl script, plumbs the network components together to the complete simulation. If he needs new network components, he is free to implement them and to set them up in his simulation as well. The event scheduler as the other major component besides network components triggers the events of the simulation (e.g. sends packets, starts and stops tracing). Some parts of ns2 are written in C++ for efficiency reasons. The data path (written in C++) is separated from the control path (written in OTcl). Data path object are compiled and then made available to the OTcl interpreter through an OTcl linkage (tclcl) which maps methods and member variables of the C++ object to methods and variables of the linked OTcl object. The C++ objects are controlled by OTcl objects. It is possible to add methods and member variables to a C++ linked OTcl object. A linked class hierarchy in C++ has its corresponding class hierarchy in OTcl (fig. 5.2). Results obtained by ns 2 (trace files) have to be processed by other tools, e.g. the Network Animator (NAM).

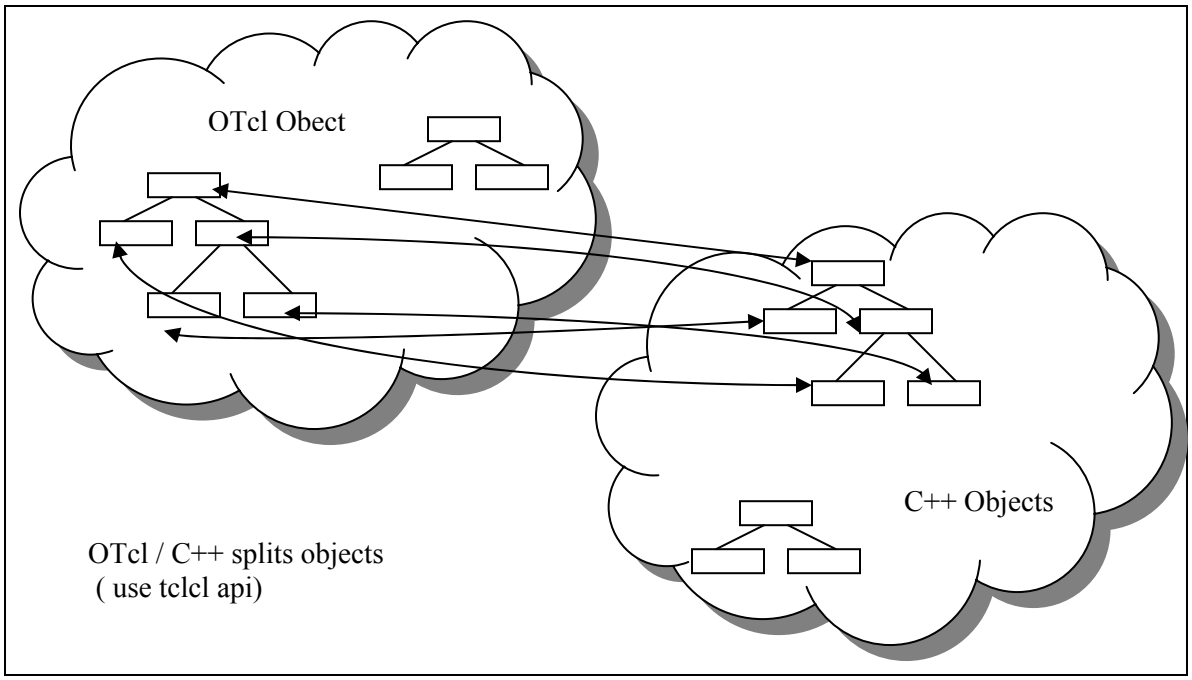

Figure 5.2 OTcl and C++: the duality

5.2 Functionalities of ns

Functionalities for wired, wireless networks, tracing, and visualization are available in ns2.

- **Support for the wired world include**
  - Routing DV, LS, PIM-SM
  - Transport protocols: TCP and UDP for unicast and SRM for multicast
  - Traffic sources: web, ftp, telnet, cbr (constant bit rate), stochastic, real audio
  - Different types of Queues: drop-tail, RED, FQ, SFQ, DRR
  - Quality of Service: Integrated Services and Differentiated Services
  - Emulation
- **Support for the wireless world include**
  - Ad hoc routing with different protocols, e.g. AODV, DSR, DSDV, TORA
  - Wired-cum-wireless networks
  - Mobile IP
  - Directed diffusion
  - Satellite
  - Sensor-MAC
  - Multiple propagation models (Free space, two-ray ground, shadowing)
  - Energy models
- **Tracing**
- **Visualisation**
  - Network Animator (NAM)
  - TraceGraph
- **Utilities**

In a recent work, the Monarch research group in CMU developed support for simulating multi-hop wireless networks complete with physical, data link and MAC layer models on *ns-2*. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer. The 802.11 DCF uses Request-to-send (RTS) and Clear-

to-send (CTS) control packets for "unicast" data transmission to a neighboring node. The RTS/CTS exchange precedes the data packet transmission and implements a form of *virtual carrier sensing* and channel reservation to reduce the impact of the well-known *hidden terminal problem*. Data packet transmission is followed by an ACK. "Broadcast" data packets and the RTS control packets are sent using physical carrier sensing. An unslotted CSMA technique with collision avoidance (CSMA/CA) is used to transmit these packets . The radio model uses characteristics similar to a commercial radio interface, Lucent's Wave LAN. Wave LAN is a shared-media radio with a nominal bit-rate of 2 Mb/sec and a nominal radio range of 250 meters

The routing protocol model "sees" all data packets transmitted or forwarded, and "responds" by invoking routing activities as appropriate. The RREQ packets are treated as broadcast packets in the MAC. RREP, RERR and data packets are all unicast packets with a specified neighbor as the MAC destination. Both protocols detect link breakage using feedback from the MAC layer. A signal is sent to the routing layer when the MAC layer fails to deliver a unicast packet to the next hop. This is indicated, for example, by failure to receive CTS after an RTS, or absence of an ACK following data transmission.

Both protocols maintain a *send buffer* of 64 packets. It buffers all data packets waiting for a route, e.g., packets for which route discovery has started, but no reply has arrived yet. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 sec. All packets (both data and routing) sent by the routing layer are queued at the *interface queue* until the MAC layer can transmit them. The interface queue is FIFO, with a maximum size of 64. Routing packets are given higher priority than data packets in the interface queue.

5.3 Performance Metrics

The following performance metrics are evaluated:

Packet delivery ratio The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

Average end-to-end delay This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times.

Routing overhead The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission. The first two metrics are the most important for best effort traffic.

5.4 Experiment Environment

Hardware Desktop PC: Pentium IV, 256MB Memory, Ethernet

Operating System Red hat 9.0, Linux kernel 2.6

Network Simulator ns-2, version 2.7 with CMU MANET extension.

5.5 Simulation Data Generation

Simulation data are created by related scenario and communication generation tools – *cbrgen.tcl* and *setdest*. *cbrgen.tcl* – this traffic generator script can be used to create CBR and TCP traffics connections between wireless mobile nodes. So the command line looks like the following:

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

For the simulations carried out, traffic models were generated for 10, 20, 30, 40, 50 nodes with cbr traffic sources, with maximum connections of 8 at a rate of 4 , 5, 10 , 15, 20 kbps.

setdest – the node-movement generator.

```
./setdest [-n num_of_nodes] [-p pausetime] [-s maxspeed] [-t simtime] \  [-x maxx] [-
y maxy] > [outdir/movement-file]
```

Mobility models were created for the simulations using 10, 20, 30, 40, 50 nodes, with pause times of 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 seconds, maximum speed of 20m/s, topology boundary of 500x500 , 670 x 670 and simulation time of 100secs.

5.6 Raw Data Parsing

After each simulation, trace files recording the traffic and node movements are generated. These files need to be parsed in order to extract the information needed to measure the performance metrics. The trace file format looks like:

```
s -t 0.267662078 -Hs 0 -Hd -1 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 –Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 –Ii 20 -Is 0.255 -Id -1.255 –It
```

This row of the trace file means:
a packet was sent (*s*) at time (*t*) 0.267662078 sec, from source node (*Hs*) 0 to destination node (*Hd*) 1. The source node id (*Ni*) is 0, it's x-co-ordinate (*Nx*) is 5.00, it's yco-ordinate (*Ny*) is 2.00, it's z-co-ordinate (*Nz*) is 0.00, it's energy level (*Ne*) is 1.000000, the trace level (*Nl*) is RTR and the node event (*Nw*) is blank. The MAC level information is given by duration (*Ma*) 0, destination Ethernet address (*Md*) 0,the source Ethernet address (*Ms*) is 0 and Ethernet type (*Mt*) is 0. The IP packet level information like packet id (*Ii*), source address. source port number is given by (Is) while the destination address. destination port number is (*Id*).

5.6.1 Packet Delivery Ratio

Calculate the number of "received packets" of the trace form:

```
/^s *- Nl AGT.*-Is (\d{1,3})\.\d{1,3} -Id (\d{1,3})\.\d{1,3}.*-It cbr.*-Ii (\d{1,6})/
```

AGT => Agent Level Trace

Calculate the number of "received packets" of the trace form:

```
/^r -t (\d{1,3}\.\d{9}).*-Nl AGT.*-Is (\d{1,3})\.\d{1,3} -Id (\d{1,3})\.\d{1,3}.*-It cbr.*-Ii (\d{1,6})/
```

packet delivery fraction (pdf %) = (received packets/ sent packets) *100

Received packets and sent packets number could be easily obtained from the first element of each line of the trace file.

5.6.2 Average End-End Packet Delivery Time

For each packet with id (*Ii*) of trace level (*AGT*) and type (*cbr*), calculate the send(*s*) time (*t*) and the receive (*r*) time (*t*) and average it.

5.6.3 Routing Overhead

Calculate the routing packet sent:

/[s or f]. * -N RTR. * -It (? : AODV | DSR |DSDV| message) –II(1,4)/

Routing Overhead = (routing packets sent / receives)

We conducted simulation of AODV , DSR , DSDV and TORA routing protocols using a following parameters :

- Number of Nodes : 30
- Pause time for node movement : 10 m/s
- Bandwidth : 2 Mbps
- Maximum speed : 20 meters per second
- Traffic Type : Constant Bit Rate
- CBR sources sending rate : 4 packets per second
- Data packet size : 512 bytes
- Simulation time : 100 seconds
- Environment Size : 670m x 670m

6.1 Simulation of AODV protocols

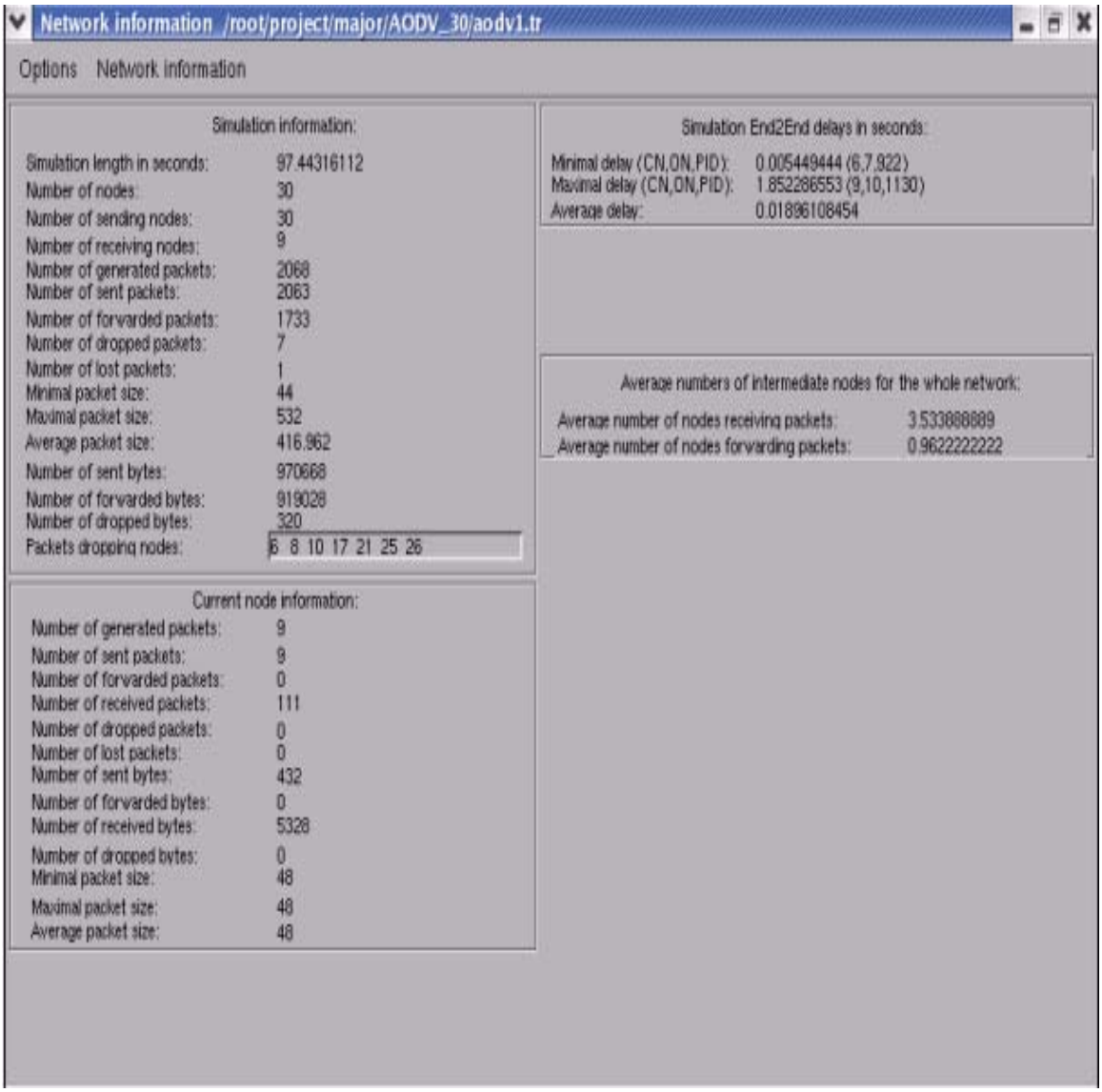

Figure 6.1.1 : Simulation Information

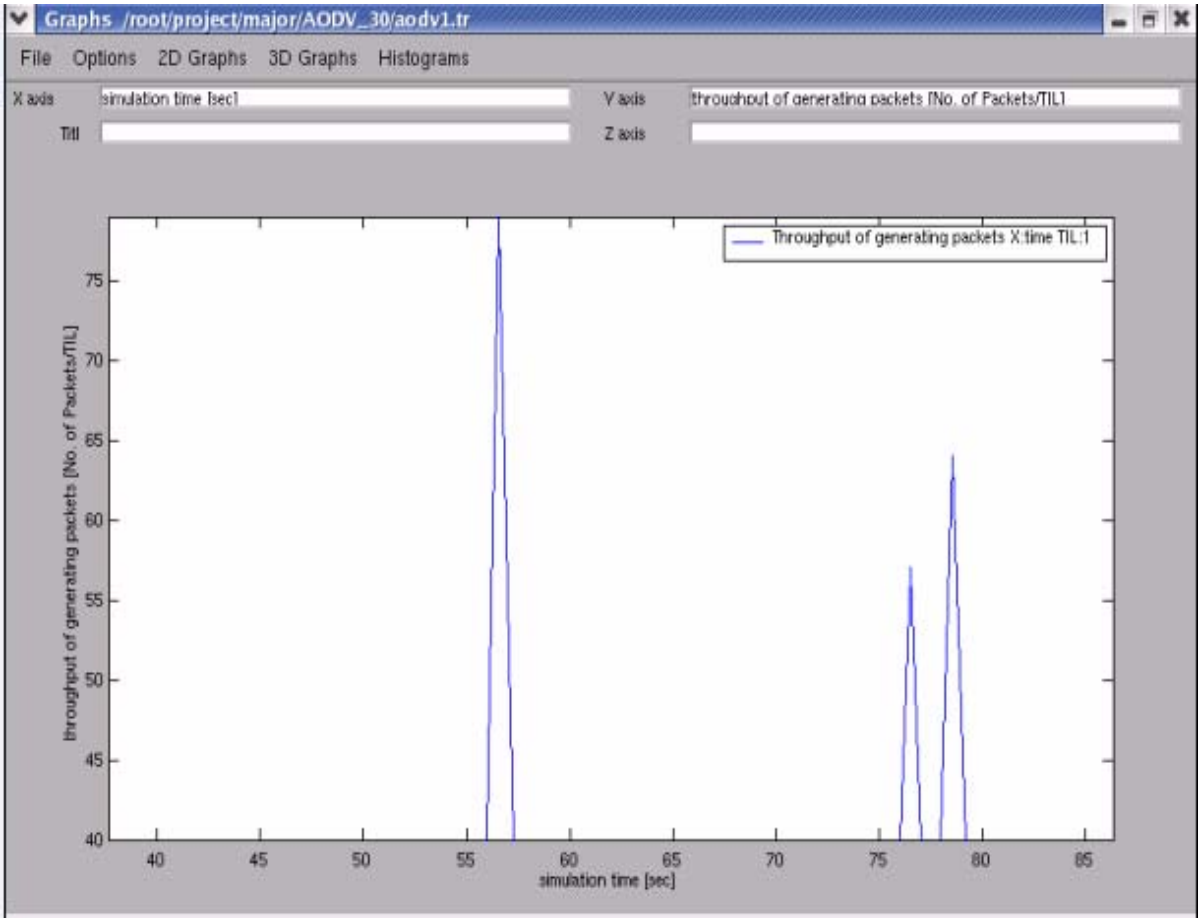

Figure 6.1.2 : Throughput of generating Packets

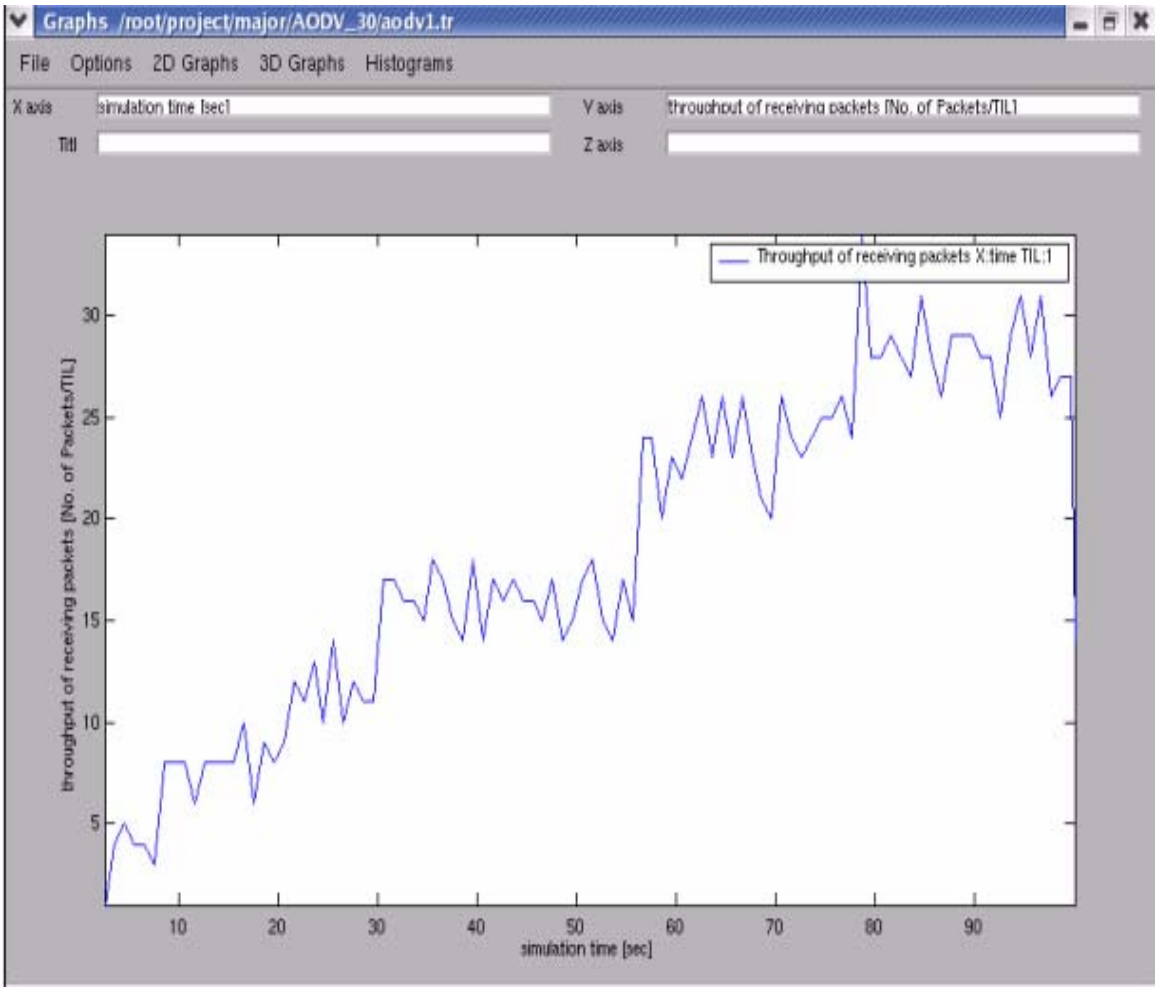

Figure 6.1.3 : Throughput of Receiving Packets

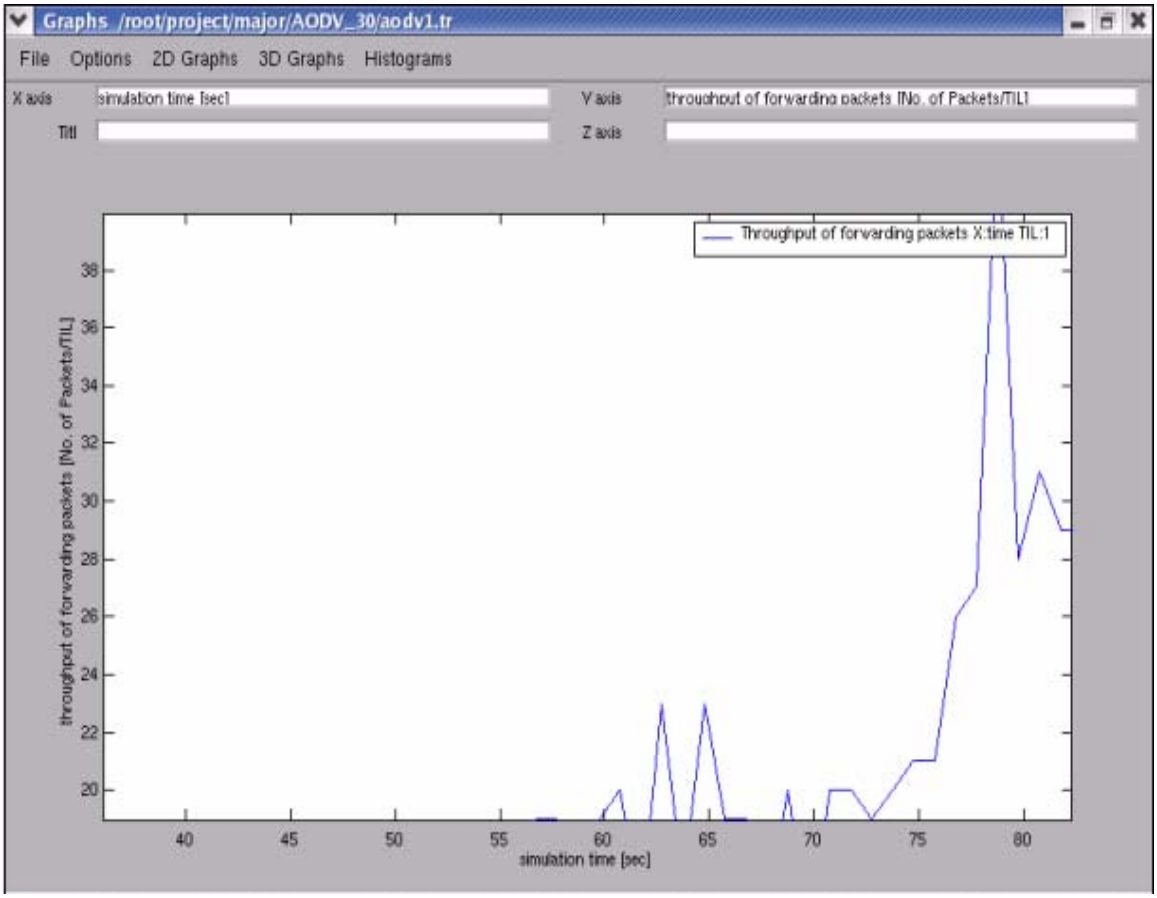

Figure 6.1.4 : Throughput of forwarding Packets

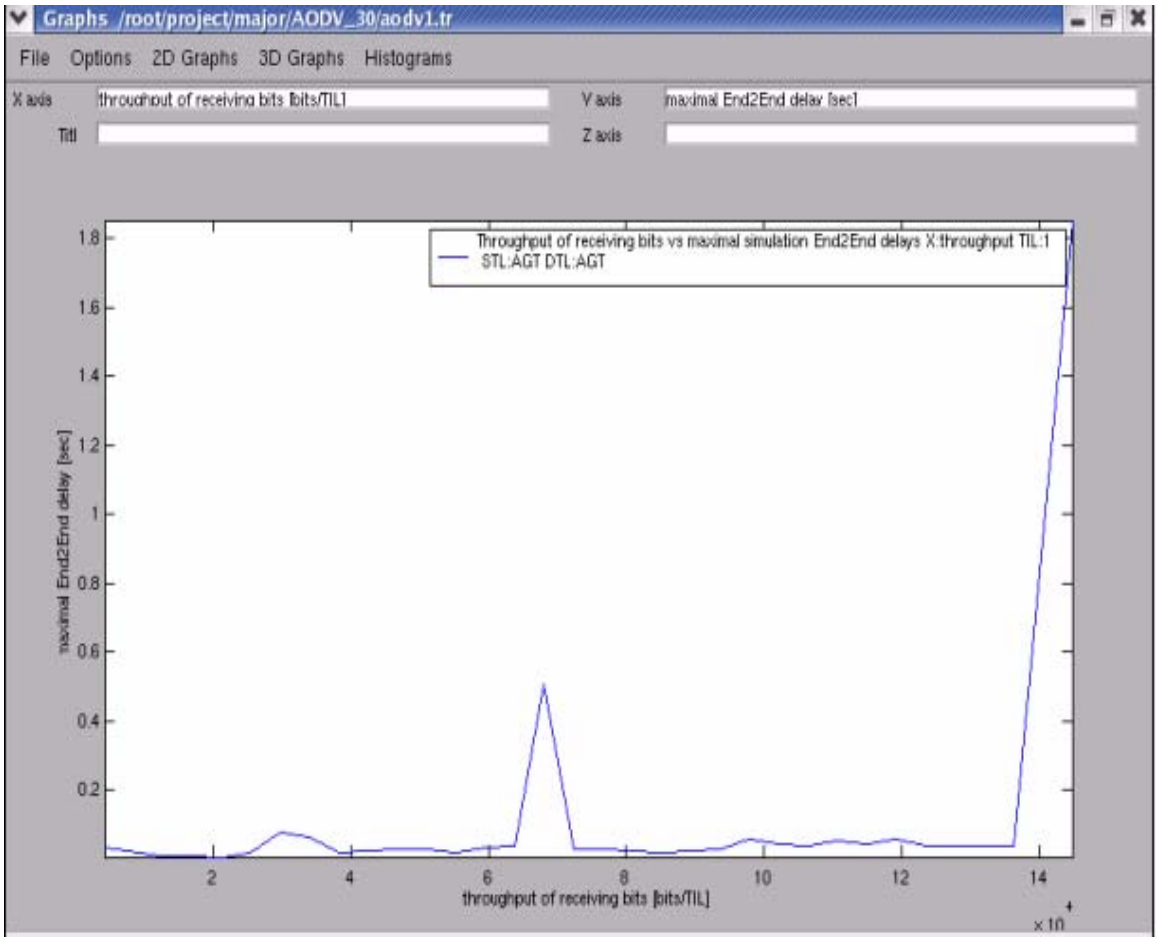

Figure 6.1.5 : Throughput of Receiving bits Vs Maximal simulation End2End Delays

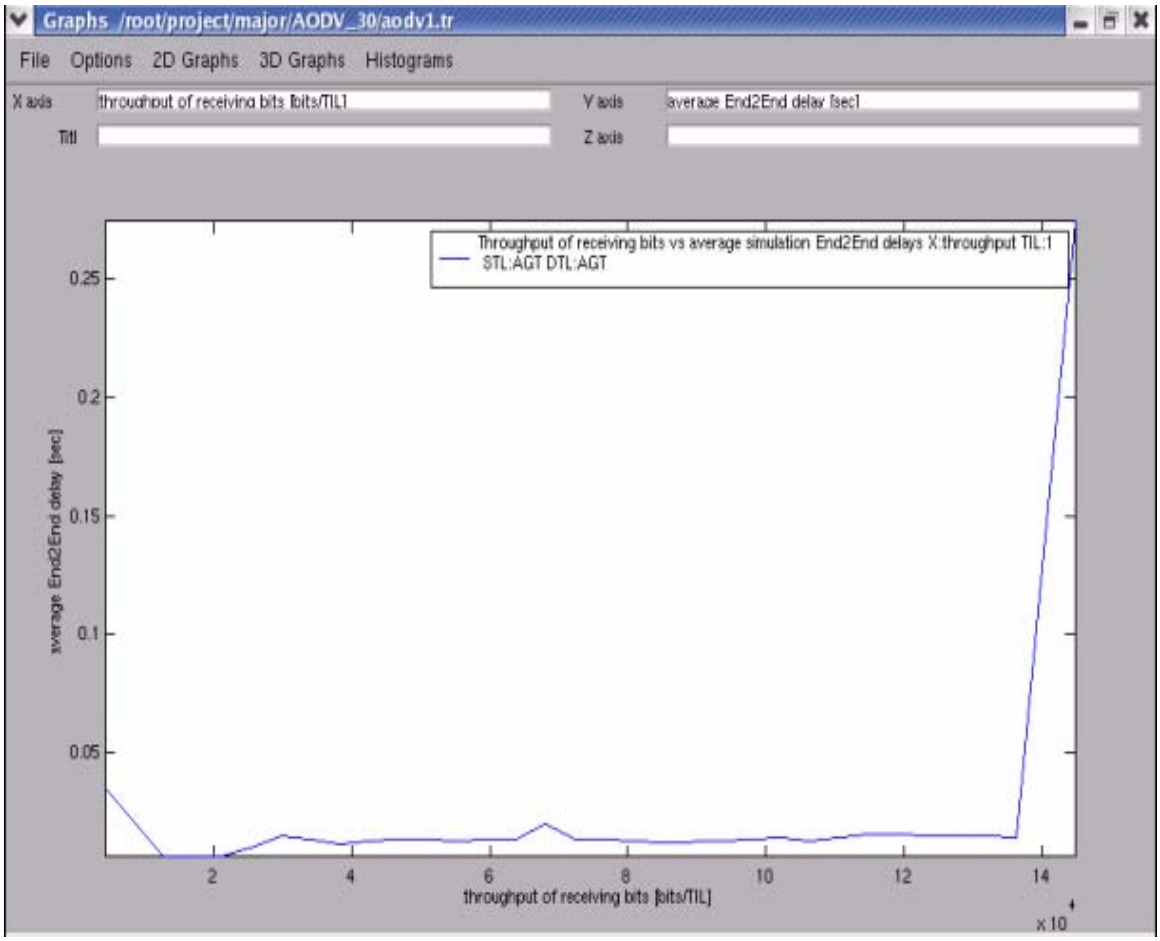

Figure 6.1.6 : Throughput of Receiving bits Vs Average simulation End2End Delays

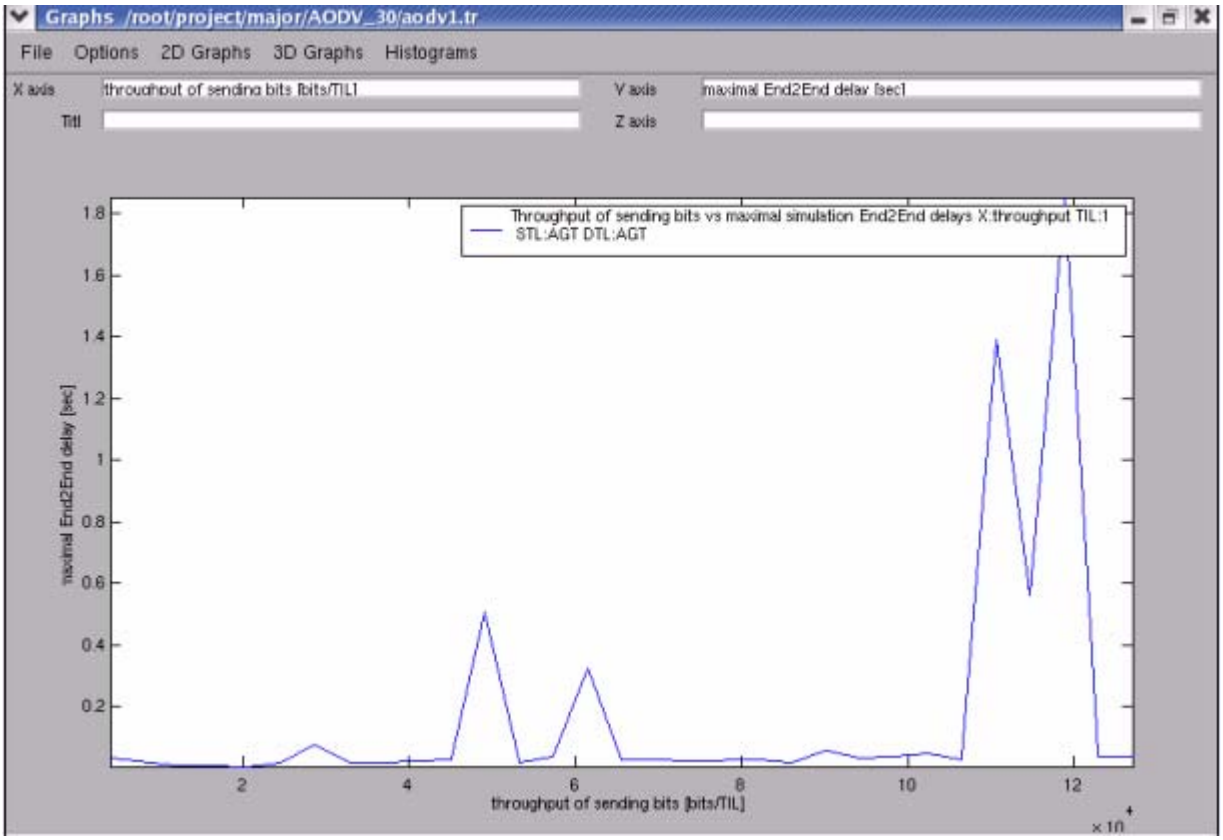

Figure 6.1.7 : Throughput of Sending bits Vs Maximal simulation End2End Delays

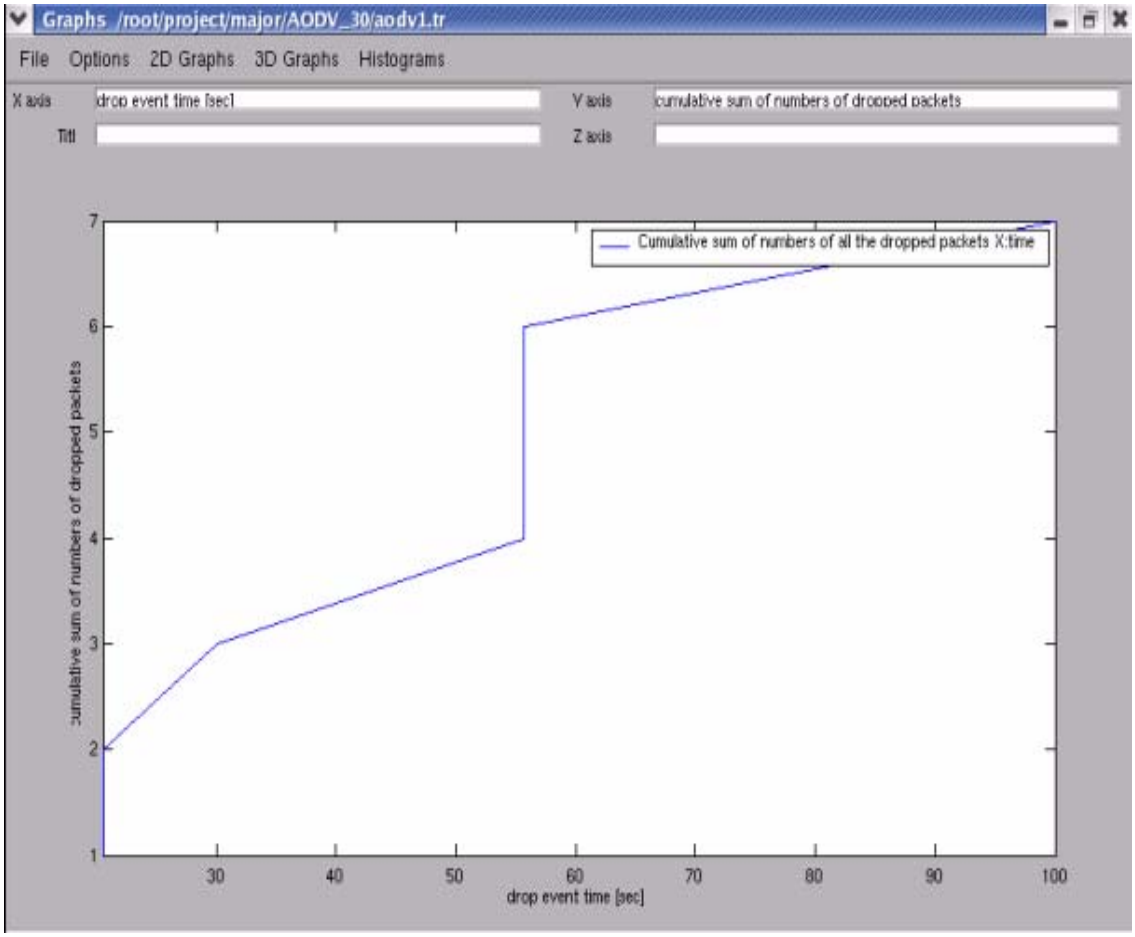

Figure 6.1.8 : Cumulative sum of numbers of all the dropped Packets

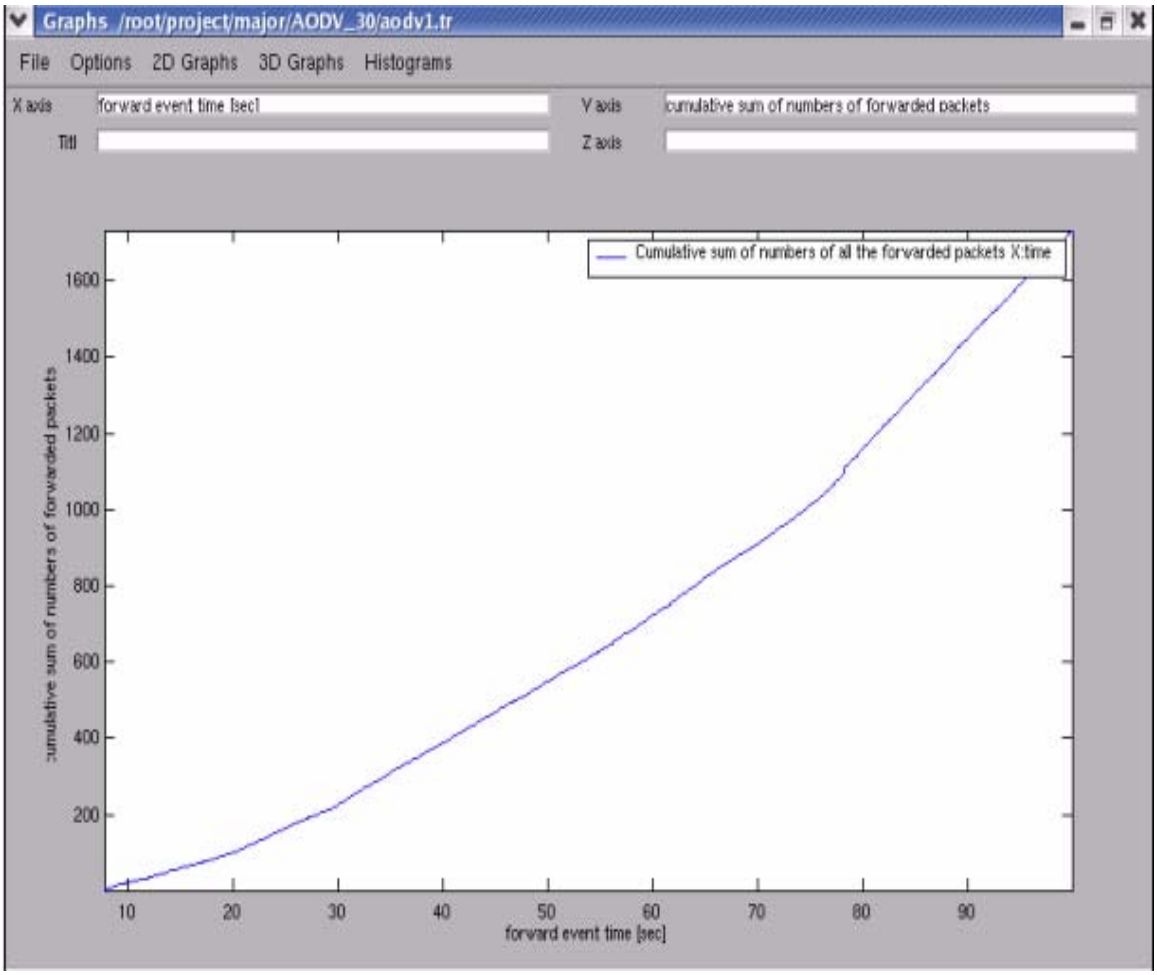

Figure 6.1.9 Cumulative sum of numbers of all the Forwarded Packets

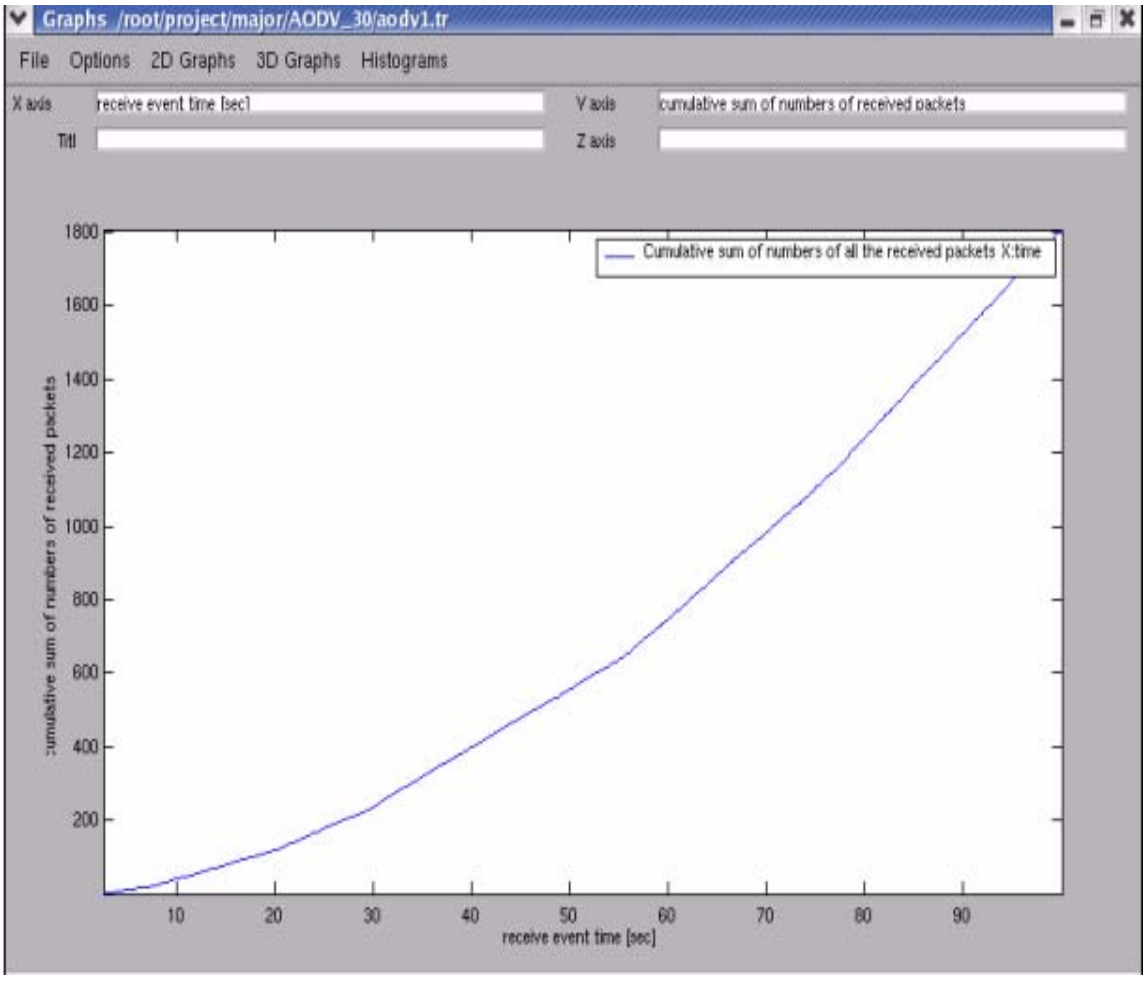

Figure 6.1.10 : Cumulative sum of numbers of all the Received Packets

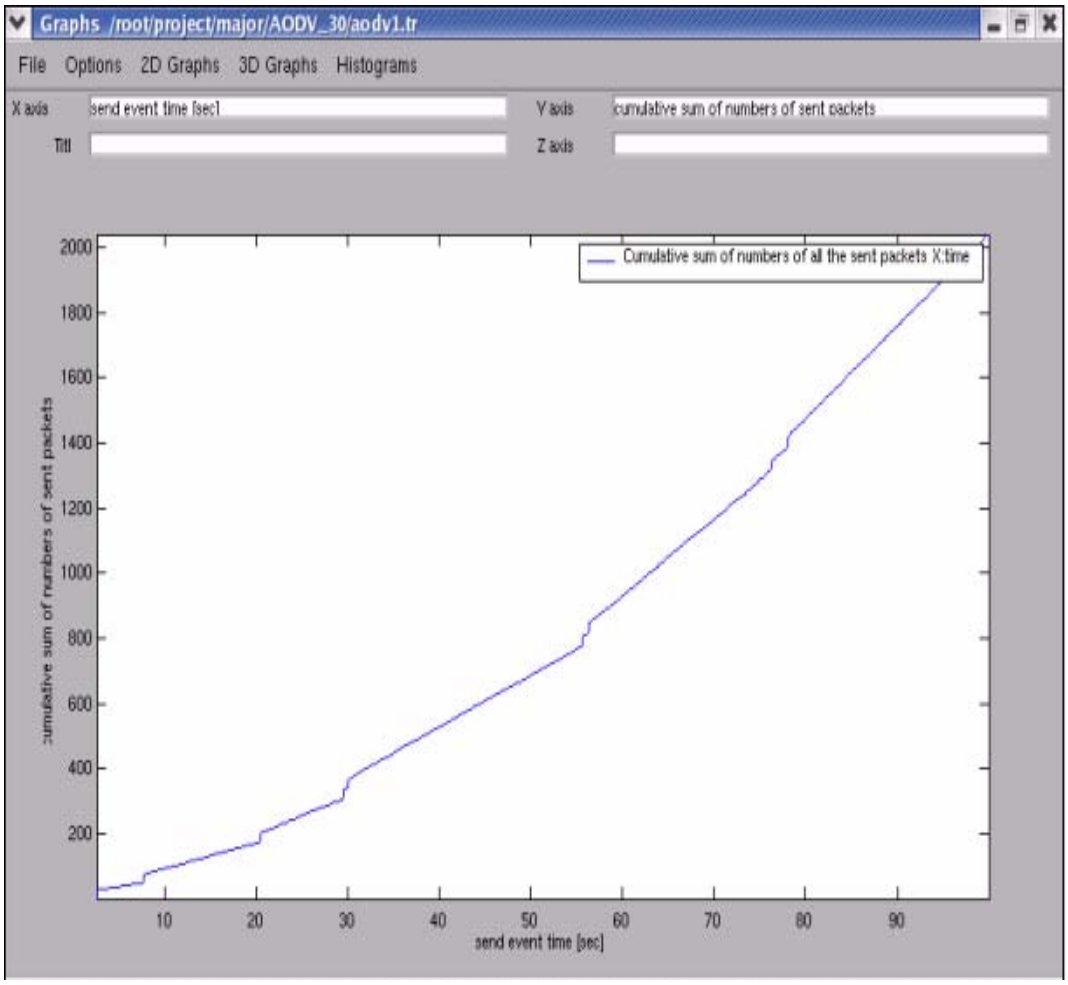

Figure 6.1.11 : Cumulative sum of numbers of all the Sent Packets

6.2 Simulation of DSR Protocol

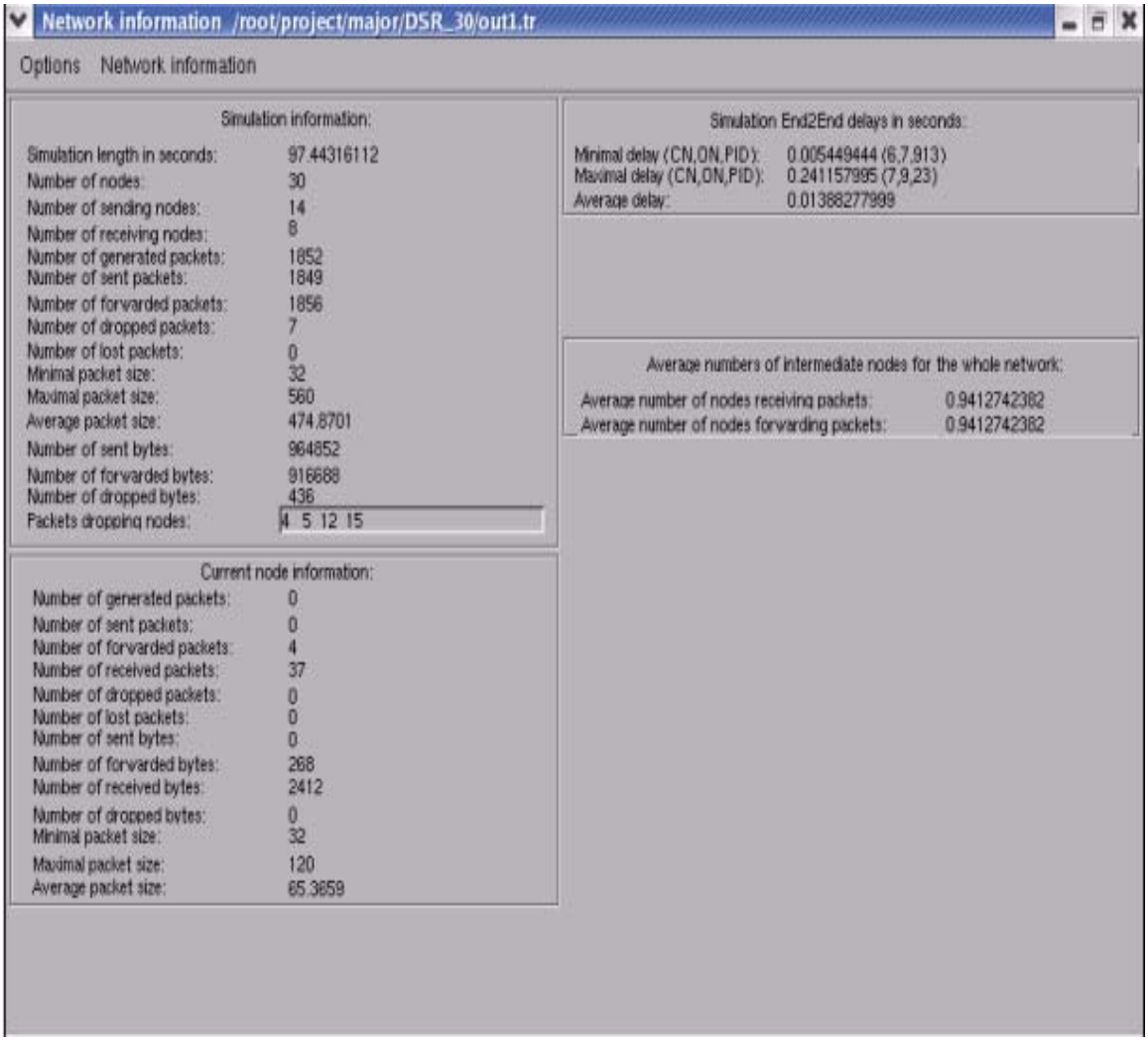

Figure 6.2.1 : Simulation Information

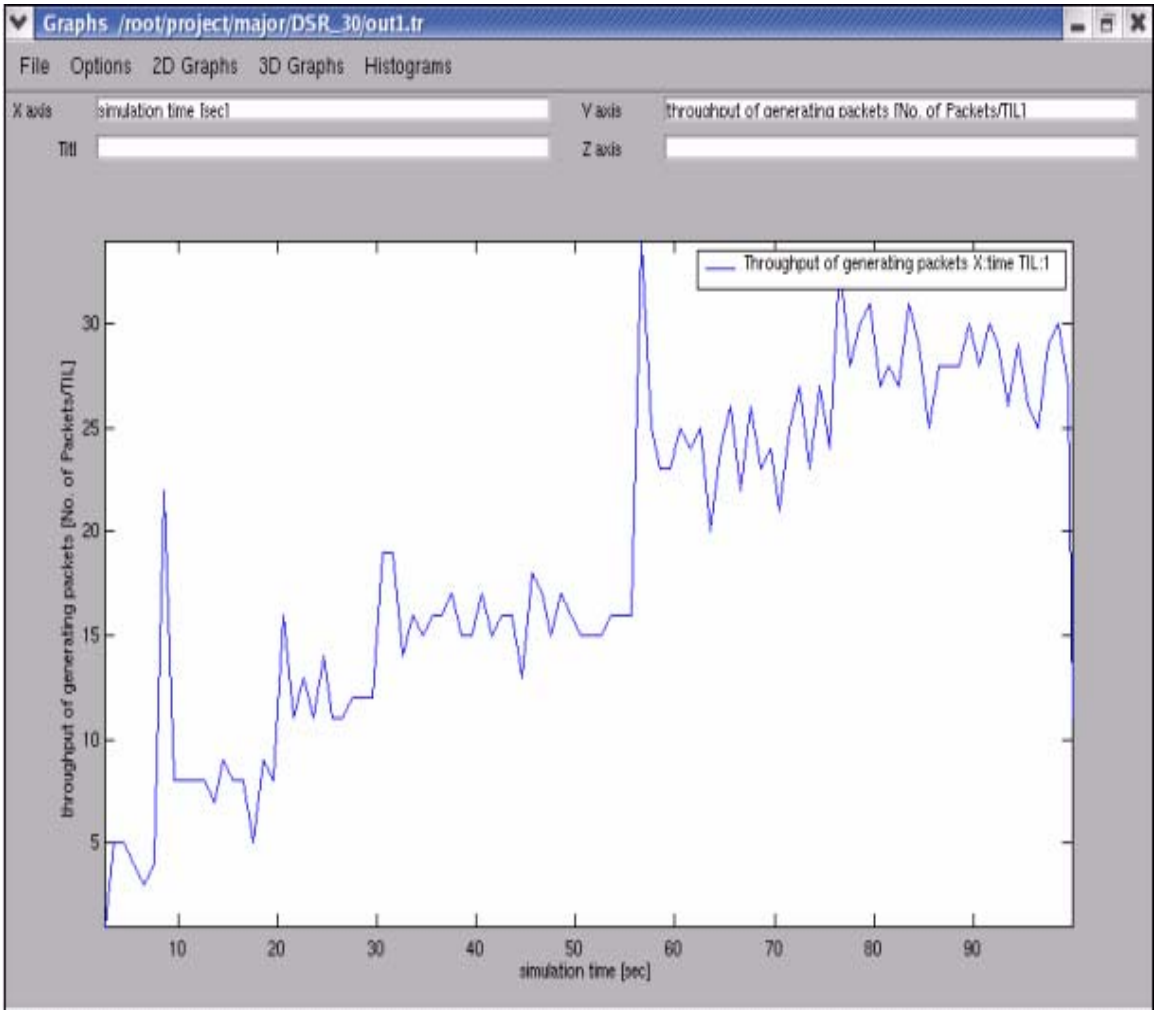

Figure 6.2.2 : Throughput of generating Packets

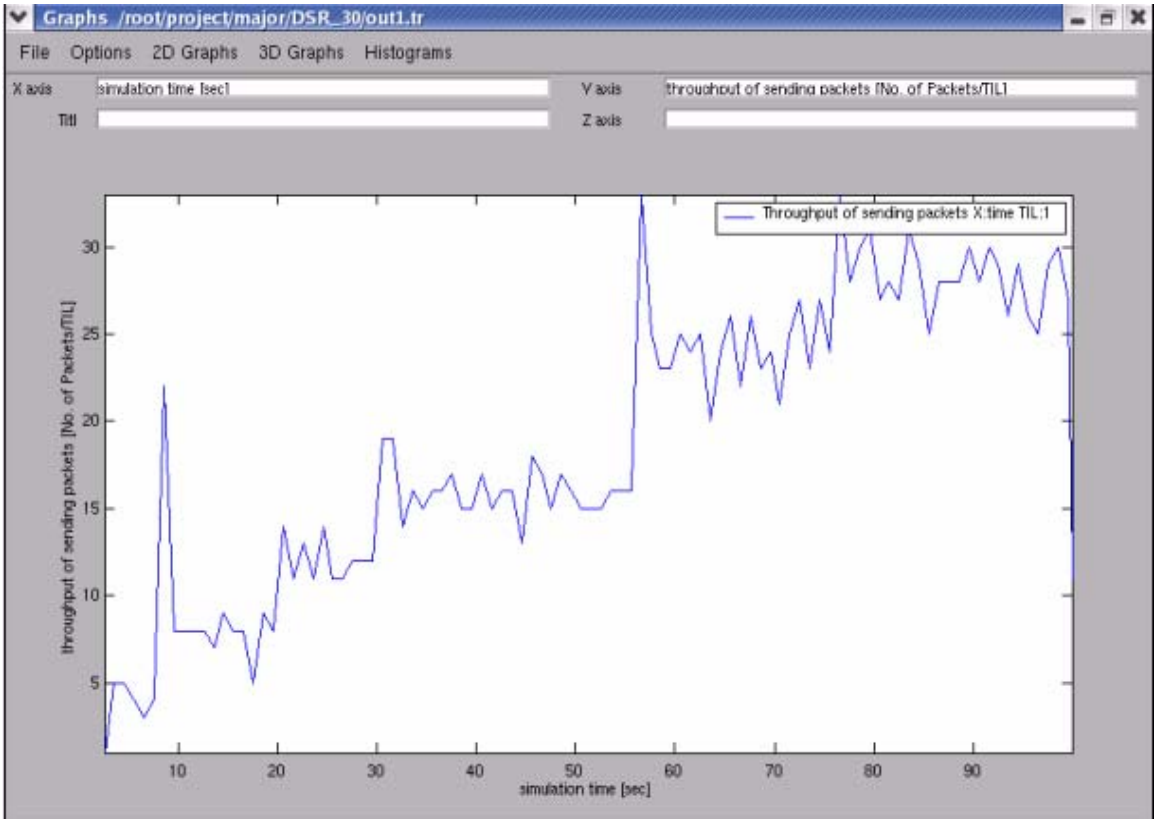

Figure 6.2.3 : Throughput of Sending Packets

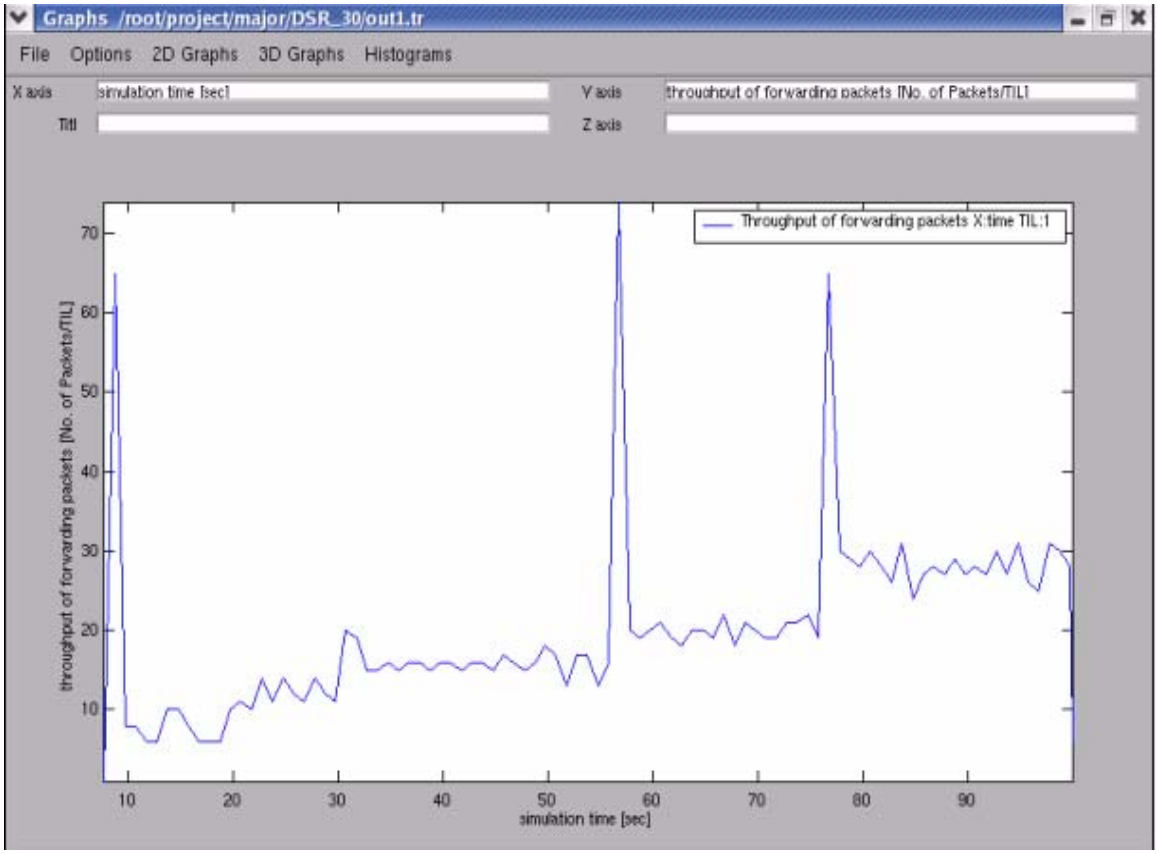

Figure 6.2.4 : Throughput of forwarding Packets

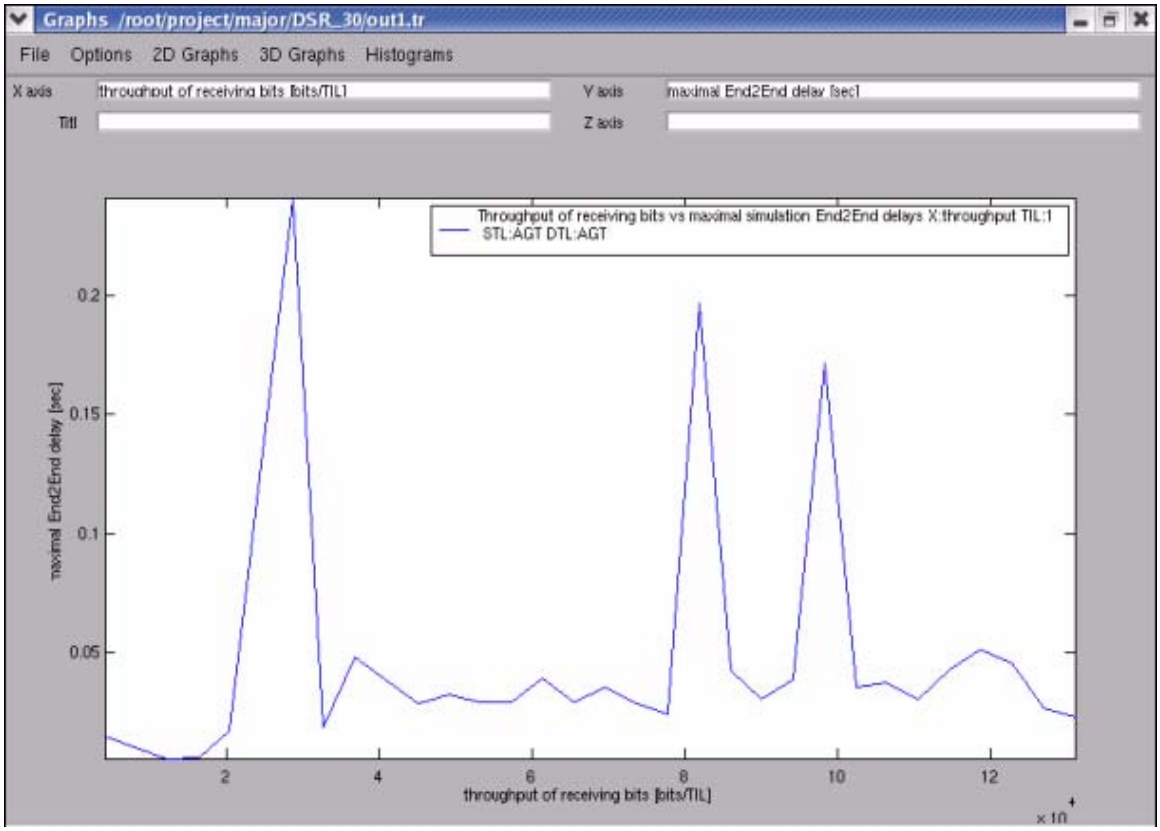

Figure 6.2.5 : Throughput of receiving bits Vs maximal simulation End2End delay

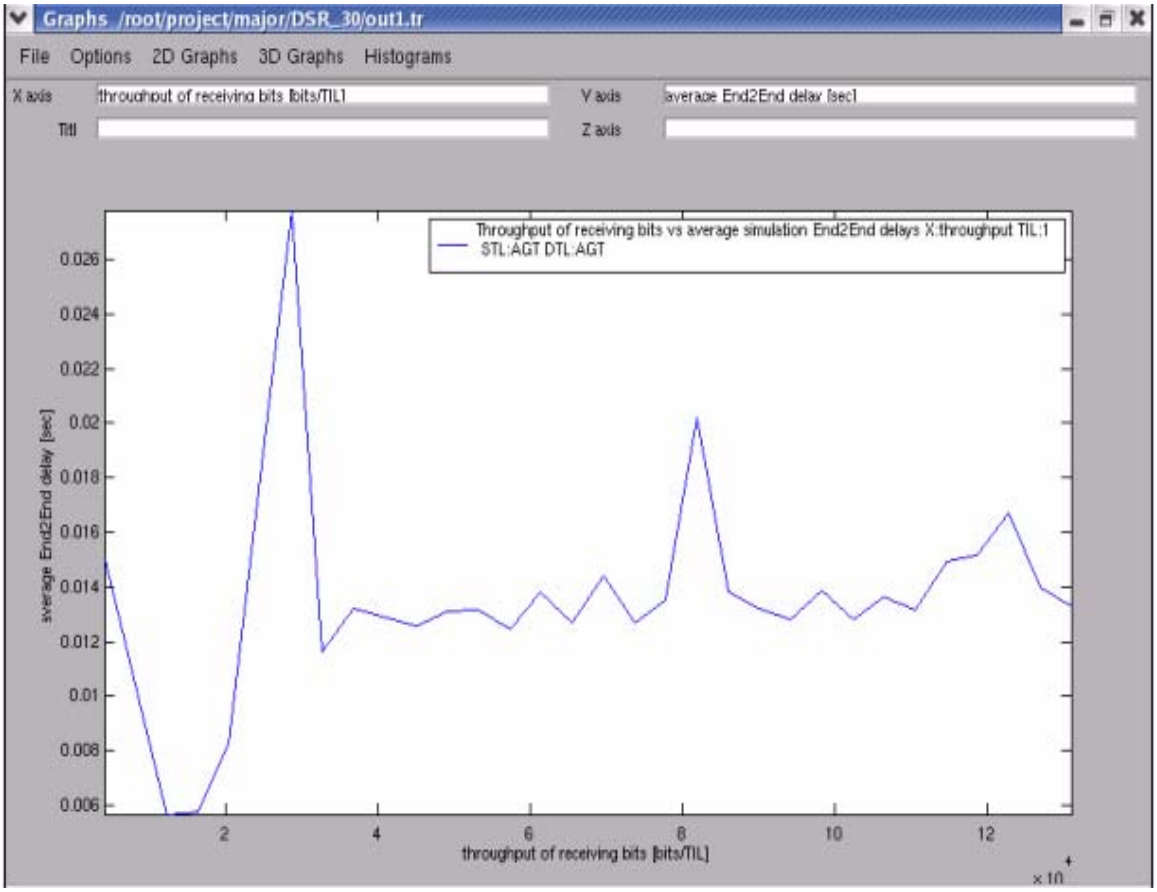

Figure 6.2.6 : Throughput of receiving bits Vs average simulation End2End delay

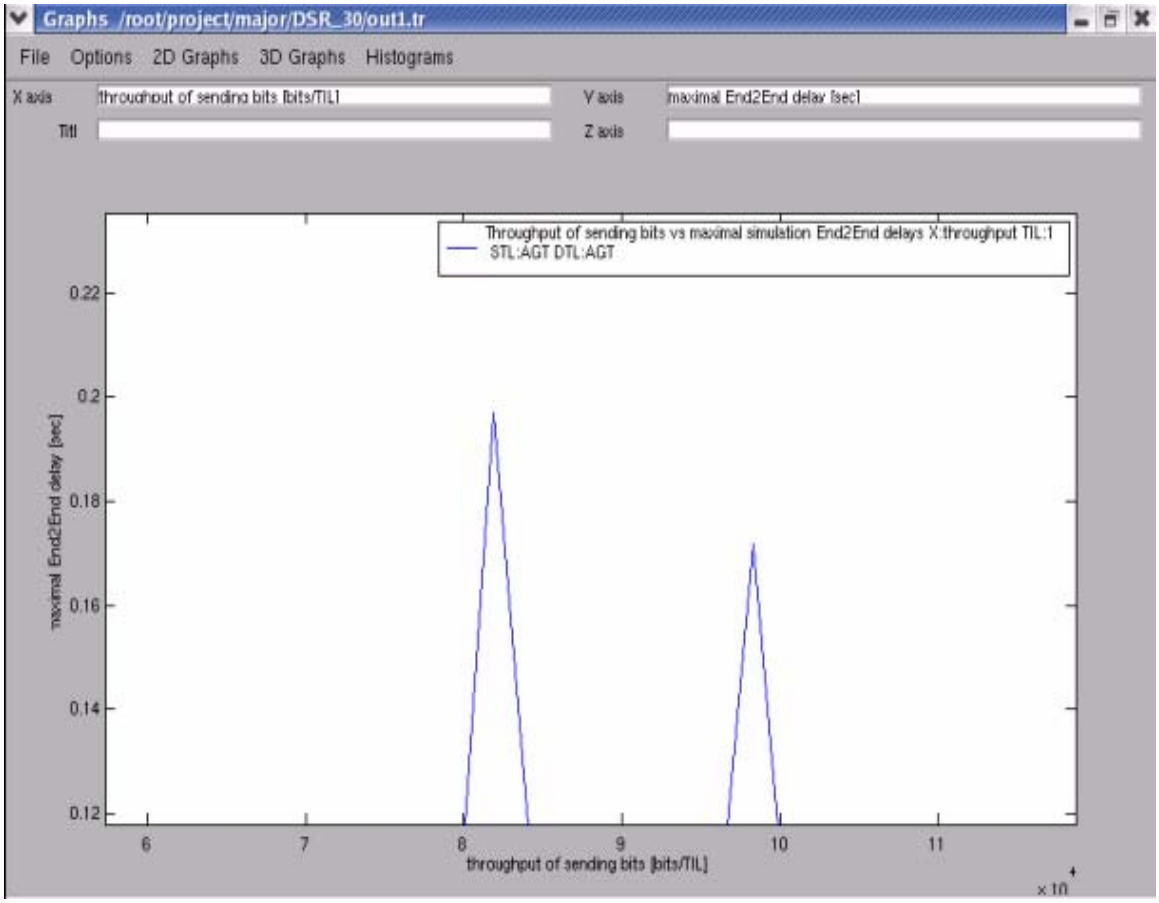

Figure 6.2.7 Throughput of sending bits Vs maximal simulation End2End delay

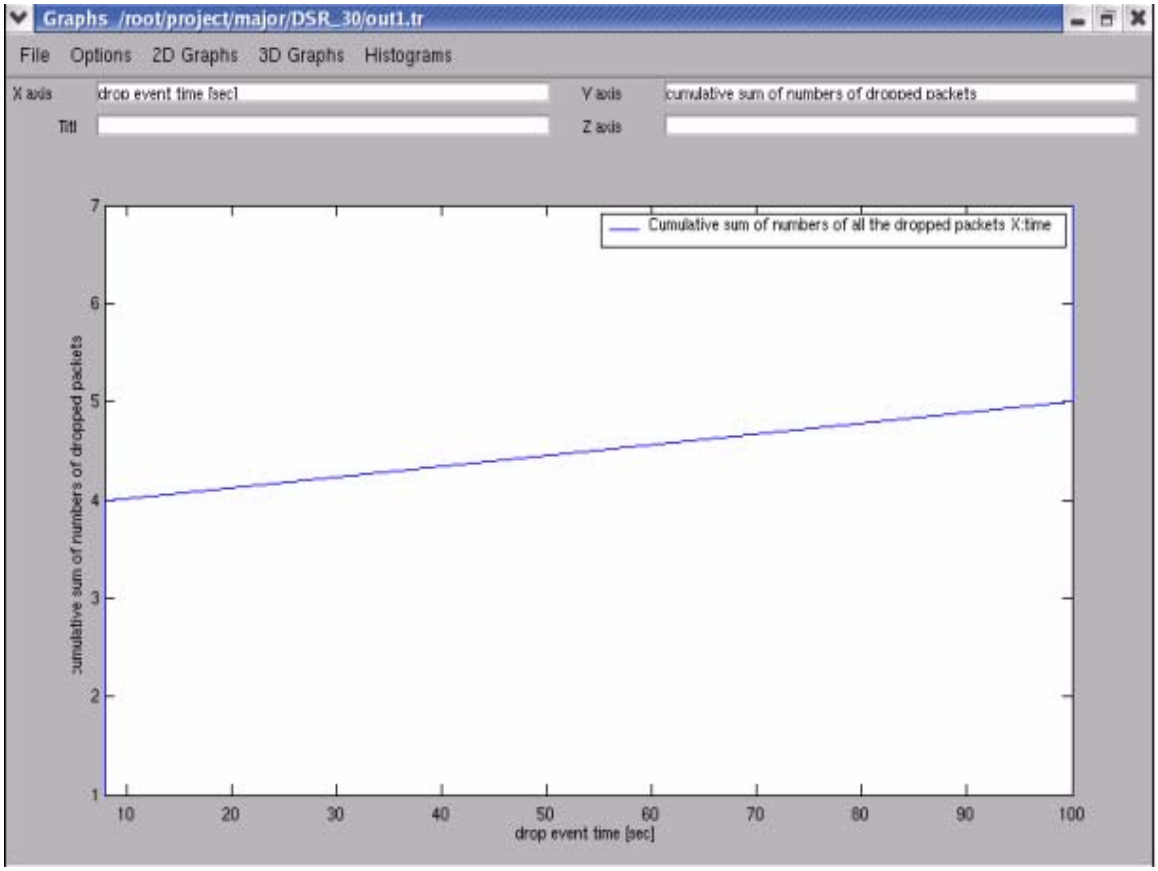

Figure 6.2.8 : Cumulative sum of numbers of all the dropped Packets

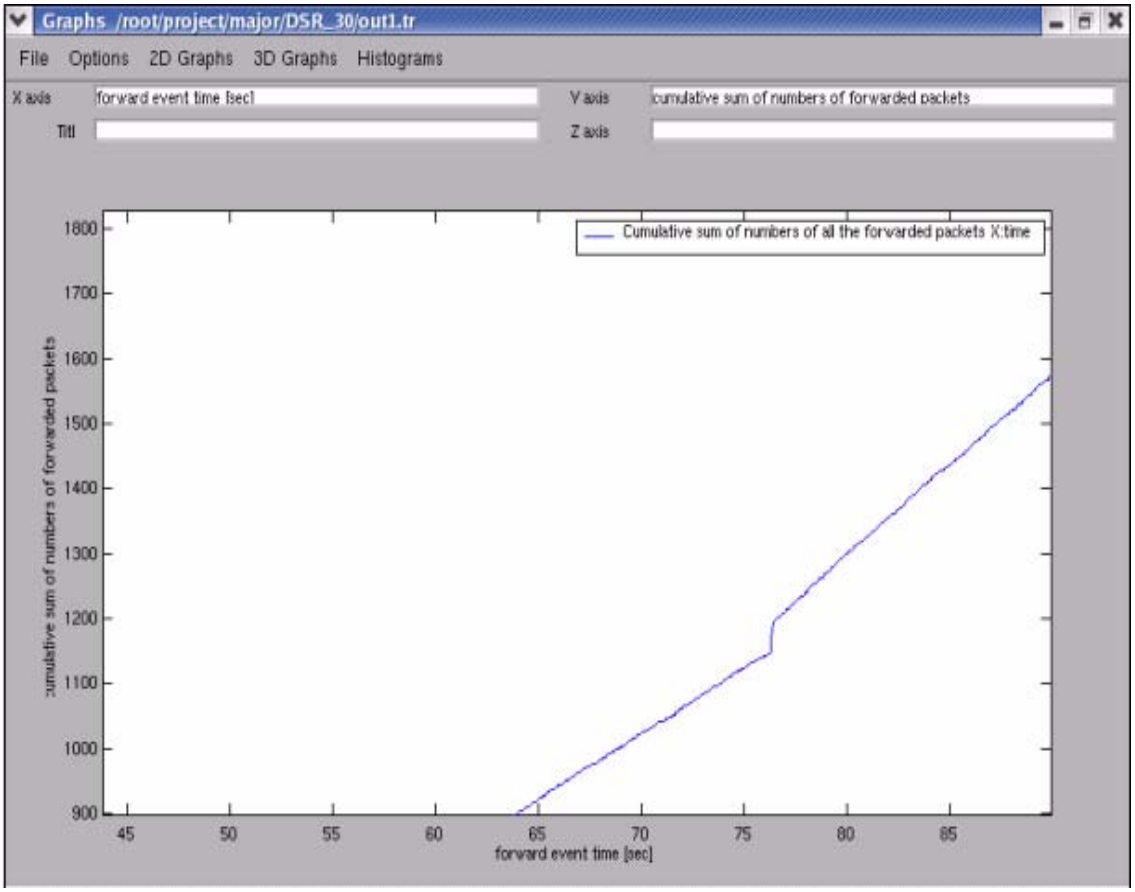

Figure 6.2.9 : Cumulative sum of numbers of all the forwarded Packets

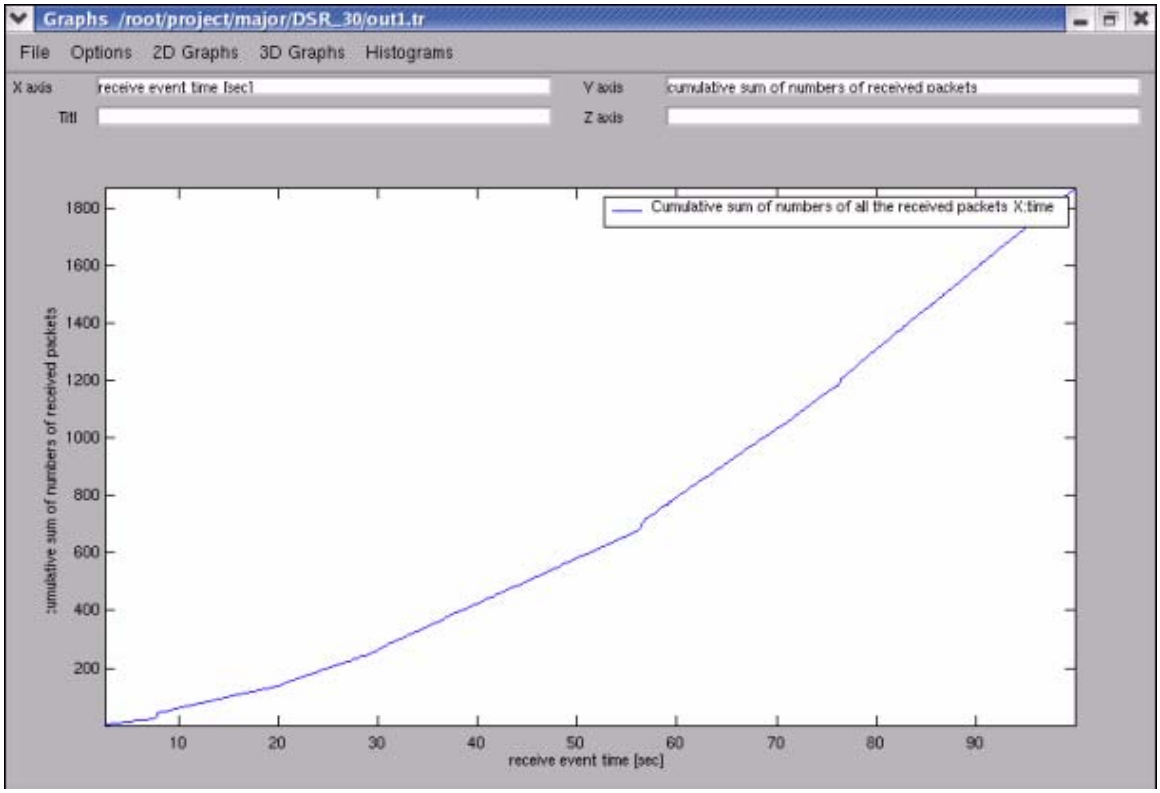

Figure 6.2.10 : Cumulative sum of numbers of all the received Packets

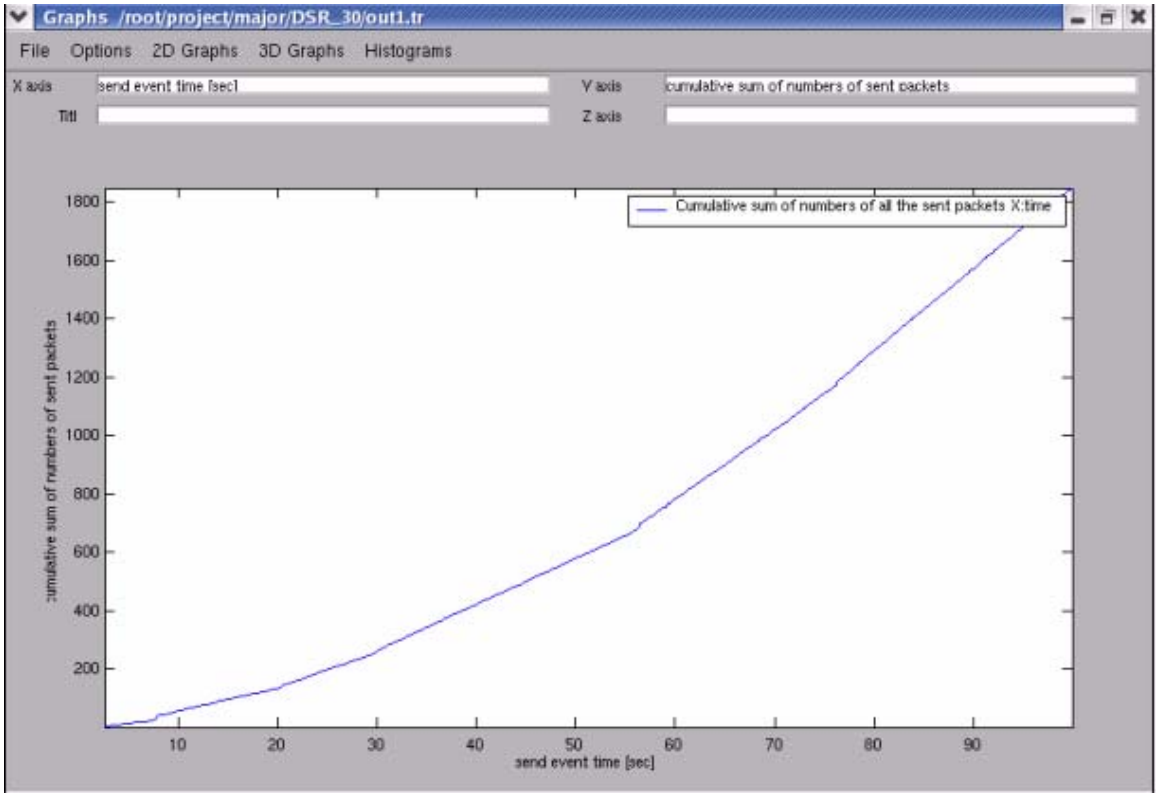

Figure 6.2.11 : Cumulative sum of numbers of all the sent Packets

6.3 Simulation of DSDV Protocol

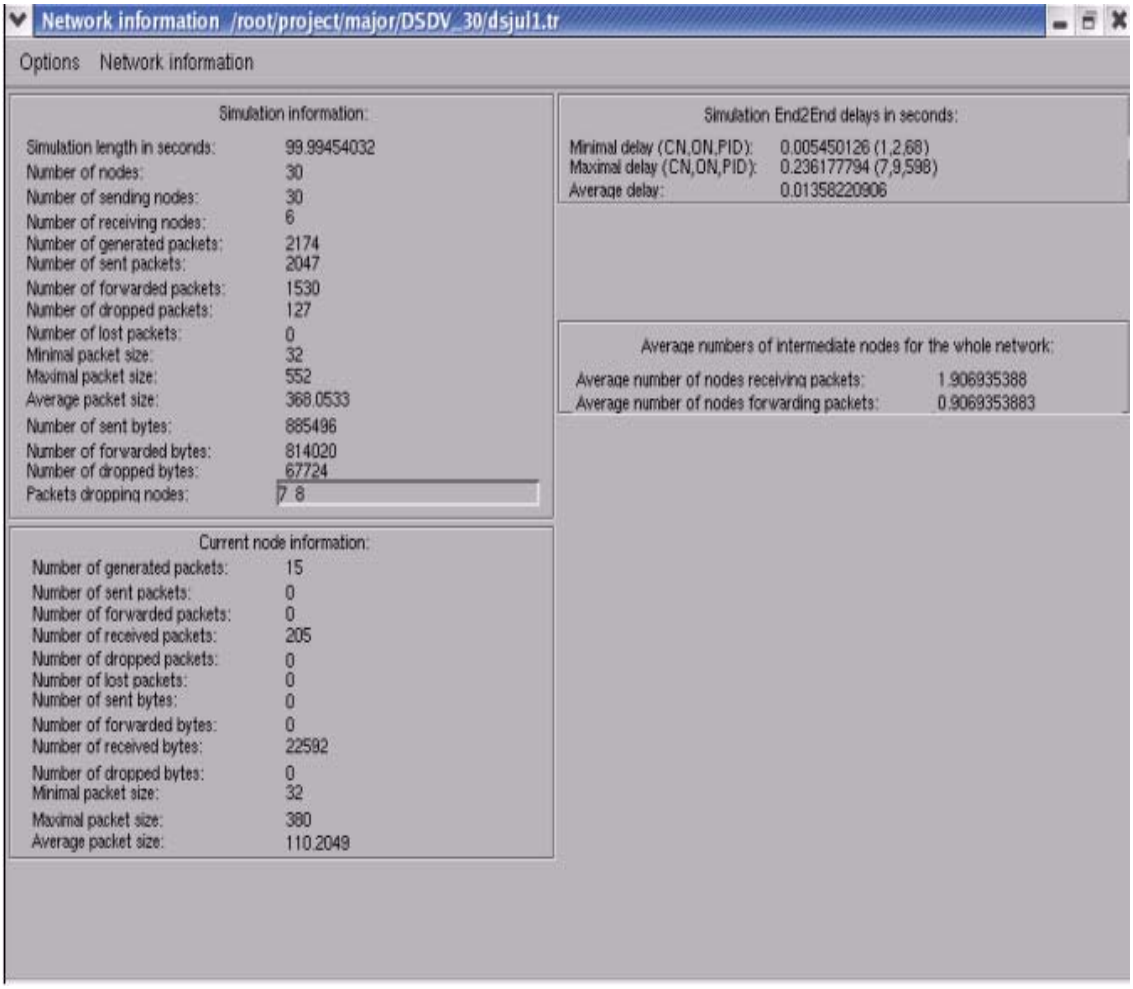

Figure 6.3.1 : Simulation Information

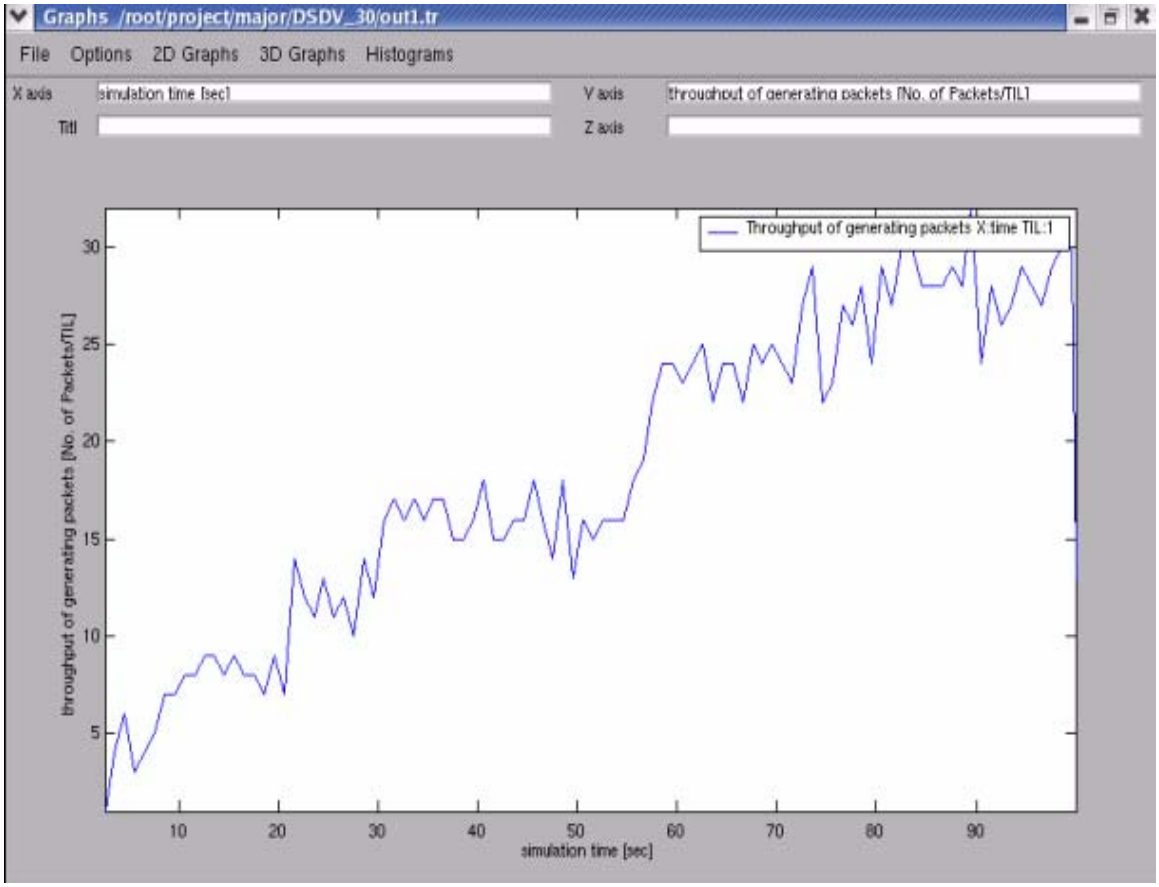

Figure 6.3.2 : Throughput of generating Packets

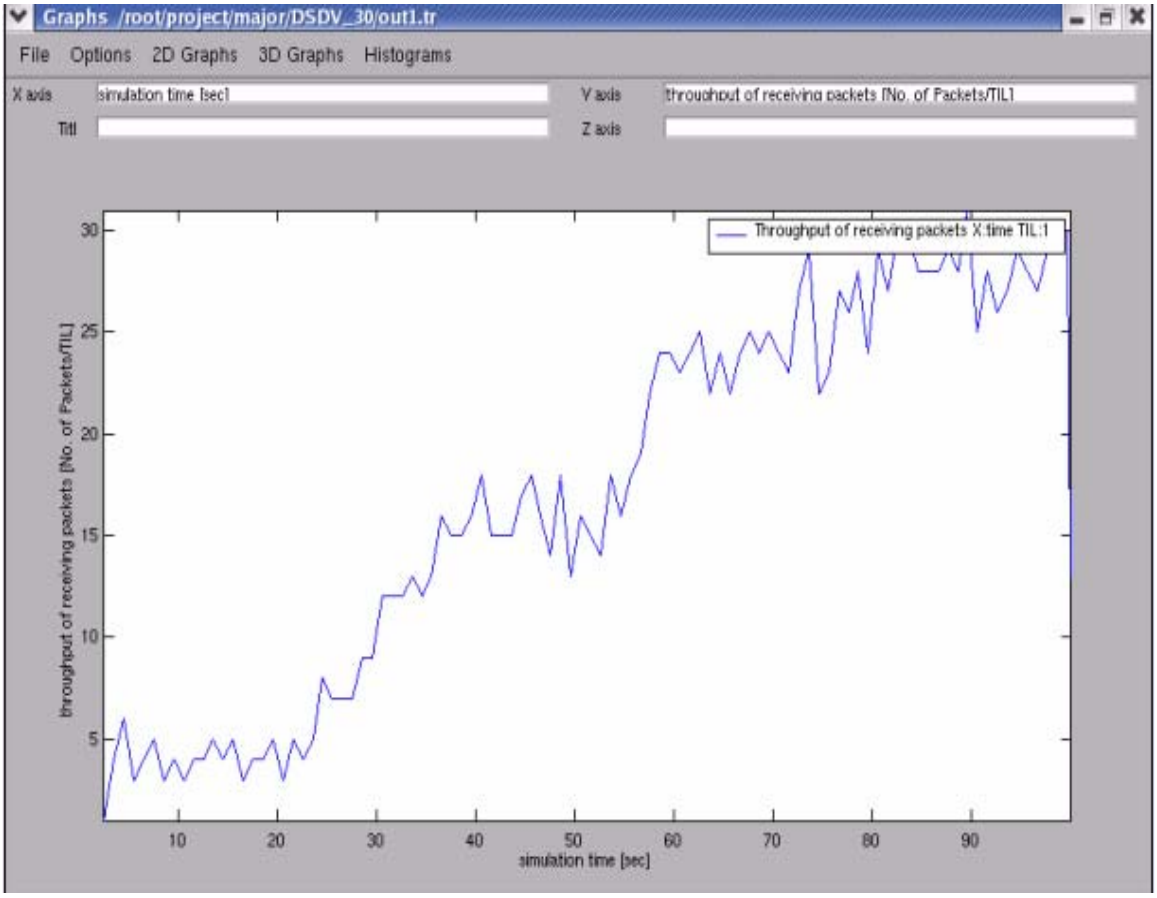

Figure 6.3.3 : Throughput of receiving Packets

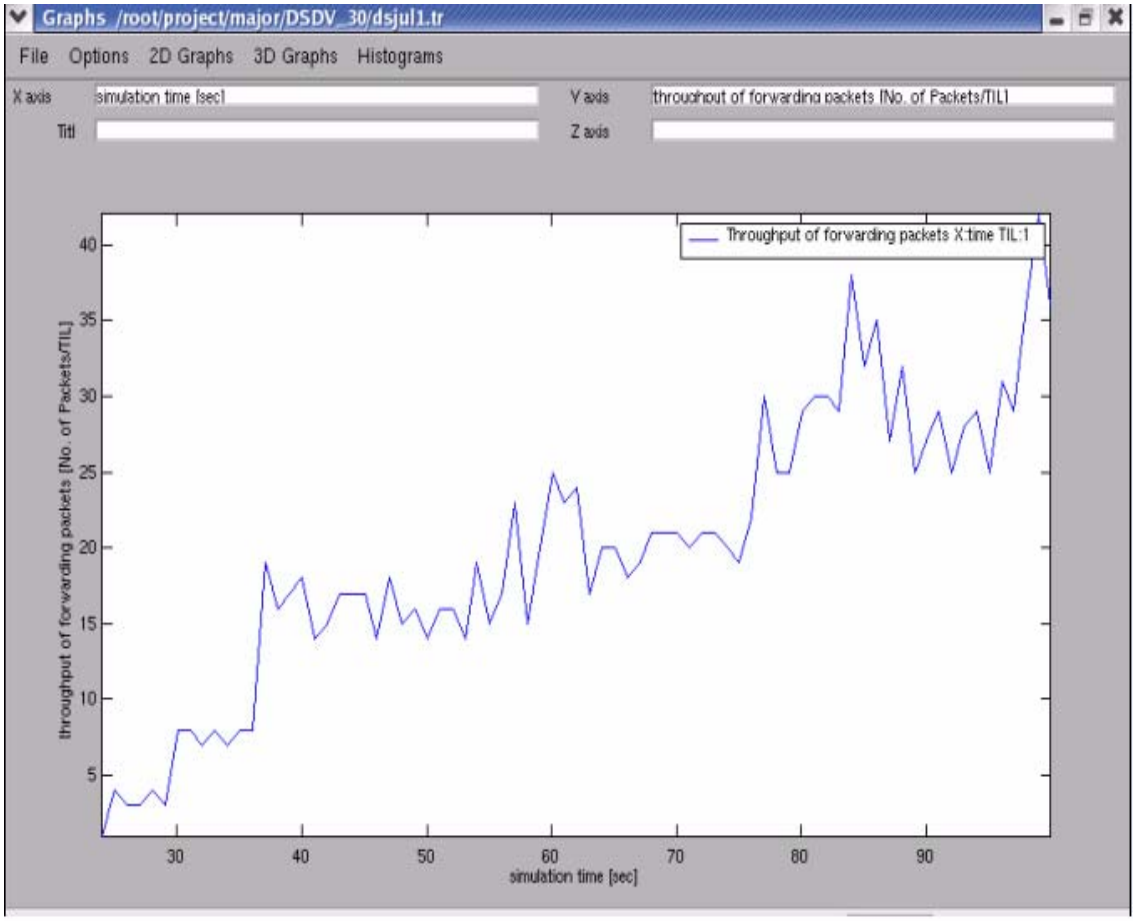

Figure 6.3.4 : Throughput of forwarding Packets

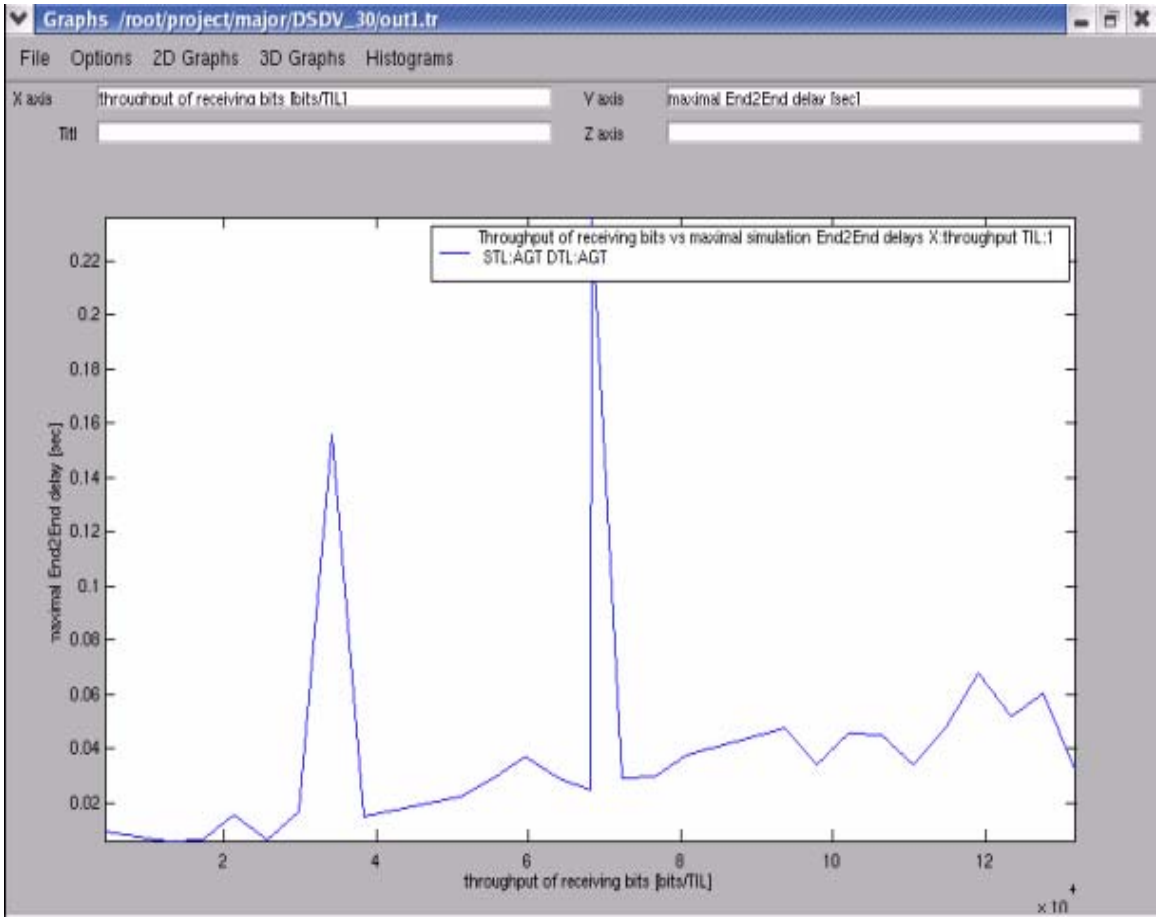

Figure 6.3.5 : Throughput of receiving bits Vs maximal simulation End2End Delay

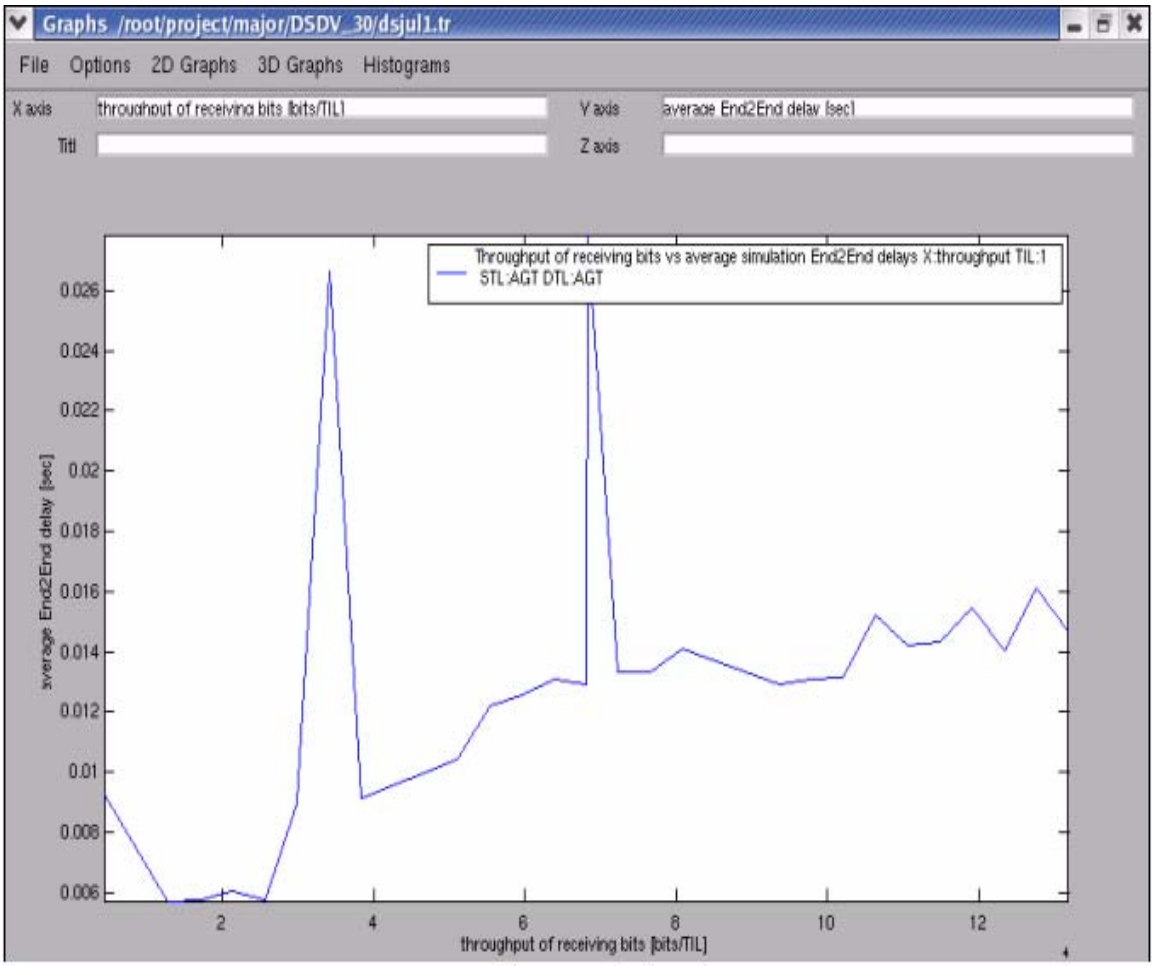

Figure 6.3.6 : Throughput of receiving bits Vs maximal simulation End2End Delay

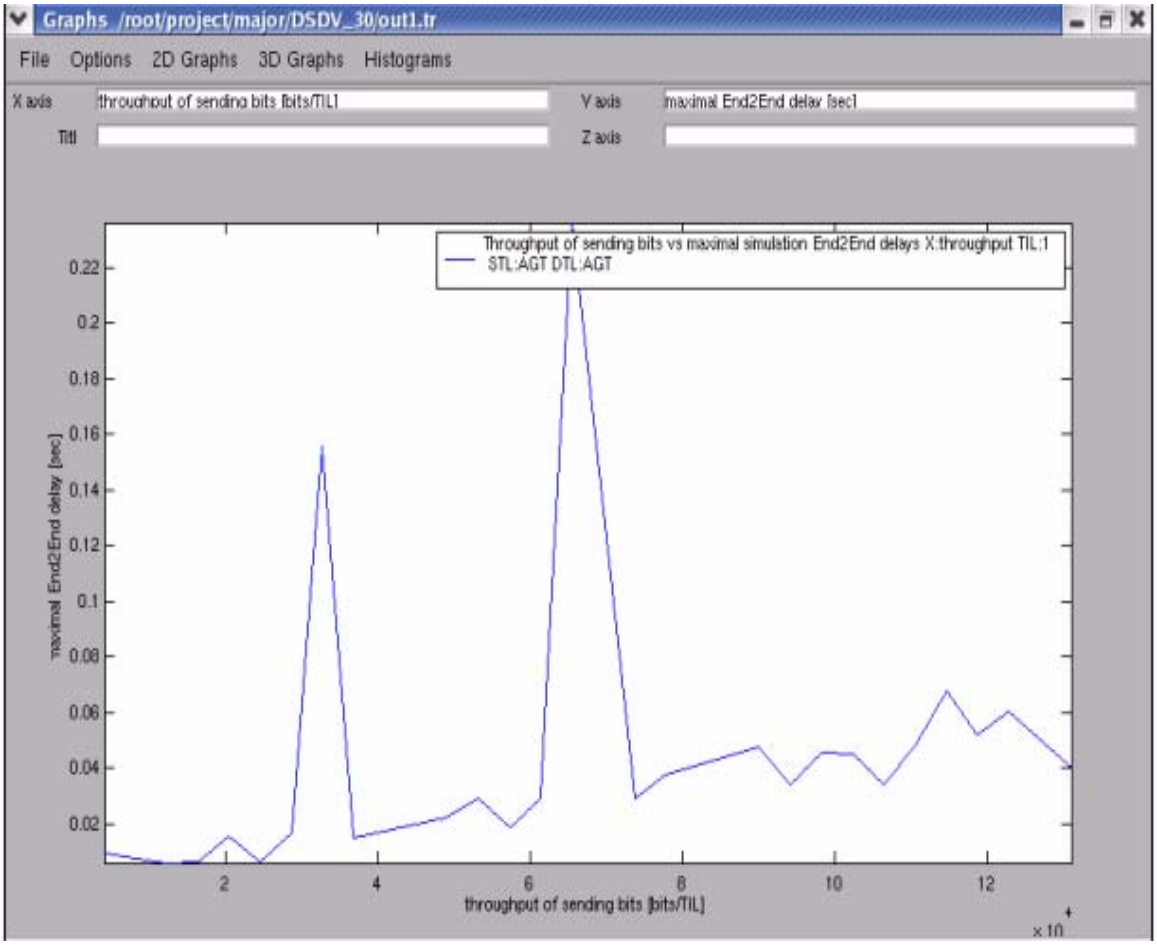

Figure 6.3.7 : Throughput of sending bits Vs maximal simulation End2End Delay

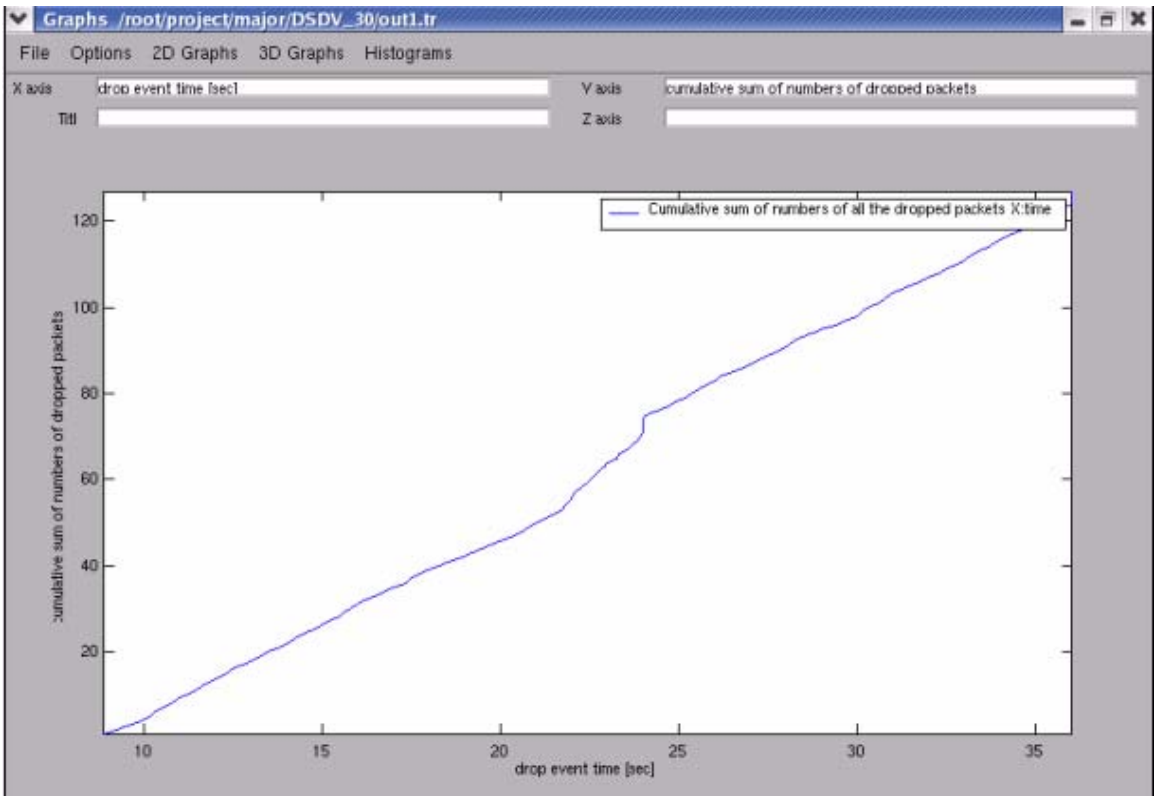

Figure 6.3.8 : Cumulative sum of numbers of all dropped Packets

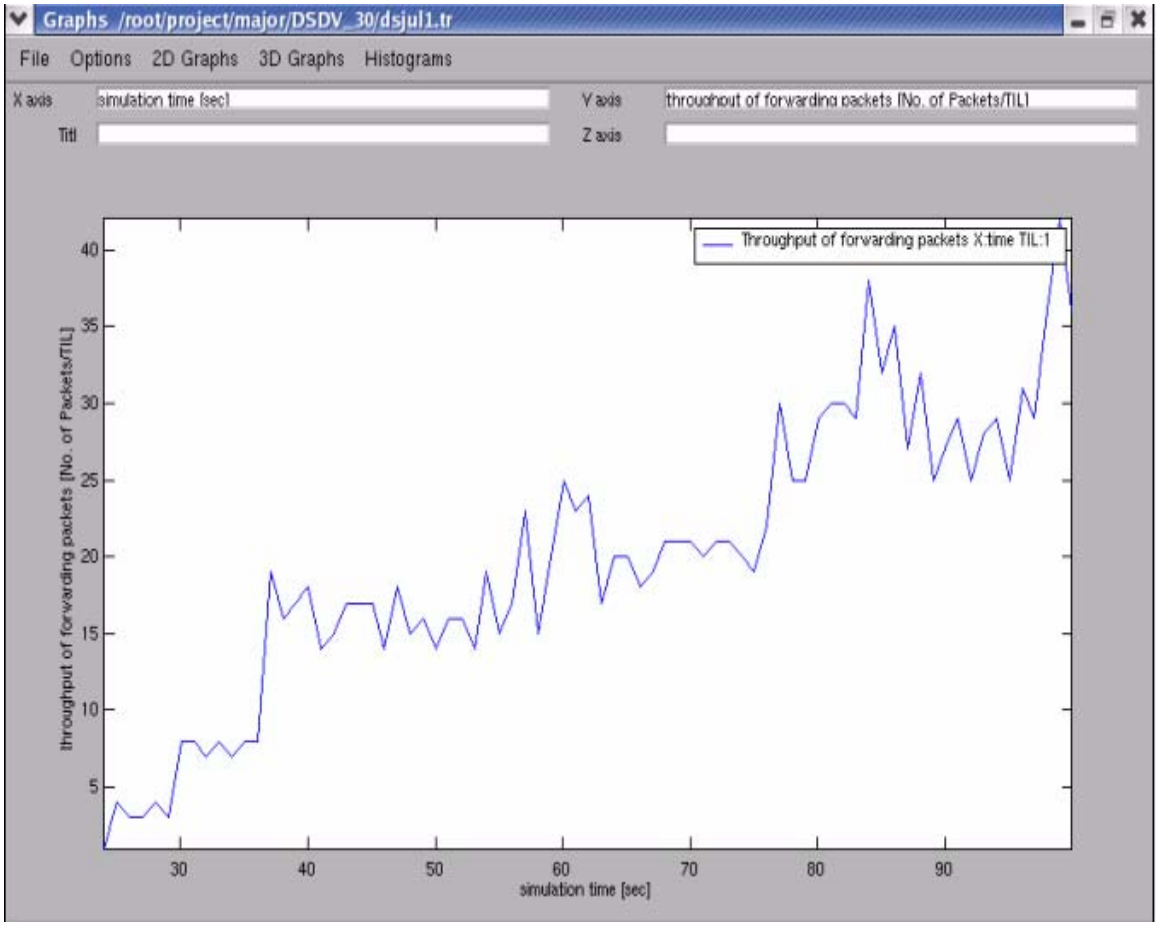

Figure 6.3.9 : Cumulative sum of numbers of all forwarding Packets

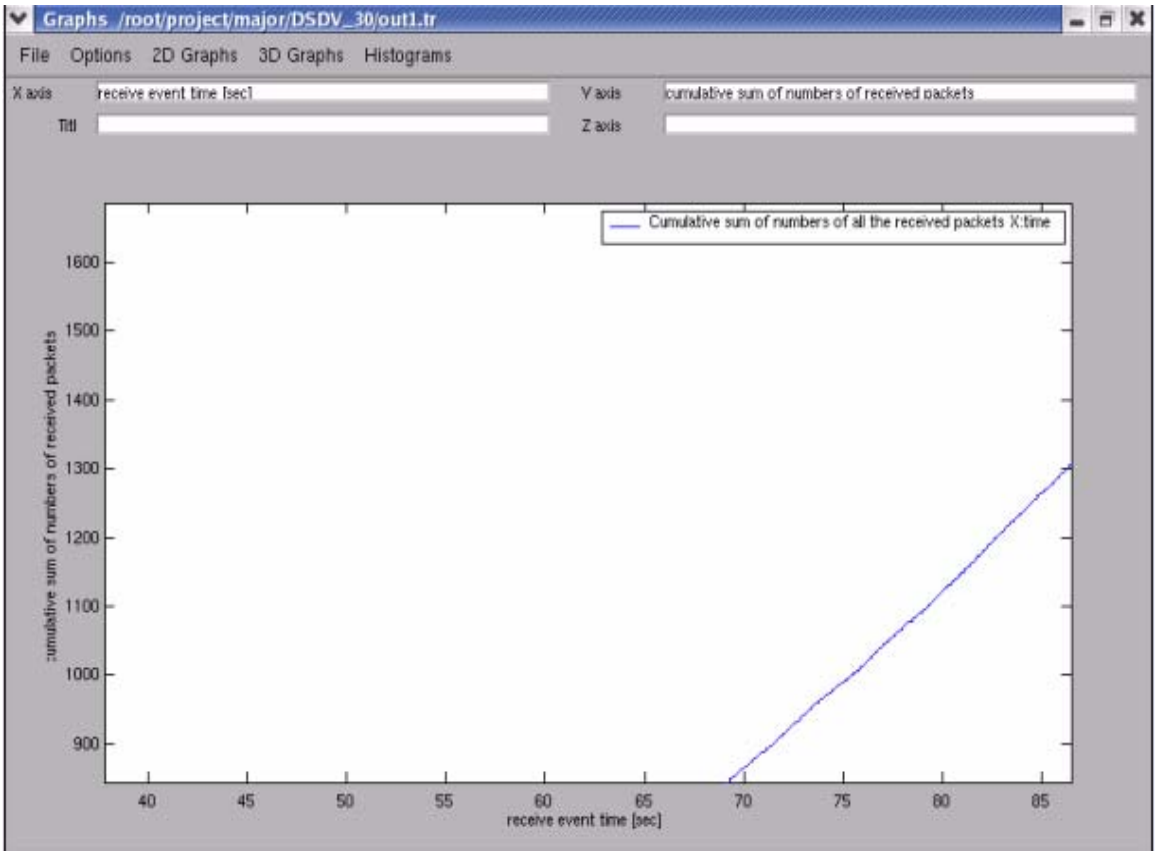

Figure 6.3.10 : Cumulative sum of numbers of all the received Packets

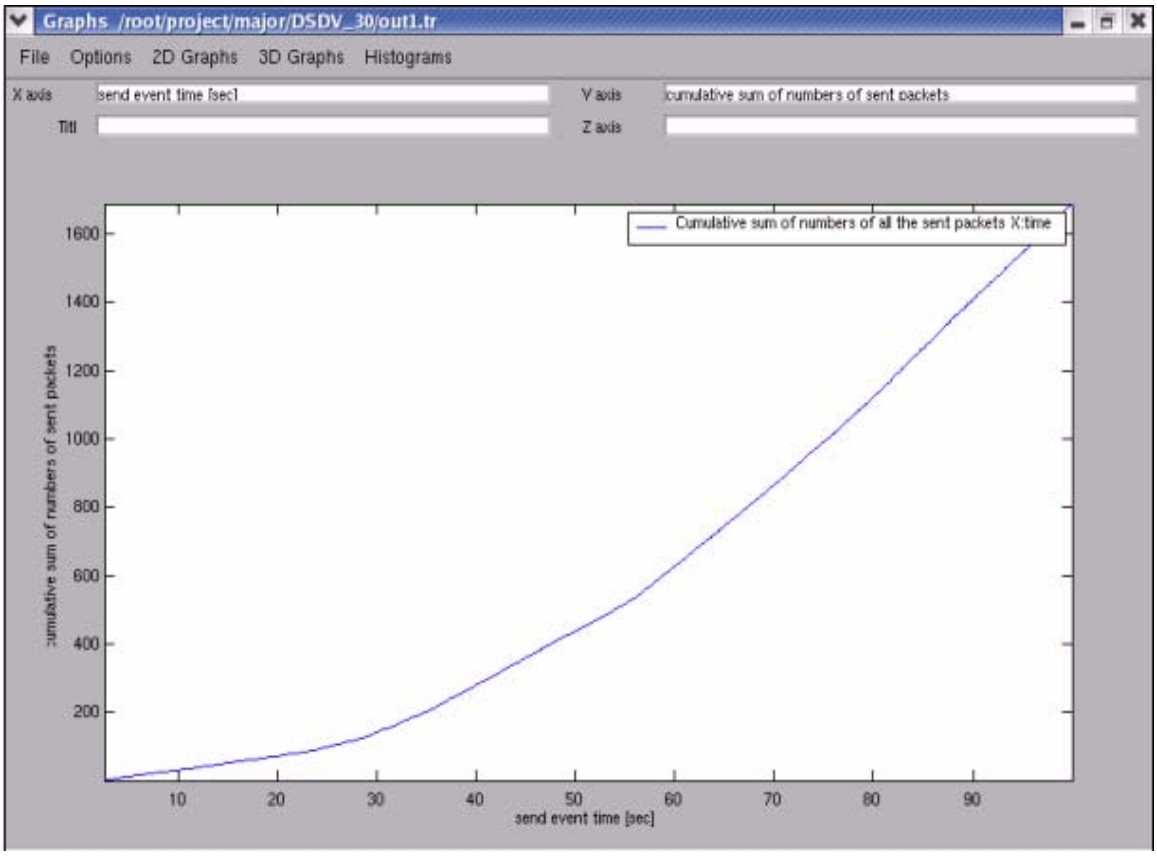

Figure 6.3.11 : Cumulative sum of numbers of all the sent Packets

6.4 Simulation of TORA protocol

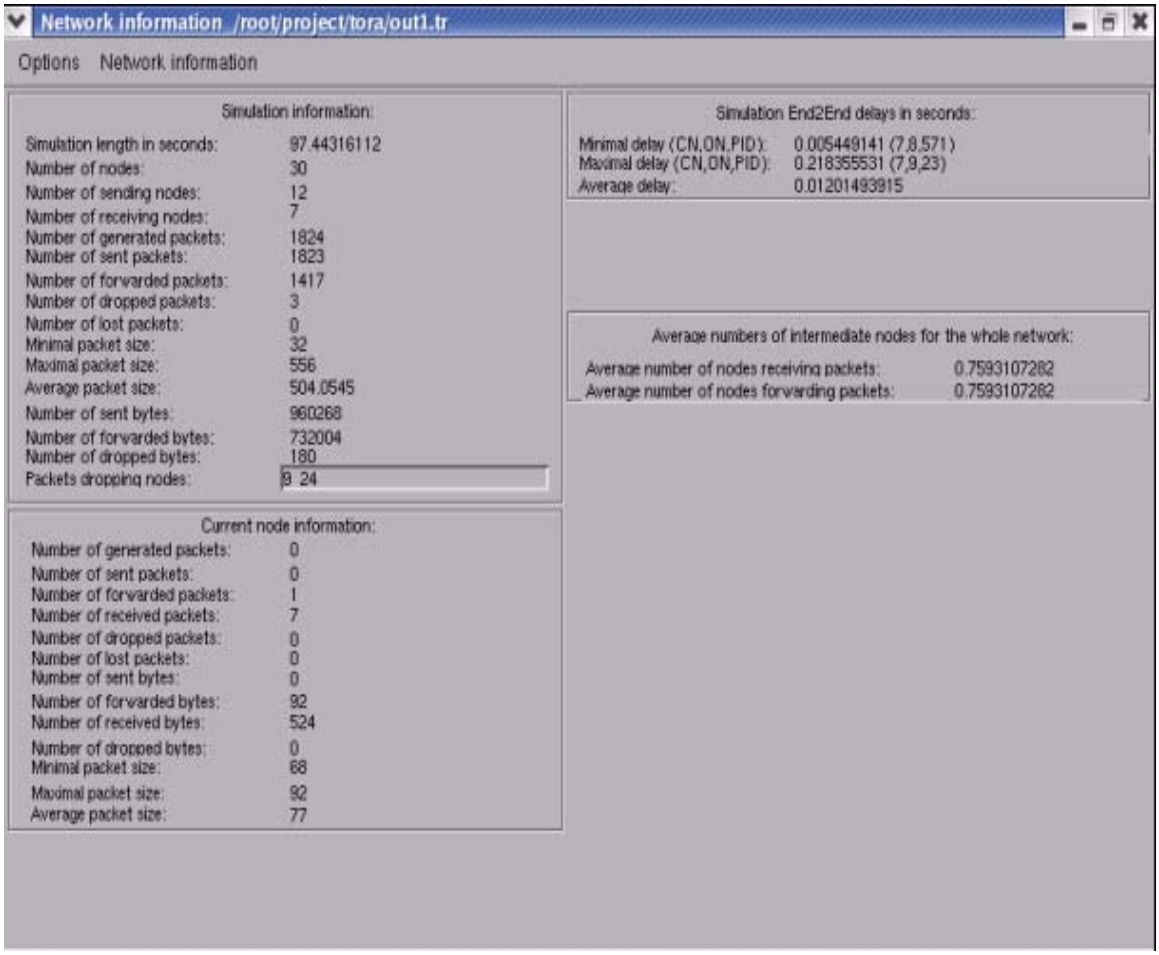

Figure 6.4.1 : Simulation Information

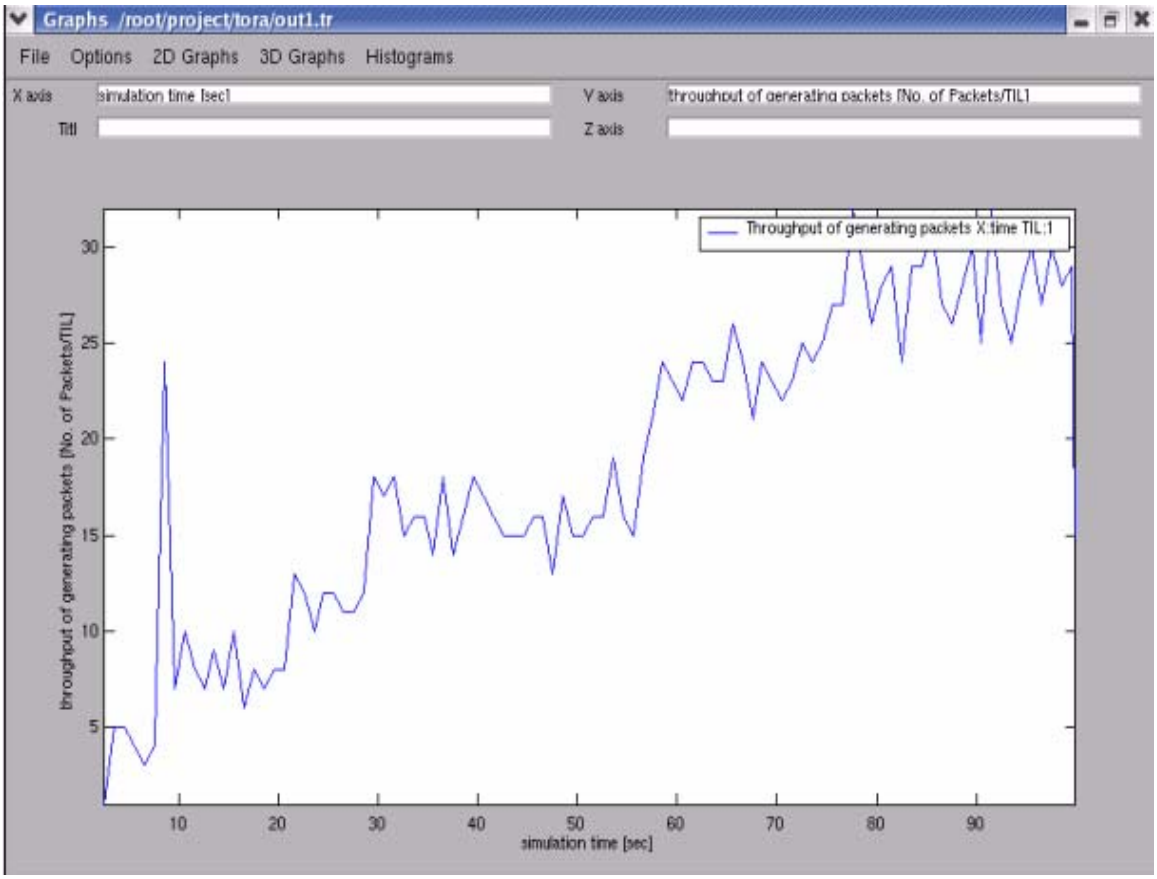

Figure 6.4.2 : Throughput of generating Packets

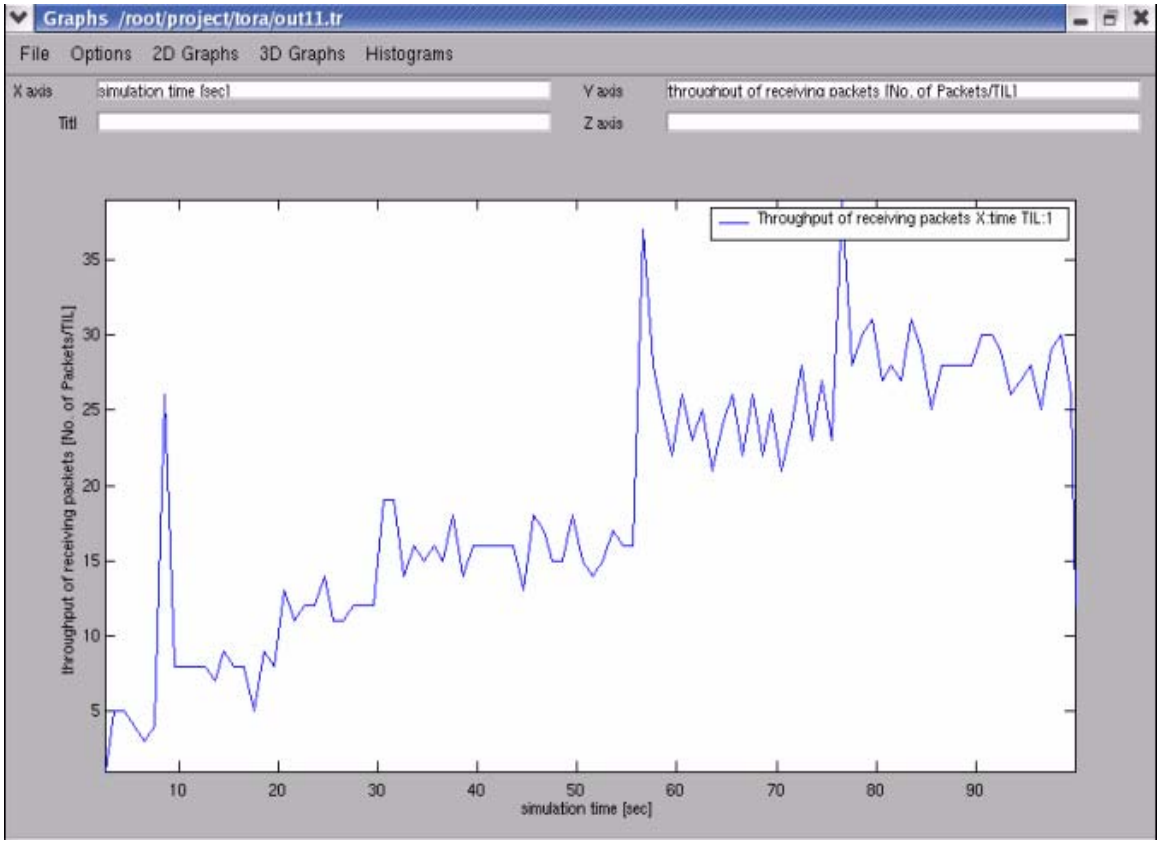

Figure 6.4.3 : Throughput of Receiving Packets

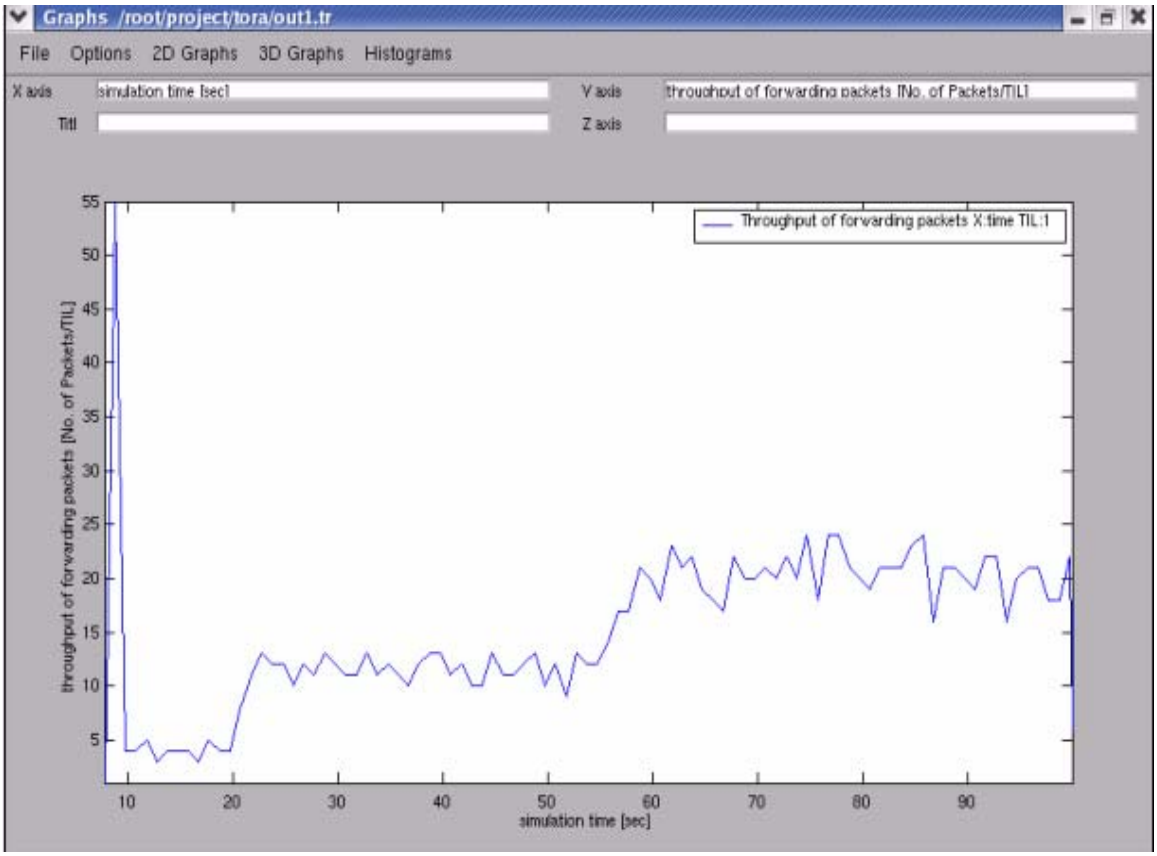

Figure 6.4.4 : Throughput of forwarding Packets

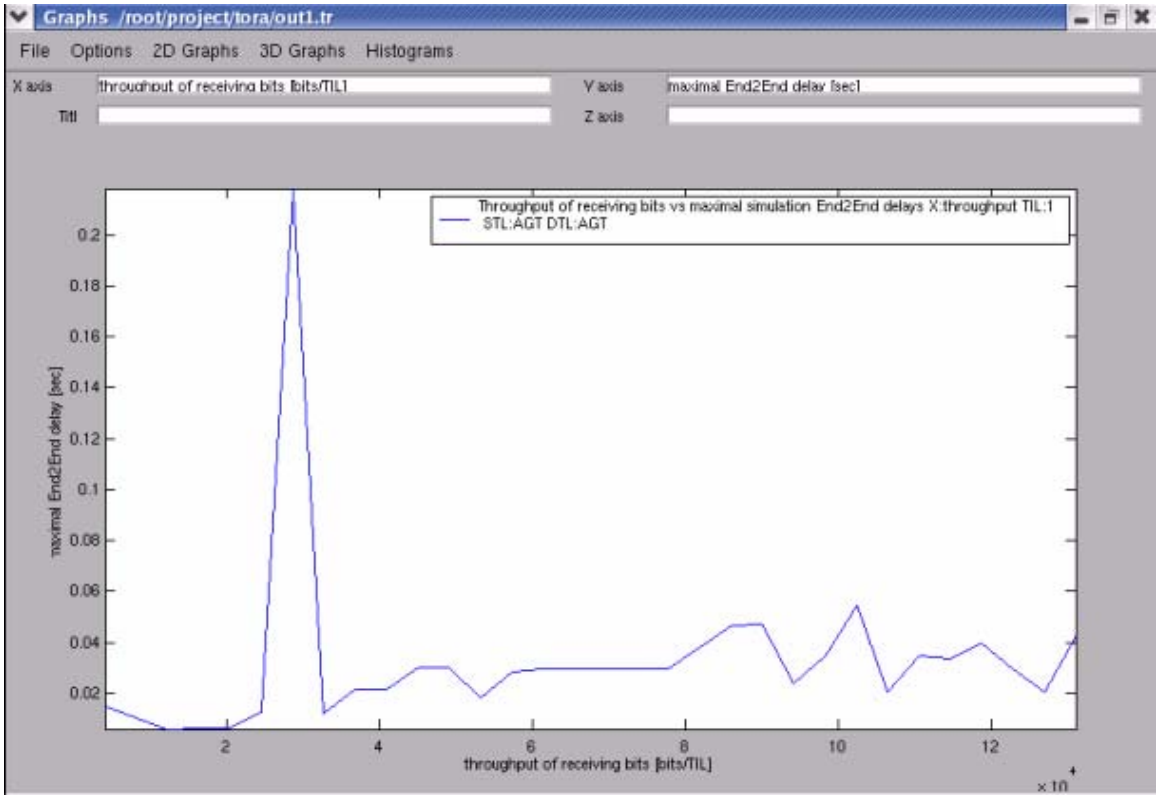

Figure 6.4.5: Throughput of receiving bits Vs maximal simulation End2End Delay

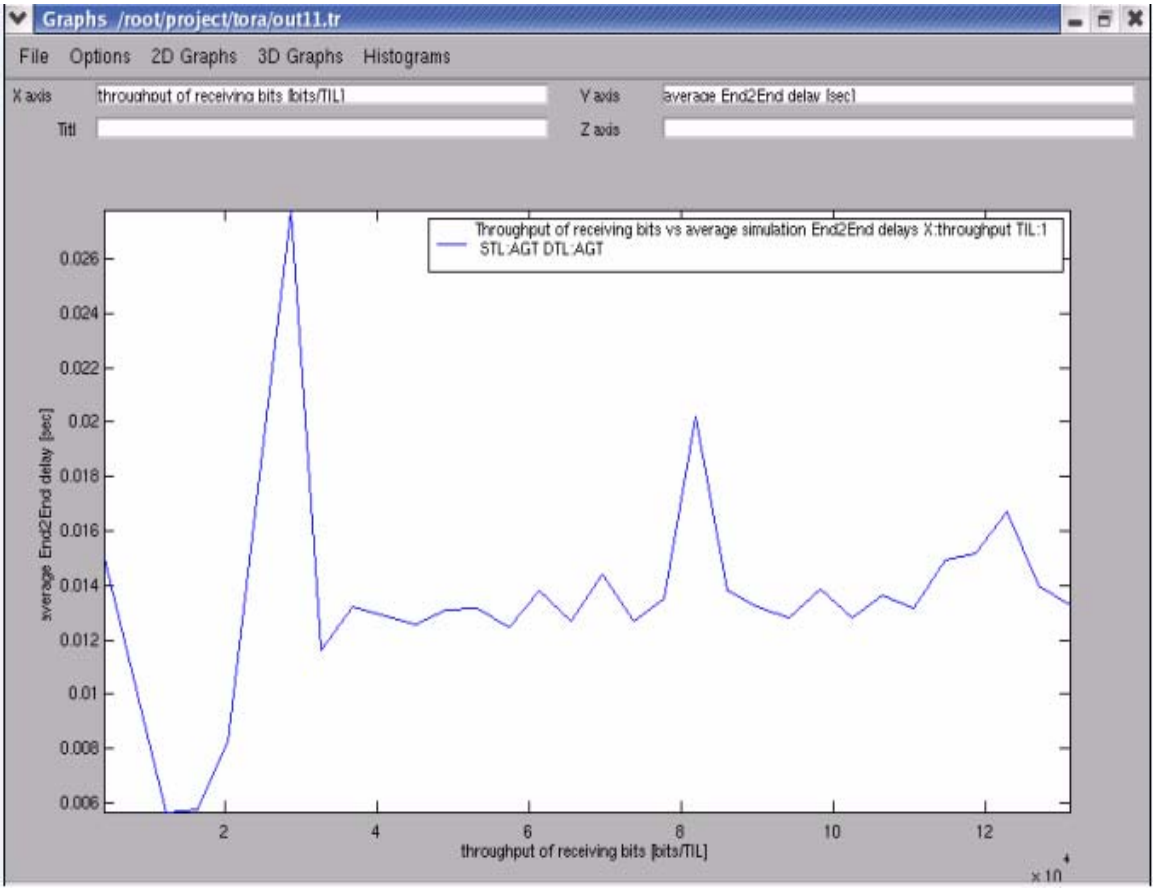

Figure 6.4.6 : Throughput of Receiving bits Vs average simulation End2End Delay

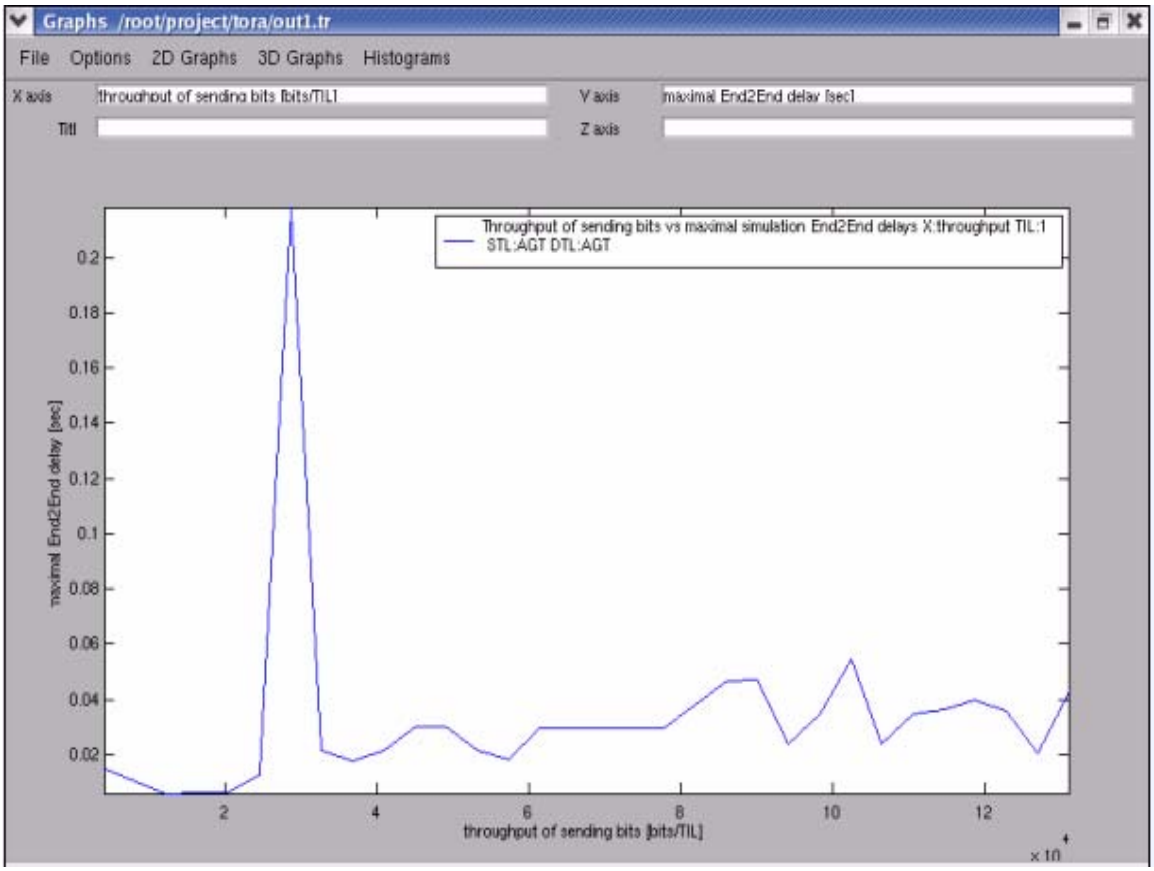

Figure 6.4.7 : Throughput of sending bits Vs maximal simulation End2End Delay

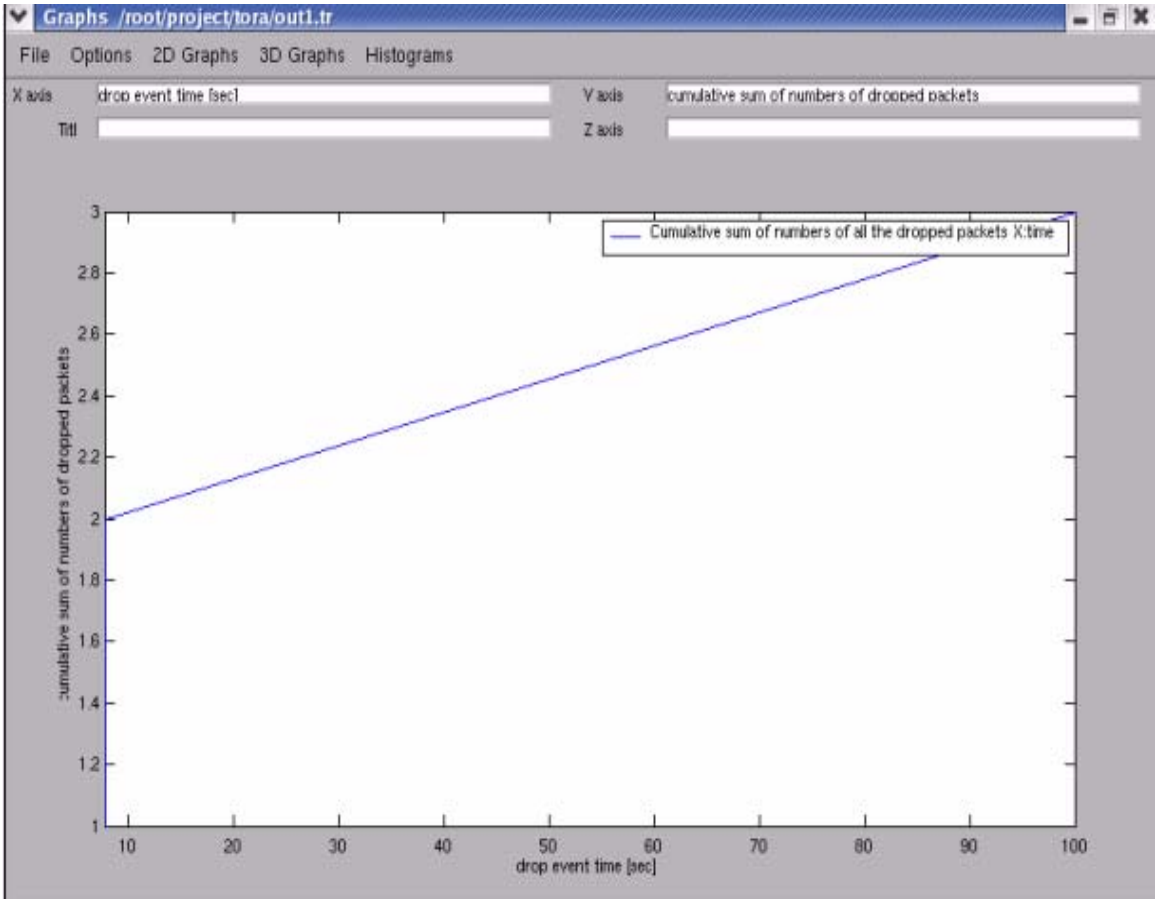

Figure 6.4.8 : Cumulative sum of numbers of all the dropped Packets

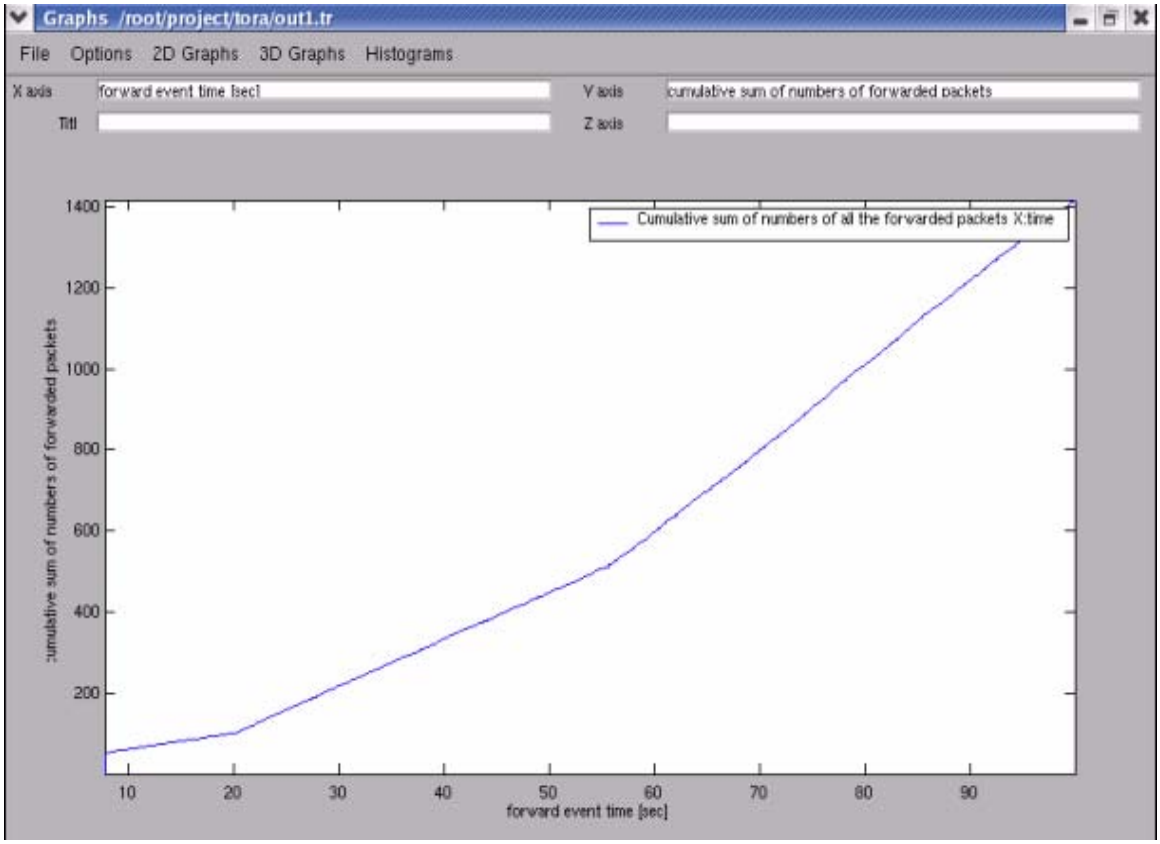

Figure 6.4.9 : Cumulative sum of numbers of all the forward Packets

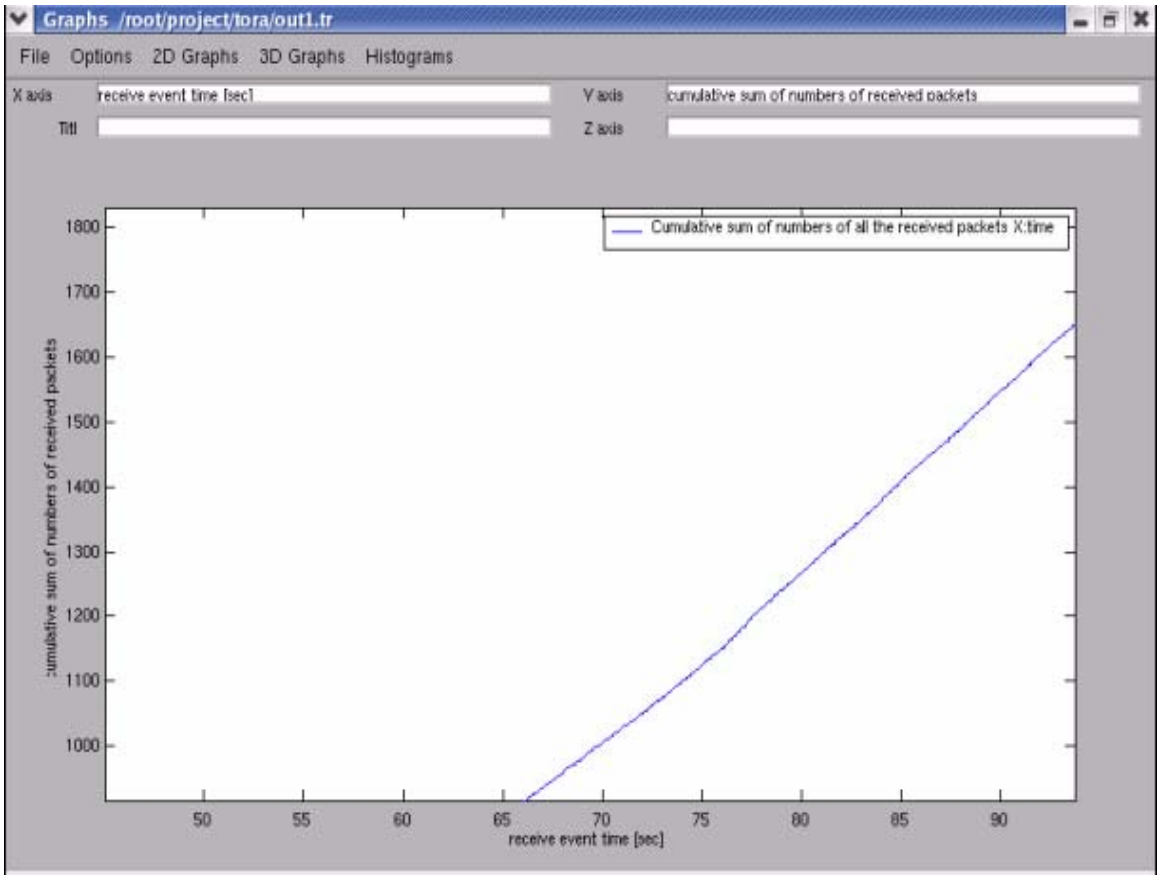

Figure 6.4.10 : Cumulative sum of numbers of the received Packets

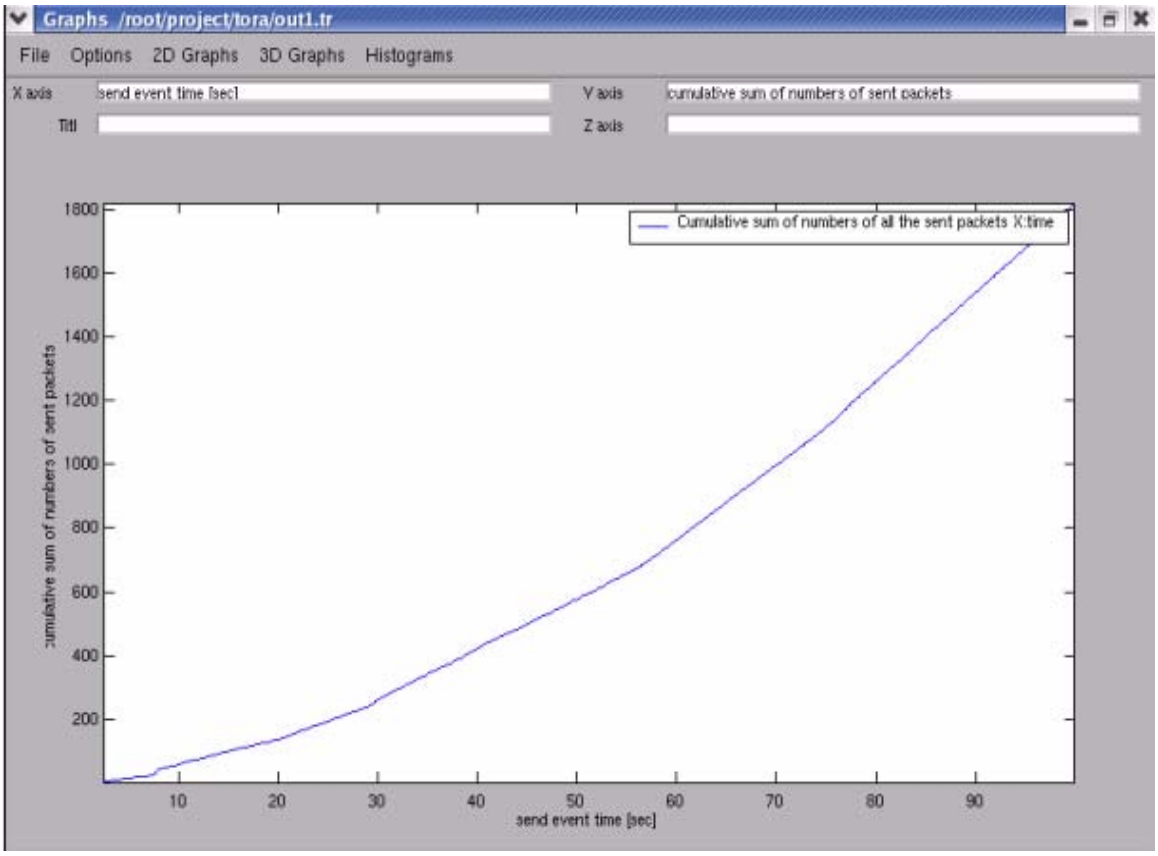

Figure 6.4.11 : Cumulative sum of numbers of the sent Packets

6.5 Comparison of routing Protocol

All of the protocols deliver a greater percentage of originated data packets when there is little node mobility (i.e. at large pause time), converging to 100% delivery when there is no node motion .

6.5.1 Packet Delivery Fraction Vs Pause time

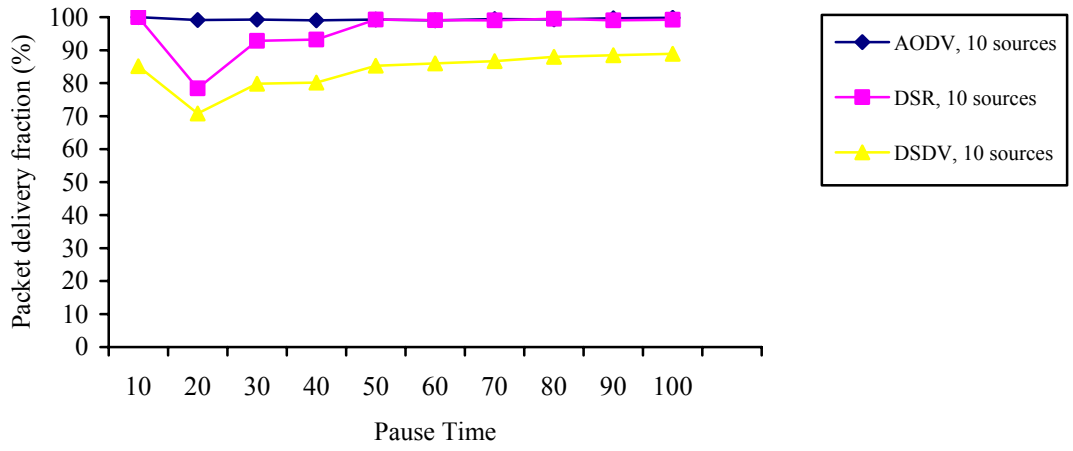

Figure 6.5.1.1

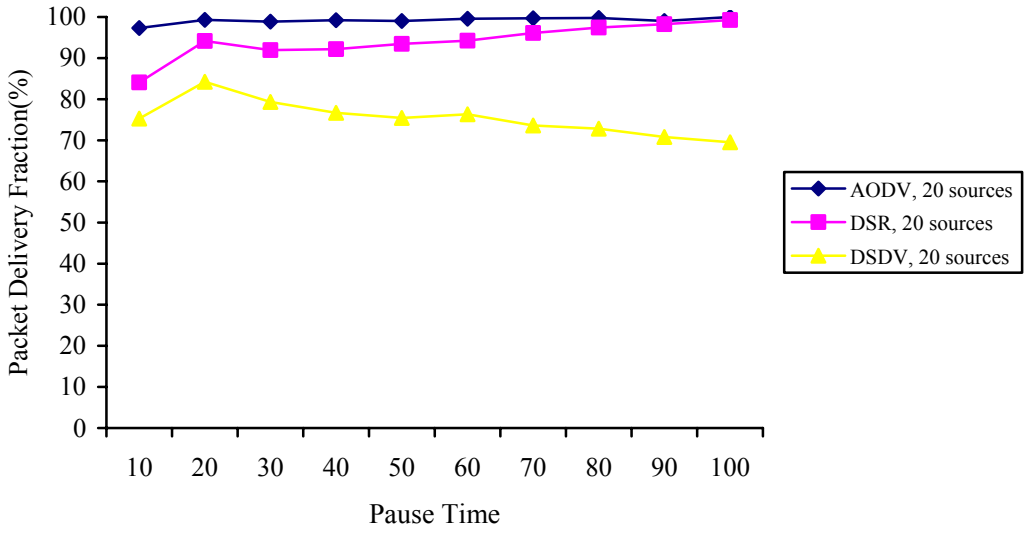

Figure 6.5.1.2

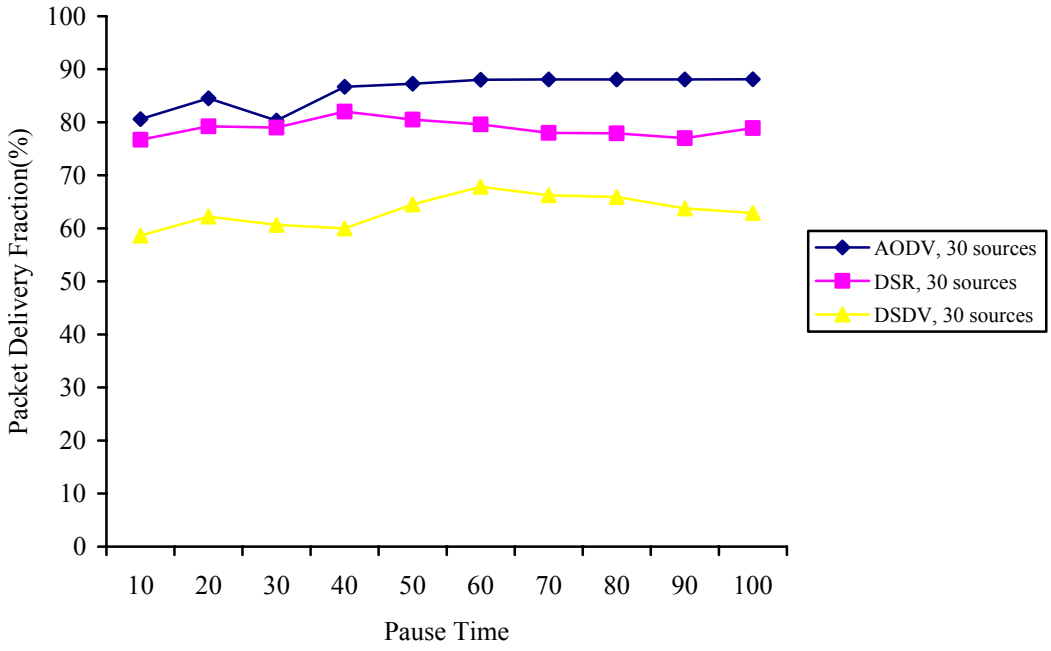

Figure 6.5.1.3

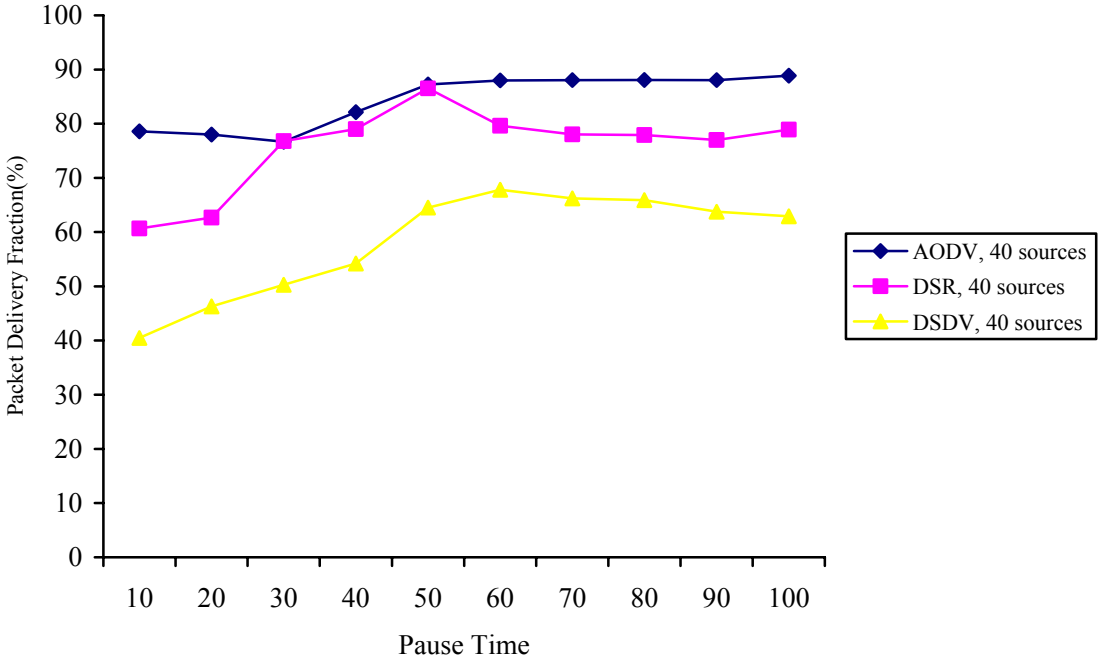

Figure 6.5.1.4

From these graph we see that, Packet delivery fraction are quite similar for AODV and DSR. The value ranges from 85 to 100%. With large amount of pause time and number of sources , AODV performs slightly better than DSR and DSDV. In case of DSDV, packet delivery fraction ranges from 40 to 80%. As the number of sources increases the packet delivery fraction decreases. The reason for the low packet delivery ratio is the fact that DSDV protocols store only one route to each destination and if the route is broken for some reason the node has no route to that destination and therefore all packets for that destination have to dropped until new route is received.

6.5.2 Normalized Load

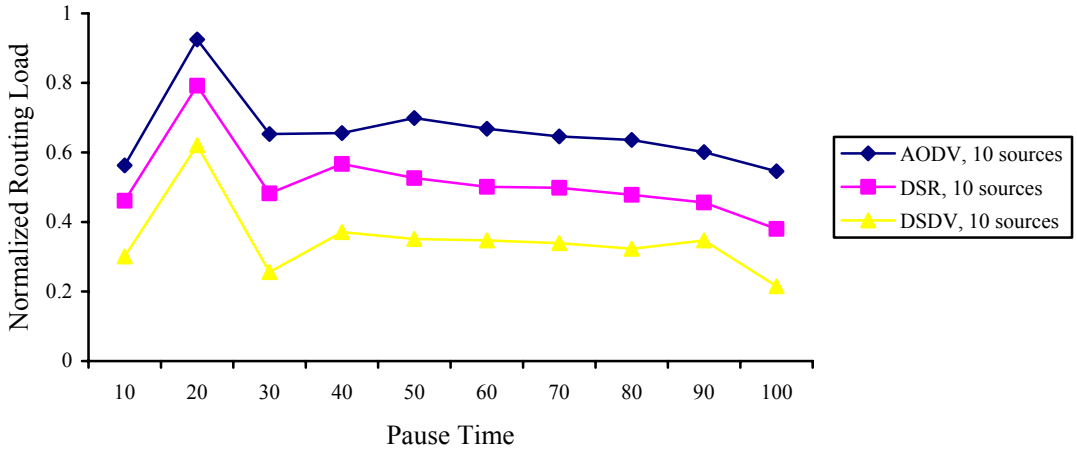

Figure 6.5.2.1

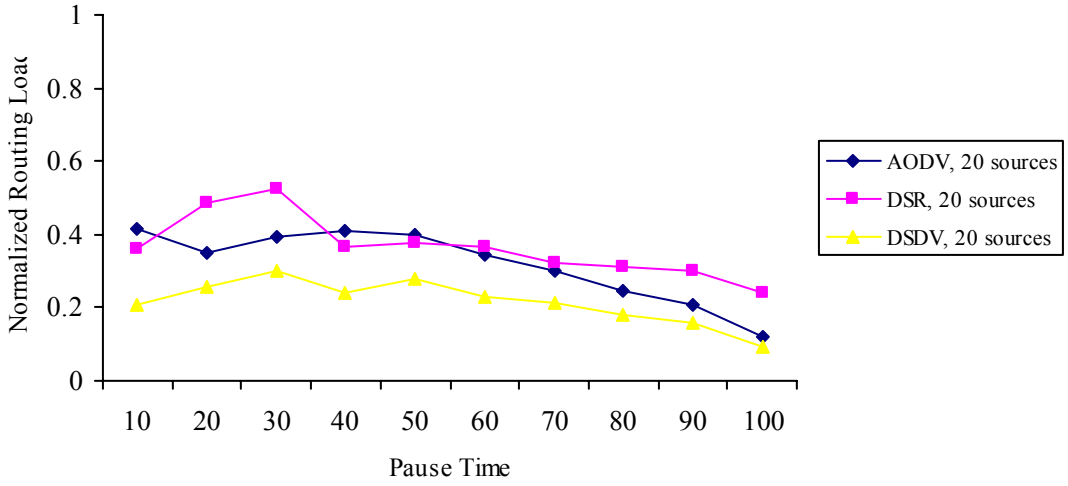

Figure 6.5.2.2

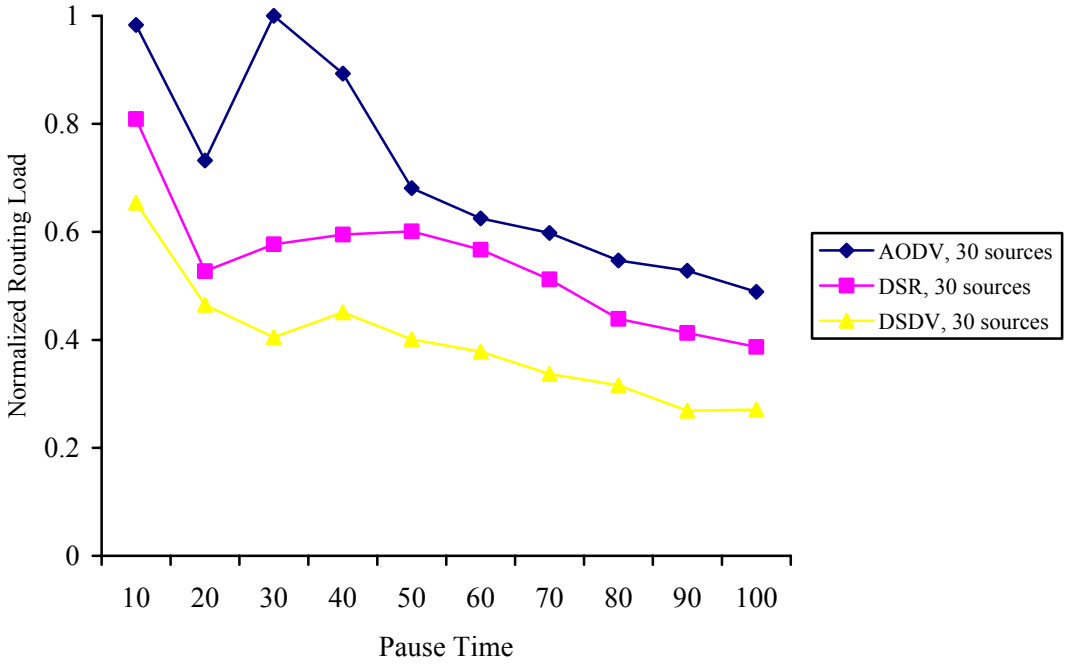

Figure 6.5.2.3

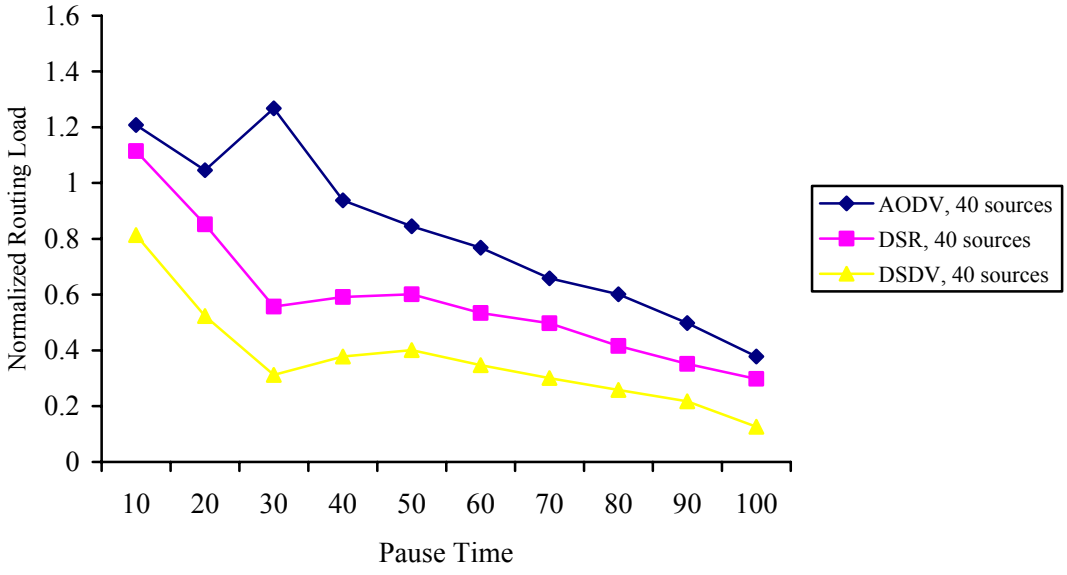

Figure 6.5.2.4

In all cases, DSDV perform well in comparison of DSR and AODV protocols . DSDV perform least Normalized routing load with increase in number of sources. AODV seems to have really high Normalized routing load because of the flooding of route requests throughout the network whenever a node needs a route to a destination. DSR avoids flooding of route discovery packets to some extent by sending the initial route discovery packet with a TTL of 1 so that only the adjacent nodes get the request. If this fails, only then does it resort to flooding the route discovery packets. Another reason for the high Normalized routing load of AODV is that AODV doesn't cache multiple routes to a destination like DSR does and has to resort to route discovery more often.

6.5.3 Effects of Increasing Bit Rate

In this section, we will analyze and compare the protocols based on the effects of changing the Bit-Rate (KBps). As shown in figure 6.5.3.1, the throughput decreased in all protocols with increased bit-rate. The degradation is largest in AODV.

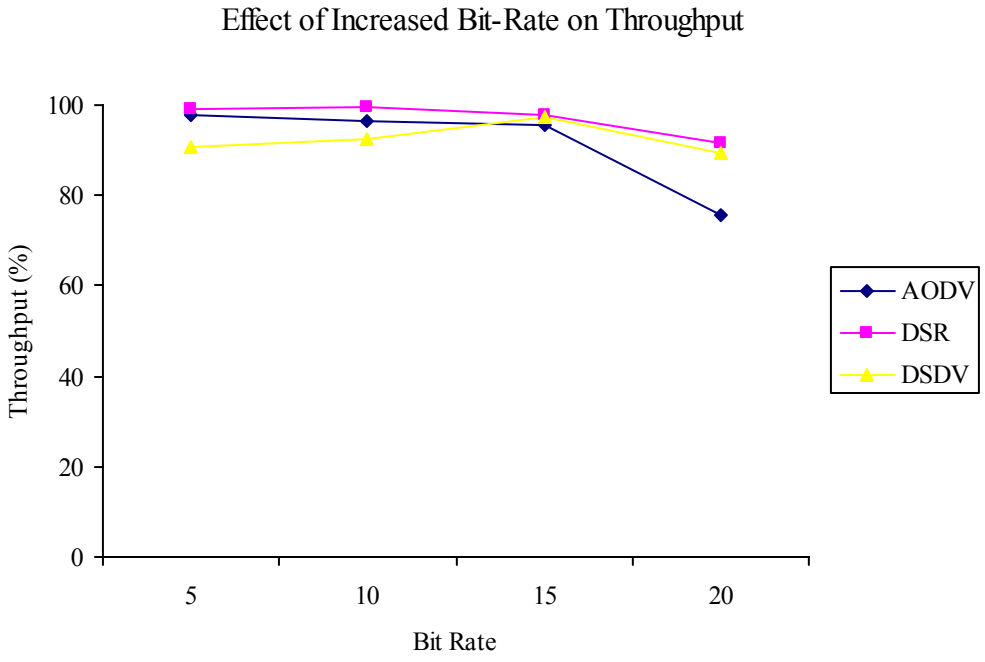

Figure 6.5.3.1

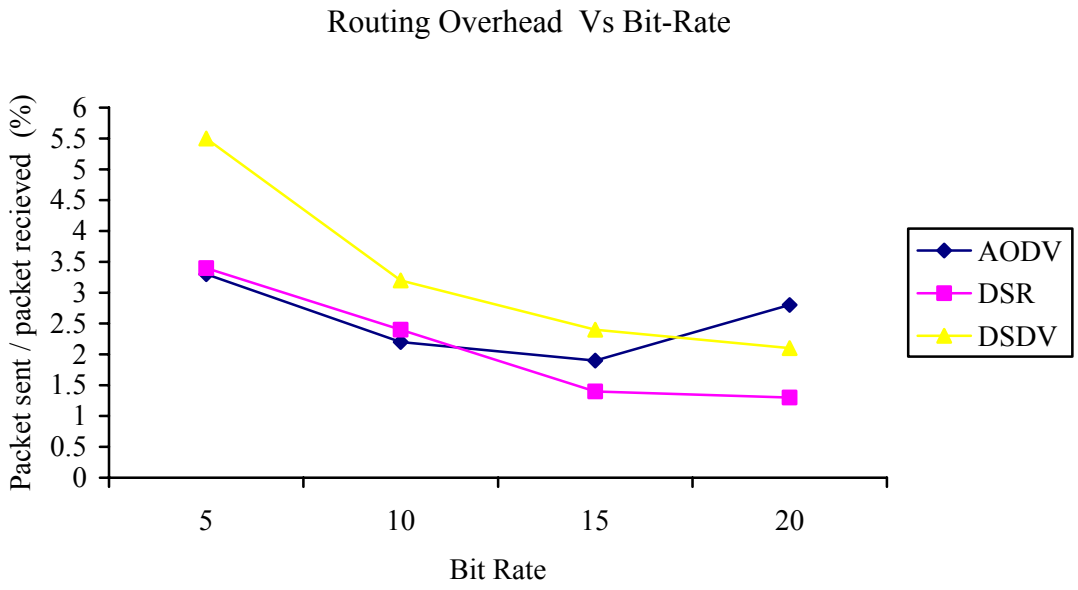

Figure 6.5.3.2

We could expect the routing overhead to increase as the bit-rate increases. From figure, 6.5.3.2. The routing overhead actually decreases as the bit-rate increases. This is because as the bit-rate is increased, for every router packet sent the number of data packets that is received increases. Only one route discovery packet is used for finding out the route from more data packets and the ratio decreases. DSR perform least Routing Overhead with increase in number of bit-rates. In above case, routing overhead of AODV increases as the bit-rate increases.

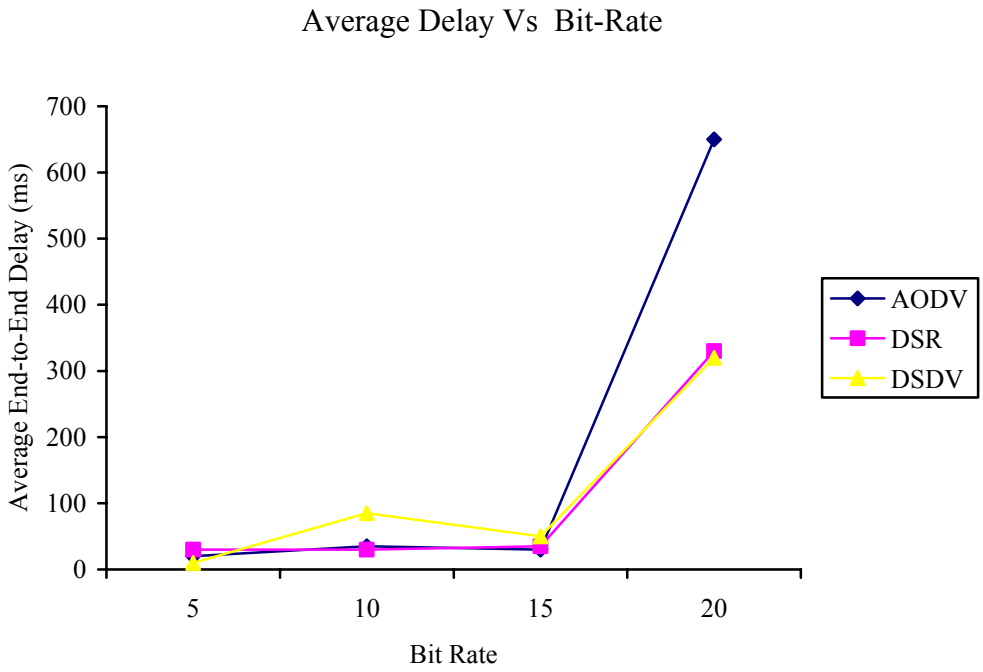

Figure 6.5.3.3

Figure 6.5.3.3, shows that as the bit-rate increases the average end-to-end delay also increases, i.e. the time taken for the packets to reach the destination increases. The increase in delay for AODV is more than that in DSR and DSDV. Thus it takes more time to establish a route in AODV.

Chapter 7

Conclusion and Future Work

In this thesis, we actually targeted two aims, one of them was to identify the various techniques used for classification of various routing protocols in wireless Ad-hoc networks and second goal was to perform the simulation of routing protocols using Network Simulator (NS2) tool, which is a powerful and one of the most reputable tools for simulating network protocols, wired or wireless network and performing the comparison based on their throughput, routing overhead and average end-to-end delay, data packet delivery fraction under a wide range of simulated conditions. We have changed pause time, number of sources and Bit-Rate.

We simulated all four routing protocols (AODV, DSR, DSDV, TORA) in ad hoc network of 30 mobile nodes about and communicating with each other and presents the simulation results. During the simulation of protocols, we observed that, DSR (Dynamic Source Routing) protocol was quite successful under various trading routing overhead or average delay and throughput. DSDV did not increase the number of routing packets and this caused degradation in throughput. AODV managed throughputs as high as DSR, but with higher routing overhead and delay. DSR , however, consistently generates less routing load than AODV. The poor delay and throughput performances of DSR are mainly attributed to aggressive use of caching, and lack of any mechanism to expire stale routes or to determine the freshness of routes when multiple choices are available. Aggressive caching, however, seems to help DSR at low loads and also keeps its routing load down. We believe that mechanisms to expire routes and/or determine freshness of routes, will benefit DSR's performance significantly. On the other hand, AODV's routing loads can be reduced considerably by source routing the request and reply packets in the route discovery process. Since AODV keeps track of actively used routes, multiple actively used destinations also can be searched using a single route discovery flood to control routing load. The simulation results can summaries the suitability of each of the protocol as follows: DSR is suitable for networks in which the mobiles move at moderate speed. It had lowest routing. DSDV is most suitable for small networks where changes in the topology are limited. Also DSDV could be considered for delay constraint networks.

TORA is suitable for operation in large highly dynamic mobile network environment with dense population of nodes. The main advantage of TORA is its support for multiple routes and multicasting. Thus TORA often serve as the underlying protocol for lightweight adaptive multicast algorithms. AODV has the advantages of both of them.

We might conclude that the different behavior of any particular protocol is the reason why none of them was yet declared to be the future standard. Therefore we think it is necessary to make the choice of the protocol depending on the network structure (e.g. number of nodes, types of traffic, the speed of movement of the nodes) and environment.

7.1 Future Work

In this thesis, we have covered few routing protocols in wireless ad hoc network for classification. There are still many routing protocols that can be studied and classified on the basis of various criteria's covered in this thesis. After choosing ns-2 we decided to concentrate on the ad hoc network protocols that have been implemented for it namely DSDV, DSR, AODV and TORA. We couldn't carry out the full set of simulations for ZRP, as the most of the runs didn't complete. Therefore for this thesis we restricted ourselves to AODV, DSDV, DSR and TORA.

We plan to continue with our simulations for AODV, DSR, DSDV and TORA. We would like to study the effects of changing other parameters like, the number of mobile nodes, the dimensions of simulation space, other traffic than CBR (e.g., TCP transfers) and the speed of movement of the nodes and collect other metrics if possible. DSDV shows least packet delivery ratio, since it maintains a single node for each destination. We can increase the packet delivery ratio by maintaining the multiple routes to each destination. It may cause in increase in the size of the routing update packets but will definitely improve the packet delivery ratio. We can also explore the study for finding the reasons for packet drop by exploring packet traces . We can also repeat each test with a large number of scenarios to remove the randomness if any from the results.

BIBLIOGRAPHY

[1] L. Tao. Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications. *Computer Engineering, Virginia Polytechnic Institute and State University*, 2004.

[2] C. E. Perkins. *Ad Hoc Networking*. Addison-Wesley, ISBN 0-201-30976-9, 2001.

[3] A. L. Murphy, G. C. Roman, G. Varghese. An Exercise in Formal Reasoning about Mobile Communications. *Proceedings of the Ninth International Workshop on Software Specifications and Design*, IEEE Computer Society Technical Council on Software Engineering, Japan, pages 25-33, 1998.

[4] J. Kardash. Bluetooth Architecture Overview. *Intel Technology Journal Q2*, Mobile Computing Group, Intel Corporation, 2000.

[5] N. Nikaein, S. Wu, C. Bonnet, and H. Labiod. Designing Routing Protocol For Mobile Ad Hoc Networks, 2001.

[6] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Comp. Commun. Rev.*, Oct. 1994, pp. 234–44.

[7] L. R. Ford Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton Univ. Press, 1962.

[8] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps.*, Feb. 1999, pp. 90–100.

[9] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," *Mobile Computing*, T. Imielinski and H. Korth, Eds., Kluwer, 1996, pp. 153–81.

[10] J. Broch, D. B. Johnson, and D. A. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” IETF Internet draft, draft-ietfmanet-dsr-01.txt, Dec. 1998 (work in progress).

[11] V. D. Park and M. S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” *Proc. INFOCOM ’97*, Apr. 1997.

[12] M. S. Corson and A. Ephremides, “A Distributed Routing Algorithm for Mobile Wireless Networks,” *ACM/Baltzer Wireless Networks J.*, vol. 1, no. 1, Feb. 1995, pp. 61–81

[14] M. Joa-Ng and I. T. Lu. A Peer-To-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks. *IEEE on Selected Areas in Communications*, vol. 17, no. 8, pages 1415–1425, 1999.

[13] C. K. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, ISBN: 0130078174, 2002.

[16] J. Yoon, M. Liu, and B. Noble, “Random waypoint considered harmful,” in *IEEE INFOCOMM 03*, San Francisco, USA, Mar. 2003.

[17] S. McCanne and S. Floyd. (2003, December) ns2 network simulator 2. [Online]. Available: http://www.isi.edu/nsnam/ns

[18] K. Fall, K. Varadhan, and the VINT project. (2003, December) The ns manual. [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html

[19] S. Das, C. Perkins and E. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks"

[20] http://www.ietf.org/internet-drafts/ draft-ietf-manet-appl-00.txt

[21] http://www.ietf.org/internet-drafts/ draft-ietf-manet-aodv-02.txt

[22] http://www.ietf.org/internet-drafts/ draft-ietf-manet-dsr-00.txt

[23] http://www.ietf.org/internet-drafts/ draft-ietf-manet-dsdv-00.txt

[24] http://www.ietf.org/internet-drafts/ draft-ietf-manet-tora-spec-01.txt

Following is a sample TCL script, which is being used for the simulation AODV protocols for 30 nodes.

AODV_30.tcl

```
# ======================================================================
# Default Script Options
# ======================================================================
set opt(chan)           Channel/WirelessChannel
set opt(prop)           Propagation/TwoRayGround
set opt(netif)          Phy/WirelessPhy
set opt(mac)            Mac/802_11
set opt(ifq)            Queue/DropTail/PriQueue
set opt(ll)             LL
set opt(ant)            Antenna/OmniAntenna
set opt(x)              670   ;          # X dimension of the topography
set opt(y)              670              ;# Y dimension of the topography
set opt(ifqlen)         50               ;# max packet in ifq
set opt(seed)           20.0
set opt(tr)             aodv10.tr        ;# trace file
set opt(adhocRouting)   AODV
set opt(nn)             30               ;# how many nodes are simulated
set opt(cp)             "cbr-30-test"
set opt(sc)             "scen-30-test10"
set opt(stop)           100.0            ;# simulation time

# ======================================================================

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

# Initialize Global Variables
# create simulator instance
set ns_         [new Simulator]

# set wireless channel, radio-model and topography objects
set wtopo       [new Topography]

# create trace object for ns and nam
set tracefd     [open $opt(tr) w]
$ns_ trace-all $tracefd
```

```
# use new trace file format
$ns_ use-newtrace

# define topology
$wtopo load_flatgrid $opt(x) $opt(y)

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

# define how node should be created
#global node setting
$ns_ node-config -adhocRouting $opt(adhocRouting) \
                 -llType $opt(ll) \
                 -macType $opt(mac) \
                 -ifqType $opt(ifq) \
                 -ifqLen $opt(ifqlen) \
                 -antType $opt(ant) \
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                 -channelType $opt(chan) \
                 -topoInstance $wtopo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF


#  Create the specified number of nodes [$opt(nn)] and "attach" them
#  to the channel.
for {set i 0} {$i < $opt(nn) } {incr i} {
        set node_($i) [$ns_ node]
        $node_($i) random-motion 0              ;# disable random motion
}
# Define node movement model
puts "Loading connection pattern..."
source $opt(cp)

# Define traffic model
puts "Loading scenario file..."
source $opt(sc)
# Define node initial position in nam
for {set i 0} {$i < $opt(nn)} {incr i} {

 # 30 defines the node size in nam, must adjust it according to your scenario
  # The function must be called after mobility model is defined
  $ns_ initial_node_pos $node_($i) 30
}
```

```
# Tell nodes when the simulation ends
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop).000000001 "$node_($i) reset";
}

# tell nam the simulation stop time
#$ns_ at  $opt(stop)    "$ns_ nam-end-wireless $opt(stop)"
$ns_ at  $opt(stop).000000001 "puts \"NS EXITING...\" ; $ns_ halt"
puts "Starting Simulation..."
$ns_ run
```

Following is a sample TCL script, which is being used for the simulation of DSDV, DSR and TORA protocols for 30 nodes.

Sample_30.tcl

```
# ======================================================================
# Default Script Options
# ======================================================================
set opt(chan)           Channel/WirelessChannel
set opt(prop)           Propagation/TwoRayGround
set opt(netif)          Phy/WirelessPhy
set opt(mac)            Mac/802_11
set opt(ifq)            Queue/DropTail/PriQueue
set opt(ll)             LL
set opt(ant)            Antenna/OmniAntenna

set opt(x)              670          ;# X dimension of the topography
set opt(y)              670          ;# Y dimension of the topography
set opt(cp)             "cbr-30-test"
set opt(sc)             "scen-30-test10"

set opt(ifqlen)         50           ;# max packet in ifq
set opt(nn)             30           ;# number of nodes
set opt(seed)           20.0
set opt(stop)           100.0        ;# simulation time
set opt(tr)             out10.tr              ;# trace file
set opt(rp)             dsdv         ;# routing protocol script (dsr or
dsdv)
set opt(lm)             "off"        ;# log movement

# ======================================================================

set AgentTrace          ON
set RouterTrace         ON
set MacTrace            OFF

LL set mindelay_        50us
LL set delay_           25us
LL set bandwidth_       0

Agent/Null set sport_   0
Agent/Null set dport_   0

Agent/CBR set sport_    0
```

```
Agent/CBR set dport_              0

Agent/TCPSink set sport_          0
Agent/TCPSink set dport_          0

Agent/TCP set sport_              0
Agent/TCP set dport_              0
Agent/TCP set packetSize_         512

Queue/DropTail/PriQueue set Prefer_Routing_Protocols    1

# unity gain, omni-directional antennas
# set up the antennas to be centered in the node and 1.5 meters above it
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0

# ======================================================================

proc usage { argv0 }  {
        puts "Usage: $argv0"
        puts "\tmandatory arguments:"
        puts "\t\t\[-x MAXX\] \[-y MAXY\]"
        puts "\toptional arguments:"
        puts "\t\t\[-cp conn pattern\] \[-sc scenario\] \[-nn nodes\]"
        puts "\t\t\[-seed seed\] \[-stop sec\] \[-tr tracefile\]\n"
}

proc getopt {argc argv} {
        global opt
        lappend optlist cp nn seed sc stop tr x y

        for {set i 0} {$i < $argc} {incr i} {
                set arg [lindex $argv $i]
```

```
                if {[string range $arg 0 0] != "-"} continue

                set name [string range $arg 1 end]
                set opt($name) [lindex $argv [expr $i+1]]
        }
}

proc log-movement {} {
   global logtimer ns_ ns

   set ns $ns_
   source ../mobility/timer.tcl
   Class LogTimer -superclass Timer
   LogTimer instproc timeout {} {
        global opt node_;
        for {set i 0} {$i < $opt(nn)} {incr i} {
           $node_($i) log-movement
        }
        $self sched 0.1
   }

   set logtimer [new LogTimer]
   $logtimer sched 0.1
}

# ======================================================================
# Main Program
# ======================================================================
getopt $argc $argv

source /root/ns-allinone-2.27/ns-2.27/tcl/lib/ns-bsnode.tcl
source /root/ns-allinone-2.27/ns-2.27/tcl/mobility/com.tcl

# do the get opt again incase the routing protocol file added some more
# options to look for
getopt $argc $argv

if { $opt(x) == 0 || $opt(y) == 0 } {
        usage $argv0
        exit 1
}

if {$opt(seed) > 0} {
        puts "Seeding Random number generator with $opt(seed)\n"
        ns-random $opt(seed)
}
```

```
#
# Initialize Global Variables
#
set ns_		[new Simulator]
set chan	[new $opt(chan)]
set prop	[new $opt(prop)]
set topo	[new Topography]
set tracefd	[open $opt(tr) w]

$topo load_flatgrid $opt(x) $opt(y)

$prop topography $topo

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

#
# log the mobile nodes movements if desired
#
if { $opt(lm) == "on" } {
   log-movement
}

#
#  Create the specified number of nodes $opt(nn) and "attach" them
#  the channel.
#  Each routing protocol script is expected to have defined a proc
#  create-mobile-node that builds a mobile node and inserts it into the
#  array global $node_($i)
#

if { [string compare $opt(rp) "dsr"] == 0} {
        for {set i 0} {$i < $opt(nn) } {incr i} {
                dsr-create-mobile-node $i
        }
} elseif { [string compare $opt(rp) "dsdv"] == 0} {
        for {set i 0} {$i < $opt(nn) } {incr i} {
                dsdv-create-mobile-node $i
          }
   } elseif { [string compare $opt(rp) "tora"] == 0} {
        for {set i 0} {$i < $opt(nn) } {incr i} {
                tora-create-mobile-node $i
        }
```

```
   }

#
# Source the Connection and Movement scripts
#
if { $opt(cp) == "" } {
        puts "*** NOTE: no connection pattern specified."
       set opt(cp) "none"
} else {
        puts "Loading connection pattern..."
        source $opt(cp)
}


#
# Tell all the nodes when the simulation ends
#
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop).000000001 "$node_($i) reset";
}
$ns_ at $opt(stop).00000001 "puts \"NS EXITING...\" ; $ns_ halt"


if { $opt(sc) == "" } {
        puts "*** NOTE: no scenario file specified."
       set opt(sc) "none"
} else {
        puts "Loading scenario file..."
        source $opt(sc)
        puts "Load complete..."
}

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

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