PERFORMANCE BASED COMPARATIVE ANALYSIS OF AODV, DSR AND DSDV PROTOCOLS w.r.t UDP and TCP

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

Master of Engineering
in
Software Engineering

By:
Darshan Singh Kali Rai
(80731028)

Under the supervision of:
Ashima Singh
Lecturer

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

JULY 2009

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Performance Based Comparative Analysis of AODV, DSR and DSDV Protocols w.r.t UDP and TCP"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Ms. Ashima Singh* and refers other researcher's works which are duly listed in the reference section.

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

(Darshan Singh Kali Rai)

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

(Ashima Singh)
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(RAJESH BHATIA) 4/07/09
Head
Computer Science & Engineering. Department
Thapar University
Patiala

(R.K.SHARMA) 20/7/09
Dean (Academic Affairs)
Thapar University,
Patiala.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my guide **Ms. Ashima Singh,** Lecturer, Computer Science and Engineering Department for immense help, guidance, stimulating suggestions, and encouragement all the time with this thesis work. This work would have not been possible without her encouragement. She always provided a motivating and enthusiastic atmosphere to work with; it was a great pleasure to do this thesis under her supervision.

I am equally grateful to **Ms. Seema Bawa,** Professor and Head, Computer Science and Engineering Department for their appreciation and satisfactorily healing me off my inexperienced inquisitions about the new subject.

I would also like to thank all the staff members and PhD Scholar Ms. Shashi Bhanwar who were always there at the need of the hours and provided with all the help and facilities, which I required for the completion of my thesis.

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

My heartfelt thanks go to my friends especially Arindam Choudhury and Paritosh Sharma who were always there to help me in the tough situations encountered during the course of my thesis and taught me the definition of a true friend.

I am also very thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection which made my stay at Thapar University memorable.

Darshan Singh Kali Rai

(80731028)

Abstract

A Mobile Ad hoc NETwork (MANET) is a kind of wireless ad-hoc network, and is a self configuring network of mobile routers (and associated hosts) connected by wireless links – the union of which forms 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. There are various routing protocols available for MANETs. The most popular ones are DSR, AODV and DSDV.

In this thesis, an attempt has been made to compare these three protocols on the performance basis under different environments. The comparison has been done under two protocols namely UDP and TCP. The tools used for the simulation are NS2 which is the main simulator, NAM (Network Animator) and Tracegraph which is used for preparing the graphs from the trace files.

The results presented in this thesis work clearly indicate that the different protocols behave differently under different environments. The results also illustrate the important characteristics of different protocols based on their performance and thus suggest some improvements in the respective protocols.

Keywords: MANET, AODV, DSR, DSDV, NS2, NAM, UDP, TCP, Tracegraph.

Table of Contents

List of Figures

List of Tables

Chapter 1
INTRODUCTION

1.1 Background

A Mobile Ad hoc NETwork (MANET) is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links – the union of which forms 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.

Wireless networks have become increasingly popular in the network industry. They can provide mobile users with ubiquitous communication capability and information access regardless of locations. Conventional wireless networks are often connected to a wired network so that the ATM (Asynchronous Transfer Mode) or Internet connections can be extended to mobile users. This kind of wireless network requires a fixed wireline backbone infrastructure. All mobile hosts in a communication cell can reach a base station on the wireline network in one-hop radio transmission. In parallel with the conventional wireless networks, another type of model, based on radio to radio multi-hopping, has neither fixed base stations nor a wired backbone infrastructure. In some application environments, such as battlefield communications, disaster recovery etc., the wired network is not available and multi-hop wireless networks provide the only feasible means for communication and information access. This kind of network is called Mobile Ad hoc NETwork (MANET). It is also expected to play an important role in civilian forums such as campus recreation, conferences, and electronic classrooms etc [6].

Mobile hosts and wireless networking hardware are becoming widely available, and extensive work has been done recently in integrating these elements into traditional networks such as the Internet. Often sometimes, mobile users may want to communicate with each other without the barriers of fixed infrastructure like fixed backbones or confined within a certain area. For example, a group of students may

want to communicate with each other to share some lecture notes, assignments etc.; friends or business associates may meet somewhere and may want to share some files; some disaster recovery team may want to setup a network in some emergency to share the details of the situation with each other. In such situations, a temporary network can be setup without a centralized infrastructure. These are some examples where MANETs can be employed efficiently and effectively.

1.2 Issues in MANETs

If there are only two nodes that want to communicate with each other and are located very closely to each other, then no specific routing protocols or routing decisions are necessary. On the other hand, if there are a number of mobile hosts wishing to communicate, then the routing protocols come into play because in this case, some critical decisions have to be made such as which is the optimal route from the source to the destination which is very important because often, the mobile nodes operate on some kind of battery power. Thus it becomes necessary to transfer the data with the minimal delay so as to waste less power. There may also be some kind of compression involved which could be provided by the protocol so as to waste less bandwidth. Further, there is also a need of some type of encryption so as to protect the data from prying eyes. In addition to this, Quality of Service support is also needed so that the least packet drop can be obtained.

The other factors which need to be considered while choosing a protocol for MANETs are as follows:

i. **Multicasting**: This is the ability to send packets to multiple nodes at once. This is similar to broadcasting except the fact that the broadcasting is done to all the nodes in the network. This is important as it takes less time to transfer data to multiple nodes.
ii. **Loop Free**: A path taken by a packet never transits the same intermediate node twice before it arrives at the destination. To improve the overall, we want the routing protocol to guarantee that the routes supplied are loop-free. This avoids any waste of bandwidth or CPU consumption.

iii. **Multiple routes**: If one route gets broken due to some disaster, then the data could be sent through some other route. Thus the protocol should allow creating multiple routes.

iv. **Distributed Operation**: The protocol should of course be distributed. It should not be dependent on a centralized node.

v. **Reactive**: It means that the routes are discovered between a source and destination only when the need arises to send data. Some protocols are reactive while others are proactive which means that the route is discovered to various nodes without waiting for the need.

vi. **Unidirectional Link Support**: The radio environment can cause the formation of unidirectional links. Utilization of these links and not only the bi-directional links improves the routing protocol performance.

vii. **Power Conservation**: The nodes in an ad-hoc network can be laptops and thin clients, such as PDAs that are very limited in battery power and therefore use some sort of stand-by mode to save power. It is therefore important that the routing protocol has support for these sleep-modes [14] [19].

If there are multiple nodes wishing to communicate with each other and one or more of them are beyond the vicinity of the node who wants to send data to them, then that node can send data to the nearest node who in turn can transfer to the next node and in this way, the data can be transferred. In figure 1.1, let's suppose that node A wants to

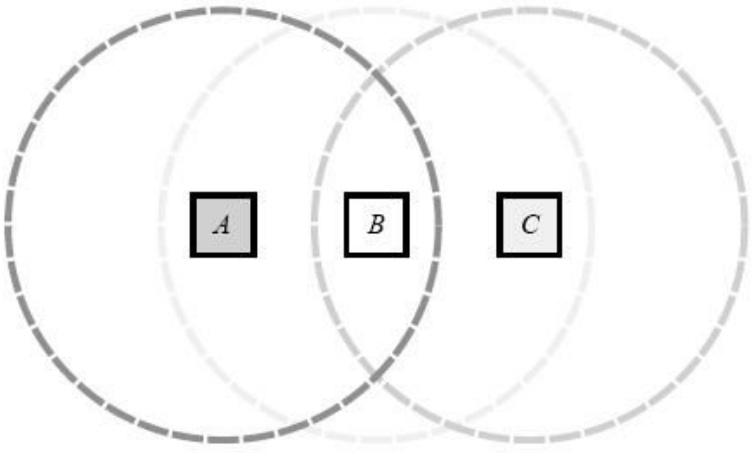

Figure 1.1: A Simple Ad Hoc Network with Three Nodes [5]

send data to node C but node C is not in the range of node A. Then in this case, node A may use the services of node B to transfer data since node B's range overlaps with both the node A and node B. Indeed, the routing problem in a real ad hoc network may be more complicated than this example suggests, due to the inherent nonuniform propagation characteristics of wireless transmissions and due to the possibility that any or all of the hosts involved may move at any time [5].

In MANETs, the nodes must be able to relay traffic since communicating nodes might be out of range. A MANET can be a standalone network or it can be connected to external networks.

Some of the mechanisms that are required in a MANET are:

- Multihop operation requires a routing mechanism designed for mobile nodes.
- Internet access mechanisms are required.
- Self-configuring networks require an address allocation mechanism.
- Security mechanisms are required

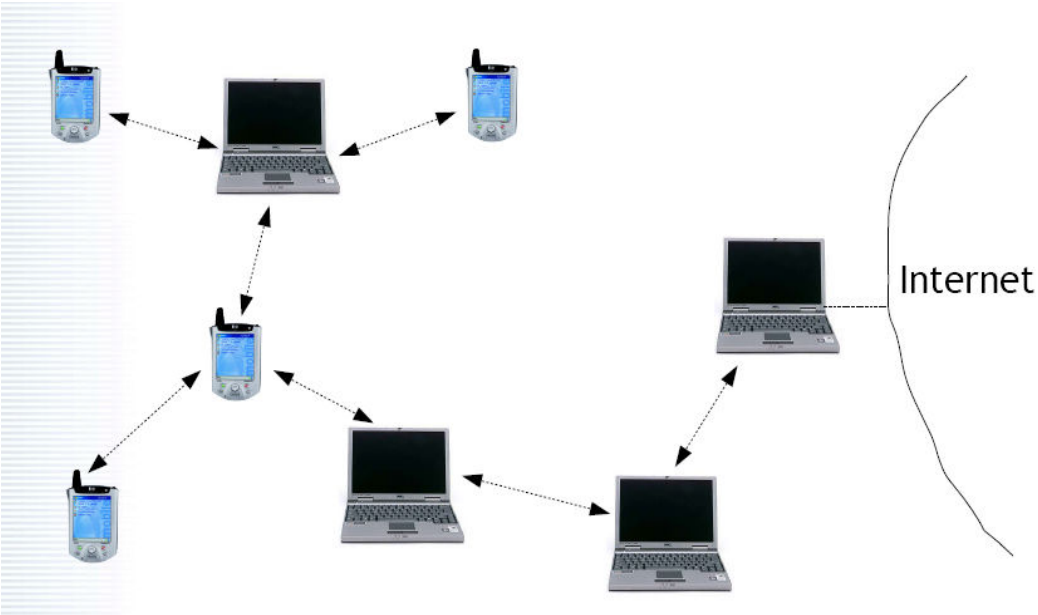

Figure 1.2: Example of a MANET [6]

1.3 WANETs

WANET is a more developed technology than the MANETs. A brief introduction to WANETs is given ahead.

A WANET is a wireless network where the wireless nodes can be located anywhere over the globe. However, the underlying design is such that the nodes believe they are part of a single-hop or multi-hop wireless network at the PHY and MAC layers. This is accomplished by using Software Defined Access Points (SoDA) that are based on the idea of Software Defined Radio (SDR). For the uplink, each SoDA samples the down-converted channel using an ADC (analog to-digital converter). The sampled data is then multicast to the other SoDAs via the Internet. At each end-point, the received digital signals from the other SoDAs are summed and sent through the DAC (digital-to-analog converter) and transmitted on a designated channel after upconversion. Then the RF environment is mixed at geographically separate locations (albeit with a time shift). This simple technique leads to a whole new model for designing and using wireless networks. Indeed, these networks can be thought of as following the end-to-end model where only the end nodes understand the PHY/MAC layers and the access points and Internet serve simply as dumb relays [27].

But MANETs as a technology accepted worldwide has still the scope to be explored more to take an advantage of multihop, multicast, loopfree, adhocity or mobile ad hoc paradigm.

Chapter 2

STATE OF THE ART IN MANETs

Mobile Ad Hoc Networks are wireless networks which do not need any infrastructure support for transferring data between nodes [16]. In these networks, nodes also work as a router, which means that they also route packets for other nodes. Thus a node can be a host or router at different times depending upon the situation i.e. if the node wants to send or receive data, then it will act as a host and if it has to just transfer some data packet to other node, then it will act as a router. Nodes are free to move, independent of each other. Topology of such networks keeps on changing dynamically which makes routing much difficult. Therefore routing is one of the most concerned areas in these networks. Normal routing protocols which work well in fixed networks do not show the same performance in Mobile Ad Hoc Networks because the requirements differ in the two scenarios. In wireless networks, routing protocols should be more dynamic so that they quickly respond to topological changes which occur frequently in these networks [17].

An IETF MANET working group has been established to standardize IP routing in mobile ad- hoc networks.

Three routing protocols are accepted as experimental RFCs, and a fourth one is coming up. These protocols fall into two categories:

1. Reactive
2. Pro-active

Re-active routing protocol does not take initiative for finding routes. It establishes routes “on demand” by flooding a query.

Some pros and cons of reactive routing protocols are:

- It does not use bandwidth except when needed.
- Much network overhead is present in the flooding process when querying for routes.
- There is initial delay in the traffic.

On the other hand pro-active routing protocols set up routes based on continuous control traffic. All routes are maintained all the time.

Some pros and cons of these protocols are:

- Constant overhead is created by control traffic.
- Routes are always available.

2.1 Quality of Service support in MANETs

A MANET can be seen as an autonomous system or a multi-hop wireless extension to the Internet. As an autonomous system, it has its own routing protocols and network management mechanisms. As a multi-hop wireless extension, it should provide a flexible and seamless access to the Internet. Recently, because of the rising popularity of multimedia applications and potential commercial usage of MANETs, QoS support in MANETs has become an unavoidable task.

Due to the bandwidth constraint and dynamic topology of Mobile Ad hoc NETworks (MANET), supporting Quality of Service (QoS) in MANETs is a challenging task. A lot of research has been done on supporting QoS in the Internet and other network architectures, but most of them are not suitable in the MANET environment [9] [2].

2.2 Characteristics of MANETs

A MANET consists of mobile platforms (combined router, host and wireless communications platforms) simply referred to as "nodes", which are free to move about arbitrarily. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people, and there may be multiple hosts per router. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interface with a fixed network--typically envisioned to operate as a stub network connecting to a fixed internetwork [12].

Ad hoc networks are also capable of handling topology changes caused by node malfunctions. For example, if some node leaves the network or shuts down because of some reason, then the network doesn’t collapse. This is because other nodes take its place and reconfigure the network accordingly i.e. set up the IP addresses and other

parameters of the network according to the situation. Also the other nodes will request new routes and the problem of link breakage will be solved. The nodes may be equipped with high power transmitters, antennas or receivers which may be omni directional, highly directional or some combination of these.

Some salient characteristics of MANETs are as follows:

i. **Dynamic Topologies**: The nodes in a MANET are free to move arbitrarily. Thus the topology may change randomly and rapidly and even at unpredictable times and may consist of both bidirectional and unidirectional links.

ii. **Bandwidth-constrained, variable capacity links**: Wireless links will continue to have significantly lower capacity than their hardwired counterparts. In addition, the realized throughput of wireless communications, after accounting for the effects of multiple access, fading, noise, and interference conditions, etc., is often very much less than a radio's maximum transmission rate. One effect of the relatively low to moderate link capacities is that congestion is typically the norm rather than the exception, i.e. aggregate application demand will likely approach or exceed network capacity frequently. As the mobile network is often simply an extension of the fixed network infrastructure, mobile ad hoc users will demand similar services. These demands will continue to increase as multimedia computing and collaborative networking applications rise [12].

iii. **Power- constrained operation**: Some or all of the nodes in the MANET may be based on limited power supplies. For these nodes, the most important system design criteria for optimization may be power conservation. The resource allocation for QoS provisioning must consider the residual battery power and the rate of battery consumption corresponding to the resource utilization. Thus all the techniques for QoS provisioning should be power-aware and power-efficient.

iv. **Limited Physical Security**: Wireless networks are more prone to security issues than wired networks because comparably less research is done in this area. The high threats of eavesdropping, spoofing etc. should be carefully considered before setting up such a network. Although a benefit of MANETs lies in the decentralized approach which they follow since it provides more robustness to single points of failure of centralized approaches [12].

v. **Unpredictable link properties**: Wireless media is very unpredictable. Packet collision is intrinsic to wireless network. Signal propagation faces difficulties such as signal fading, interference and multi-path cancellation. All these properties make the measures, such as bandwidth and delay of a wireless link, unpredictable.

vi. **Hidden and Exposed Terminal Problems***:* In the MAC layer with the traditional carrier sense multiple access (CSMA) protocol, multi-hop packet relaying introduces the "hidden terminal" and "exposed terminal" problems. The hidden terminal problem happens when signals of two, say B and C, which are out of the transmission range of each other, collide at a common receiver, say node A. An exposed terminal is created when a node *A*, is within range of and between two other nodes *B* and *C*, which are out of range of each other. When *A* wants to transmit to one of them, node *B* for example, the other node, *C* in this case, is still able to transmit to a fourth node, *D* which is in *C*'s range (but out of the range of node *A*). Here *A* is an exposed terminal to *C* but can still transmit to *B* [12].

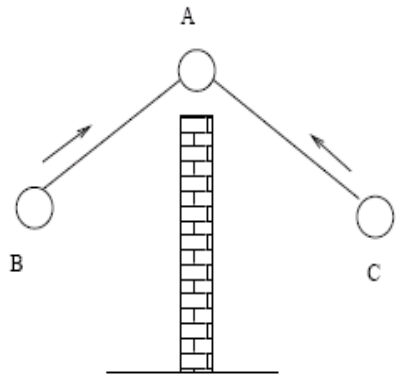

Figure 2.1: Hidden Terminal Problem [1]

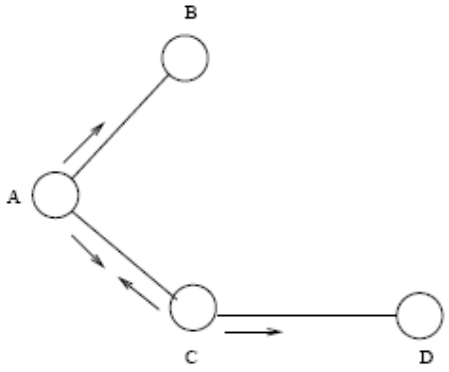

Figure 2.2: Exposed Terminal Problem [1]

vii. **Route Maintenance***:* The dynamic nature of the network topology and the changing behavior of the communication medium make the precise maintenance of network state information very difficult. Thus the routing algorithms in ad hoc networks have to operate with inherently imprecise information. Furthermore, in ad hoc networking environments, nodes can join or leave anytime. The established routing paths may be broken even during the process

of data transfer. Thus arises the need for maintenance and reconstruction of routing paths with minimal overhead and delay.

QoS-aware routing would require reservation of resources at the routers (intermediate nodes). However, with the changes in topology the intermediate nodes also change and new paths are created. Thus the reservation maintenance with the updates in the routing path becomes cumbersome.

2.3 Routing Considerations

Since several hops may be needed for a packet to reach its destination, routing protocols are needed. There are two main functions of the routing protocols. First is finding the routes from the source to the destination and the second one is to deliver the messages or data packets from the source to the destination. The second process is more complicated as it also requires some control mechanism to assure the proper delivery of data packets so that the packets are not lost in the way. This process may also use a variety of protocols and data structures.

2.3.1 Routing in Conventional Networks

While considering conventional networks, one question comes to mind that why not use the conventional routing protocols for wireless networks also. The answer is that the conventional routing protocols are designed keeping in mind the static topology configuration of wired networks. These topologies unlike the wireless topologies do not change with time. Therefore the routing protocol for these networks do not need to control certain parameters which are associated with the change in topology such as assigning new IP addresses to the nodes in the network if some node leaves the network, finding new routes to the destination, resuming data downloads which are in progress, providing some kind of compression facility etc.

Link state and distance vector would probably work very well in an ad hoc network with low mobility i.e. a network where the topology is not changing very often. Then problem that still remains is that link-state and distance vector are highly dependent on periodic control messages. As the number of network nodes can be large, the potential number of destinations is also large. This requires large and frequent exchange of data among the network nodes. This is in contradiction with the fact that

all updates in a wireless interconnected ad hoc network are transmitted over the air and thus are costly in resources such as bandwidth, battery power and CPU. Because both link-state and distance vector try to maintain routes to all reachable destinations, it is necessary to maintain these routes and this also wastes resources for the same reason as above.

Another characteristic for conventional protocols is that they assume bi-directional links, e.g. that the transmission between two hosts is equally well in both directions. In the wireless radio environment this is not always the case.

Since most of the ad hoc routing protocols use one or more of the traditional routing algorithms as their basis, it becomes necessary to have a look at the basic functioning of these conventional routing algorithms like Link State, Distance vector and source routing.

2.3.2 Link State

In link-state routing, each node maintains a view of the complete topology with a cost for each link. To keep these costs consistent, each node periodically broadcasts the link costs of its outgoing links to all other nodes using flooding. As each node receives this information, it updates its view of the network and applies a shortest path algorithm to choose the next-hop for each destination.

Some link costs in a node view can be incorrect because of long propagation delays, partitioned networks etc. Such inconsistent network topology views can lead to formation of routing-loops. These loops are however short-lived, because they disappear in the time it takes a message to traverse the diameter of the network.

2.3.3 Distance Vector

In distance vector, each node only monitors the cost of its outgoing links, but instead of broadcasting this information to all nodes, it periodically broadcasts to each of its neighbors an estimate of the shortest distance to every other node in the network. The receiving nodes then use this information to recalculate the routing tables, by using a shortest path algorithm.

Compared to link-state, distance vector is more computation efficient, easier to implement and requires much less storage space. However, it is well known that distance vector can cause the formation of both short-lived and long-lived routing loops. The primary cause for this is that the nodes choose their next-hops in a completely distributed manner based on information that can be stale.

2.3.4 Source Routing

Source routing means that each packet must carry the complete path that the packet should take through the network. The routing decision is therefore made at the source. The advantage with this approach is that it is very easy to avoid routing loops. The disadvantage is that each packet requires a slight overhead.

2.3.5 Flooding

Many routing protocols use broadcast control information, that is, send the control information from an origin node to all other nodes. A widely used form of broadcasting is flooding and operates as follows. The origin node sends its information to its neighbors. The neighbors relay it to their neighbors and so on, until the packet has reached all nodes in the network. A node will only relay a packet once and to ensure this, some sort of sequence number can be used. This sequence number is increased for each new packet a node sends.

2.3.6 Classification of Routing protocols

Routing protocols can be classified into the following categories depending upon their properties:

- Centralized vs. Distributed
- Static vs. Adaptive
- Reactive vs. Proactive

One way to categorize the routing protocols is to divide them in centralized and distributed algorithms. In centralized algorithms, all route choices are made at a central node, while in distributed algorithms, the computation of routes is shared among the network nodes.

Another classification of routing protocols relates to whether they change routes in response to the traffic input patterns. In static algorithms, the route used by source-destination pairs is fixed regardless of traffic conditions. It can only change in response to a node or link failure. This type of algorithm cannot achieve high throughput under a broad variety of traffic input patterns. Most major packet networks uses some form of adaptive routing where the routes used to route between source-destination pairs may change in response to congestion.

A third classification that is more related to ad-hoc networks is to classify the routing algorithms as either proactive or reactive. Proactive protocols attempt to continuously evaluate the routes within the network, so that when a packet needs to be forwarded, the route is already known and can be immediately used. The family of Distance-Vector protocols is an example of a proactive scheme. Reactive protocols, on the other hand, invoke a route determination procedure on demand only. Thus, when a route is needed, some sort of global search procedure is employed. The family of classical flooding algorithms belongs to the reactive group. Proactive schemes have the advantage that when a route is needed, the delay before actual packets can be sent is very small. On the other side, proactive schemes need time to converge to a steady state. This can cause problems if the topology is changing frequently.

2.4 Compromising Principles in MANETs

The dynamic nature of MANETs is attributed to several inherent characteristics, such as variable link behavior, node movements, changing network topology and variable application demands. Providing QoS in such a dynamic environment is very difficult. Two compromising principles for QoS provisioning in the MANETs are: soft QoS and QoS adaptations [16].

Because of the special properties of mobile wireless networks, some researchers have proposed the notion of soft QoS. Soft QoS means that after the connection setup, there may exist transient periods of time when the QoS specification is not honored. However, we can quantify the level of QoS satisfaction by the fraction of total disruption time over the total connection time. This ratio should not be higher than a threshold.

In a fixed –level QoS approach, a reservation is represented by a point in an n-dimensional space with coordinates defining the characteristics of the service. In a dynamic QoS, we can allow a reservation to specify a range of values, rather than a single point. With such an approach, as available resources change, the network can readjust allocations within the reservation range. Similarly, it is desirable for the applications to be able to adapt to this kind of re-allocations. A good example of this case is the layered real-time video, which requires a minimum bandwidth assurance and allows for enhanced level QoS when additional resources are available. The QoS adaptation can be also done at various layers. The physical layer should take care of changes in transmission quality, for example, by adaptively increasing or decreasing the transmission power. Similarly, the link layer should react to the changes in link error rate, including the use of automatic repeat-request (ARQ) technique. A more sophisticated technique involves adaptive error correction mechanism which will increase or decrease the amount of error correction coding in response to the changes in transmission quality or the desired QoS. As the link layer takes care of the variable bit error rate, the main effect observed by the network layer will be a change in effective throughput (bandwidth) and delay.

2.5 Research in MANETs

A lot of work has been done in supporting QoS in the Internet, but unfortunately none of them can be directly used in MANETs because of the bandwidth constraints and dynamic network topology of MANETs. To support QoS, the link state information such as delay, bandwidth, cost, loss rate, and error rate in the network should be available and manageable. However, getting and managing the link state information in MANETs is very difficult because the quality of a wireless link is apt to change with the surrounding circumstances. Furthermore, the resource limitations and the mobility of hosts make things more complicated. The challenge we face is to implement complex QoS functionality with limited available resources in a dynamic environment [16].

In the literature, the research on QoS support in MANETs includes QoS models, QoS resource reservation signaling, QoS routing, and QoS Medium Access Control

(MAC). Because all of the research just discusses a certain aspect of QoS in MANETs, it is difficult to understand the relationships among the research. Without a whole picture, it is even impossible to understand and evaluate the importance of a particular method. Here, a comprehensive introduction to the current work on QoS support in MANETs is presented. It is described as how the research differs from and coordinates with each other to deliver QoS in MANETs. The relationships among the QoS research are as follows.

First of all, a QoS model specifies an architecture in which some kinds of services could be provided in MANETs. It is the system goal which should be implemented. All other QoS components, such as QoS signaling, QoS routing, and QoS MAC must cooperate together to achieve this goal. It must first be found out what is feasible for supporting QoS in MANETs because it will influence the functionality of all other QoS components. For example, if we only want to provide differentiated quality of services, signaling for every flow state is unnecessary.

Second, QoS signaling acts as the control center in QoS support. It coordinates the behaviors of QoS routing, QoS MAC, and other components such as admission control and scheduling. The functionality of QoS signaling is determined by the QoS model.

Third, QoS routing searches for a path with enough resources but does not reserve resources. It is the QoS signaling to reserve resources (if necessary in the QoS model) along the path determined by QoS routing or other routing protocols. Hence, QoS routing enhances the chance that enough resources can be guaranteed when QoS signaling wants to reserve resources. Without QoS routing, QoS signaling can still work but the resource reservation may fail because the selected path may not have enough resources. QoS signaling will work better if it coordinates with QoS routing. However, since most QoS routing algorithms are complicated, the benefits against the cost of QoS routing in the bandwidth constraint MANETs must be balanced.

Fourth, the QoS MAC protocol is an essential component in QoS support in MANETs. All upper-layer QoS components (QoS routing and QoS signaling) are dependent on and coordinate with the QoS MAC protocol.

Finally, other components in MANETs, such as scheduling and admission control, can be borrowed from other network architectures without or with few modifications.

In the next sections, QoS Models, Resource Reservation Signaling protocols and Routing protocols are described [16].

2.5.1 QoS Models

The QoS model specifies the architecture in which certain services could be provided in the network. A QoS model for MANETs should first consider the challenges of MANETs, e.g. dynamic topology and time-varying link capacity. In addition, the potential commercial applications of MANETs require the seamless connection to the Internet. Thus the QoS model for MANETs should also consider the existing QoS architectures in the Internet. In this section, the QoS models are described for the Internet, such as IntServ and DiffServ, as background.

2.5.1.1 IntServ/RSVP for wired networks

The basic idea of the Integrated Service (IntServ) model is that the flow-specific states are kept in every IntServ-enabled router. A flow is an application session between a pair of end users. A flow-specific state should include bandwidth requirement, delay bound, and cost of the flow. In addition to Best Effort Service, IntServ proposes two service classes, Guaranteed Service and Controlled Load Service. The Guaranteed Service is provided for applications requiring fixed delay bound. The Controlled Load Service is for applications requiring reliable and enhanced best effort service. Because every router keeps the flow state information, the quantitative QoS provided by IntServ is for every individual flow.

The Resource ReSerVation Protocol (RSVP) is used as the signaling protocol to reserve resources in IntServ. Applications with Guaranteed Service or Controlled-Load Service requirements use RSVP to reserve resources before transmission.

IntServ/RSVP model is not suitable for MANETs due to the resource limitation in MANETs: 1) the amount of state information increases proportionally with the

number of flows (the scalability problem, which is also a problem for current Internet). Keeping flow state information will cost a huge storage and processing overhead for the mobile host whose storage and computing resources are scarce. 2) The RSVP signaling packets will contend for bandwidth with the data packets and consume a substantial percentage of bandwidth in MANETs; 3) every mobile host must perform the processing of admission control, classification, and scheduling. This is a heavy burden for the resource-limited mobile hosts.

2.5.1.2 DiffServ

Differentiated Service (DiffServ) is designed to overcome the difficulty of implementing and deploying IntServ and RSVP in the Internet backbone. DiffServ provides a limited number of aggregated classes in order to avoid the scalability problem of IntServ.

DiffServ may be a possible solution to the MANET QoS model because it is lightweight in interior routers. In addition, it provides Assured Service, which is a feasible service context in MANET. However, since DiffServ is designed for fixed wire networks, we still face some challenges to implement DiffServ in MANETs. First, it is ambiguous as to what the boundary routers in MANETs are. Second, the concept of Service Level Agreement (SLA) in the Internet does not exist in MANETs. The SLA is a kind of contract between a customer and its Internet Service Provider (ISP) that specifies the forwarding services the customer should receive. How to make a SLA in MANETs is difficult because there is no obvious scheme for the mobile nodes to negotiate the traffic rules.

2.5.1.3 FQMM

A Flexible QoS Model for MANET (FQMM) is proposed. It considers the characteristics of MANETs and tries to take advantage of both the per-flow service granularity in IntServ and the service differentiation in DiffServ [10].

The provisioning in FQMM, which is used to determine and allocate the resources at various mobile nodes, is a hybrid scheme of per-flow provisioning as in InterServ and per-class provisioning as in DiffServ. FQMM tries to preserve the per-flow granularity for a small portion of traffic in MANET, given that a large amount of the

traffic belongs to per aggregate of flows, that is, per-class granularity. A traffic conditioner is placed at the ingress nodes where the traffic originates. It is responsible for re-marking the traffic streams, discarding or shaping packets according to the traffic profile, which describes the temporal properties of a traffic stream such as rate and burst size.

FQMM is the first attempt at proposing a QoS model for MANETs. However, some problems still need be solved. First, this should be specified as how many sessions could be served by per flow granularity. Without an explicit control on the number of services with per-flow granularity, the scalability problem still exists. Second, just as in DiffServ, the interior nodes forward packets according to a certain PHB that is labeled in the DS field. It is difficult to code the PHB in the DS field if the PHB includes per-flow granularity considering the DS field is at most 8 bits without extension. Finally, making a dynamically negotiated traffic profile is very difficult.

2.5.2 Signaling

QoS signaling is used to reserve and release resources, set up, tear down, and renegotiate flows in the networks. Two distinct mechanisms should be included in a QoS signaling system. First, the QoS signaling information must be reliably carried out between the routers. Second, the QoS signaling information must be correctly interpreted and the relative processing should be activated. Based on the first mechanism, the QoS signaling system can be divided into in-band signaling and out-of-band signaling. The in-band signaling refers to the fact that control information is carried along with data packets; the out-of-band signaling refers to the approach that uses explicit control packets.

2.5.2.1 RSVP

Resource reSerVation Protocol (RSVP) is adopted as the signaling system in the Internet. It has been introduced here simply in order to provide a useful comparison of signaling in the Internet with signaling in MANETs.

RSVP is an out-of-band signaling system. The main motivation for RSVP is to allow efficient support for establishing multicast and unicast connections. For simplicity of

explanation, just the unicast case is described here. When a source host wants to send information to a receiver, it first sends a PATH message to the receiver. The PATH message includes the specification of the traffic characteristics such as rate and burst size. Each intermediate router forwards the PATH message to the next hop determined by the routing protocol. Upon getting the PATH message, the receiver sends back a RESV message to the sender. This RESV message includes the resource requirement for the flow. When a router receives the RESV message, it checks if the required resource can be satisfied. If yes, it allocates the resource for the flow, stores the flow state information, and goes on forwarding the RESV message back to the sender. Otherwise, the resource request is rejected and an error message is sent to the receiver [9].

RSVP has two important characteristics. First, it is the receiver, instead of the sender, that initiates the resource request. Note that different receivers may have different requirements in the multicast case. Second, the flow and reservation information is periodically refreshed. This feature is important in case of link failures. Currently, RSVP has been modified and extended to include more mechanisms, for example, resource reservation for aggregation of flows.

RSVP is not suitable for MANET since the signaling overhead of RSVP is heavy for the mobile hosts. The signaling control message will contend with data packets for the channel and cost a large amount of bandwidth. Furthermore, it is not adaptive for the time-varying topology because it has no mechanism to rapidly respond to the topology change in MANETs.

2.5.2.2 INSIGNIA

INSIGNIA is an in-band signaling system that supports QoS in MANETs. It is probably the first signaling protocol designed solely for MANETs. The signaling control information is carried in the IP option of every IP data packet, which is called the INSIGNIA option. Like RSVP, the service granularity supported by INSIGNIA is per-flow management. Each flow state information is established, restored, adapted and removed over an end-to-end session in response to topology change and end-to-end quality of service condition.

For fast responding to the changes in network topology and end-to-end quality of service conditions, INSIGNIA uses QoS reports to inform the source node of the status of the real-time flows. The destination node actively monitors the received flows and calculates QoS statistical results such as loss rate, delay, and throughput. The QoS reports are periodically sent to the source node. Through this kind of feedback information, the source node can take corresponding actions to adapt the flows to observed network conditions.

As a whole, INSIGNIA is an effective signaling protocol for MANETs. Coordinating with other network components (viz. routing protocol, scheduling, and admission control), INSIGNIA can efficiently deliver adaptive real-time flows in MANETs. However, since the flow state information should be kept in the mobile hosts, the scalability problem may hinder its deployment in the future.

2.5.3 Routing Algorithms

Most QoS routing algorithms represent an extension of existing classic best-effort routing algorithms. Many routing protocols have been developed which support establishing and maintaining multi-hop routes between nodes in MANETs. These algorithms can be classified into two different categories: on-demand (reactive) such as DSR, AODV, and TORA, and table-driven (proactive) such as Destination Sequenced Distance Vector protocol (DSDV).

In the on-demand protocols, routes are discovered between a source and a destination only when the need arises to send data. This provides a reduced overhead of communication and scalability. In the table-driven protocols, routing tables which contain routing information between all nodes are generated and maintained continuously regardless of the need of any given node to communicate at that time. With this approach, the latency for route acquisition is relatively small, which might be necessary for certain applications, but the cost of communications overhead incurred in the continued update of information for routes which might not be used for a long time if at all is too high. Furthermore, this approach requires more memory due to significant increase in the size of the routing table. These requirements put limits on the size and density of the network. A third hybrid approach, the Zone Routing

Protocol (ZRP), has also been proposed and attempts to reap the benefits of both methods. In ZRP, the network is divided into zones. A proactive table driven strategy is used for establishment and maintenance of routes between nodes of the same zone, and a reactive on-demand strategy is used for communication between nodes of different zones. This approach can be effective in larger networks with applications that exhibit a relatively high degree of locality of communication, where communication between nodes with close proximity to one another is much more frequent than that between nodes which are further apart [11].

A figure showing the classification of routing protocols is shown below:

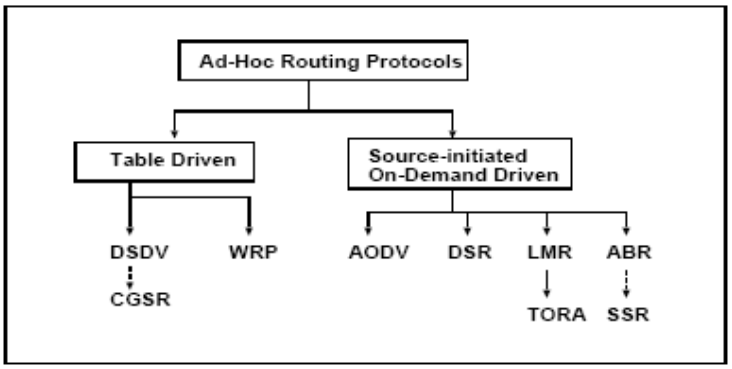

Figure 2.3: A Classification of Routing Protocols [11]

Before presenting the current approaches for design and implementation of QoS routing protocols, it is important to briefly discuss the existing best-effort routing protocols which exist for MANETs. Many routing protocols have been designed to discover and maintain routes between source and destination nodes [8].

Among the most important and classic routing algorithms for MANETs that have evolved are three basic types. Each of these three basic types has its own advantages, disadvantages, and appropriateness of use in certain types of ad hoc networks depending on the mobility, number of nodes involved, node density, underlying link layer technology, and general characteristics of the environment and applications being supported. These three routing algorithms are: (1) reactive (on demand) such as DSR (Dynamic Source Routing) protocol, AODV (Ad hoc On Demand Distance Vector) routing protocol, and TORA (Temporally Ordered Routing Algorithm) protocol, and (2) proactive (table-driven) such as DSDV (Destination Sequenced

Distance Vector) protocol. There are also other types of routing protocols designed for more scalability such as (3) the ZRP (Zone Routing Protocol), which is a hybrid framework for routing in ad hoc networks (proactive within the zone and reactive between zones), in addition to others, which will be mentioned subsequently [8].

2.5.3.1 DSR- Dynamic Source Routing Protocol

DSR is one of the most well known routing algorithms for ad hoc wireless networks. It was originally developed by Johnson, Maltz, and Broch. DSR uses source routing, which allows packet routing to be loop free. It increases its efficiency by allowing nodes that are either forwarding route discovery requests or overhearing packets through promiscuous listening mode to cache the routing information for future use. DSR is also on demand, which reduces the bandwidth use especially in situations where the mobility is low. It is a simple and efficient routing protocol for use in ad hoc networks. It has two important phases, route discovery and route maintenance [8] [11].

The main algorithm works in the following manner. A node that desires communication with another node first searches its route cache to see if it already has a route to the destination. If it does not, it then initiates a route discovery mechanism. This is done by sending a Route Request message. When the node gets this route request message, it searches its own cache to see if it has a route to the destination. If it does not, it then appends its id to the packet and forwards the packet to the next node; this continues until either a node with a route to the destination is encountered (i.e. has a route in its own cache) or the destination receives the packet. In that case, the node sends a route reply packet which has a list of all of the nodes that forwarded the packet to reach the destination. This constitutes the routing information needed by the source, which can then send its data packets to the destination using this newly discovered route [8]. Although DSR can support relatively rapid rates of mobility, it is assumed that the mobility is not so high as to make flooding the only possible way to exchange packets between nodes.

2.5.3.2 AODV - The Ad Hoc On-demand Distance-Vector Protocol

AODV is another routing algorithm used in ad hoc networks. Unlike DSR, it does not use source routing, but like DSR it is on-demand. In AODV, each node maintains a

routing table which is used to store destination and next hop IP addresses as well as destination sequence numbers. Each entry in the routing table has a destination address, next hop, precursor nodes list, lifetime, and distance to destination [8] [11].

To initiate a route discovery process a node creates a route request (RREQ) packet. The packet contains the source node's IP address as well as the destination's IP address. The RREQ contains a broadcast ID, which is incremented each time the source node initiates a RREQ. The broadcast ID and the IP address of the source node form a unique identifier for the RREQ. The source node then broadcasts the packet and waits for a reply. When an intermediate node receives a RREQ, it checks to see if it has seen it before using the source and broadcast ID's of the packet. If it has seen the packet previously, it discards it. Otherwise it processes the RREQ packet. To process the packet the node sets up a reverse route entry for the source node in its route table which contains the ID of the neighbor through which it received the RREQ packet. In this way, the node knows how to forward a route reply packet (RREP) to the source if it receives one later. When a node receives the RREQ, it determines if indeed it is the indicated destination and, if not, if it has a route to respond to the RREQ. If either of those conditions is true, then it unicasts a route reply (RREP) message back to the source. If both conditions are false, i.e. if it does not have a route and it is not the indicated destination, it then broadcasts the packet to its neighbors. Ultimately, the destination node will always be able to respond to the RREQ message. When an intermediate node receives the RREP, it sets up a forward path entry to the destination in its routing table. This entry contains the IP address of the destination, the IP address of the neighbor from which the RREP arrived, and the hop count or distance to the destination. After processing the RREP packet, the node forwards it toward the source. The node can later update its routing information if it discovers a better route. This could be used for QoS routing support to choose between routes based on different criteria such as reliability and delay. To provide such support additional QoS attributes would need to be created, maintained, and stored for each route in the routing table to allow the selection of the appropriate route among multiple routes to the destination [8] [13].

An illustration of AODV is shown below:

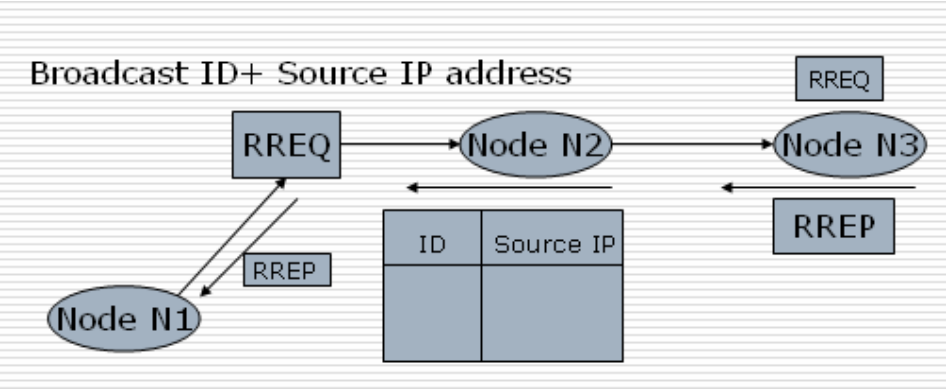

Figure 2.4: An Illustration of AODV Protocol

2.5.3.3 TORA - The Temporally Ordered Routing Algorithm

TORA is the most well known LRR (Link Reversal Routing) algorithm which provides a very adaptive type of routing. It is intended to be used in networks with rapidly changing topologies. It uses a strategy of de-coupling of far-reaching control message propagation from the dynamics of the network's topology. It is efficient to use TORA in networks where the rate of topology changes is not so fast as to make flooding the only form of transmitting messages and not so slow as to make the use of algorithms supporting shortest path calculations applicable. Therefore, the algorithm's applicability is a function of the network's size, rate of topological changes, and available bandwidth. TORA minimizes the network messages in reaction to changes in topology, which are caused by link activation and failure. The algorithm localizes the reaction to these topological changes. TORA does not maintain information sufficient to support shortest path calculation, and maintains only state information sufficient to form a DAG (directed acyclic graph) routed at the destination. The destination is therefore the only node with no outgoing links (a sink). The maintenance of the DAG provides loop free communication to the destination. It also allows the existence of multiple paths to the destination. This provides good reliability, which is desirable in ad hoc networks, and possible QoS extension support, by selecting paths with particular characteristics and that can support pre-specified QoS constraints.

TORA is source initiated and demand driven. Therefore, due to its nature, it forgoes optimal routing. It does not make sure to select the shortest possible path, even though it can be shown that due to the nature of RPY message propagation, shorter paths are more likely to form. However, it provides routing which is very adaptive and scalable with relatively small overhead bandwidth usage for control messages. In addition, lower delivery latency can be achieved.

In contrast with other earlier LLR (Link Reversal Routing) algorithms, TORA's key feature is its reaction to link failures. This reaction is structured as a temporally ordered sequence of diffusing computations with each computation consisting of a sequence of directed link reversals. Each link reversal sequence effectively conducts a search for alternative routes to the destination. The search mechanism in TORA often involves only a single pass of the distributed algorithm because it simultaneously modifies the routing tables during the outward phase of the search procedure itself. This is not the case in other approaches such as DSR and AODV which take three-pass procedures (i.e. route-error/route-request/route-reply) to discover new routes when a node loses its last route. The algorithm uses a "physical or logical clock" to provide a temporal order of topological change events, which is used to structure the protocol's reaction to changes [8] [13].

2.5.3.4 DSDV - The Destination Sequenced Distance Vector Protocol

DSDV is one of the most well known table-driven routing algorithms for MANETs. It is a distance vector protocol. In distance vector protocols, every node i maintains for each destination x a set of distances $\{d_{ij}(x)\}$ for each node j that is a neighbor of i. Node i treats neighbor k as a next hop for a packet destined to x if $d_{ik}(x)$ equals $min_j\{d_{ij}(x)\}$. The succession of next hops chosen in this manner leads to x along the shortest path. In order to keep the distance estimates up to date, each node monitors the cost of its outgoing links and periodically broadcasts to all of its neighbors its current estimate of the shortest distance to every other node in the network. The distance vector which is periodically broadcasted contains one entry for each node in the network which includes the distance from the advertising node to the destination. The distance vector algorithm described above is a classical Distributed Bellman-Ford (DBF) algorithm [11].

DSDV is a distance vector algorithm which uses sequence numbers originated and updated by the destination, to avoid the looping problem caused by stale routing information. In DSDV, each node maintains a routing table which is constantly and periodically updated (not on-demand) and advertised to each of the node's current neighbors. Each entry in the routing table has the last known destination sequence number. Each node periodically transmits updates, and it does so immediately when significant new information is available. The data broadcasted by each node will contain its new sequence number and the following information for each new route: the destination's address, the number of hops to reach the destination and the sequence number of the information received regarding that destination, as originally stamped by the destination. No assumptions about mobile hosts maintaining any sort of time synchronization or about the phase relationship of the update periods between the mobile nodes are made. Following the traditional distance-vector routing algorithms, these update packets contain information about which nodes are accessible from each node and the number of hops necessary to reach them. Routes with more recent sequence numbers are always the preferred basis for forwarding decisions. Of the paths with the same sequence number, those with the smallest metric (number of hops to the destination) will be used. The addresses stored in the route tables will correspond to the layer at which the DSDV protocol is operated. Operation at layer 3 will use network layer addresses for the next hop and destination addresses, and operation at layer 2 will use layer-2 MAC addresses [8].

An illustration of DSDV protocol is shown below:

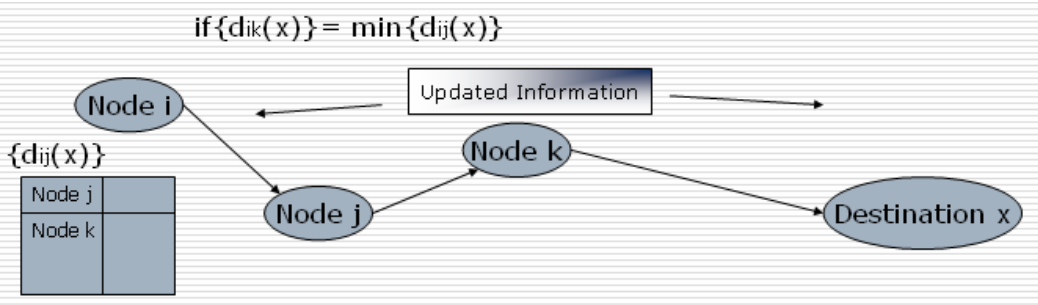

Figure 2.5: An Illustration of DSDV protocol

2.5.3.5 Other Approaches

In addition to the standard routing protocols discussed above, there exist other protocols which use different approaches. The following are some of these protocols.

- ***Location-Assisted Routing***: This approach improves route discovery and maintenance with the use of localization information which is accomplished by keeping track of the position and velocity of the mobile node. Nodes not in that general direction can be excluded from the route discovery and maintenance process to reduce bandwidth consumption and control message communication overhead.
- ***Fisheye Routing***: This is a form of routing which has nodes keeping track of more topology data for closer nodes. It is similar to the Zone Routing Protocol strategy, but with blurred boundaries between zones. This strategy can be very useful to increase the scalability of routing protocols.
- ***Cedar (Core Extraction Distributed Ad Hoc Routing)***: The strategy in this type of routing algorithm is to increase scalability by creating and maintaining a backbone for communication of route requests to avoid broadcasting such information on a network-wide basis. The difficulty is in managing these backbone nodes, which can move relative to each other. Wu and Li in [36] provide a good algorithm for constructing a core which is a connected dominating set. Further research in this area is needed [8].

Chapter 3
PROBLEM STATEMENT

There are many routing protocols in Mobile Ad Hoc NETworks, the popular ones being AODV, DSR and DSDV. Although a lot of research work is done on individual protocols but not enough research is done on comparing these protocols under different environments such as UDP and TCP. This is essential considering the fact that these protocols behave differently or perform differently in different environments. By analyzing how a protocol performs under a certain environment, the shortcomings of the protocol can be found out and more research could be done on removing those shortcomings. Further, this research work also helps in choosing a protocol best suited to particular conditions by finding out the pros and cons of the tested protocols.

The objective of this thesis is to analyze, simulate and do a comparative analysis of three MANET routing protocols namely AODV (Ad Hoc On Demand Distance Vector), DSR (Dynamic Source Routing) and DSDV (Destination Sequenced Distance Vector) under different environments. These three protocols have different properties and based on the way they are designed, they behave differently in different environments. Therefore it becomes essential to analyze each protocol by simulating it in an ideal environment and find out how it performs, so that appropriate methodologies could be followed in the future research works to improve on the areas where a protocol is lacking.

The major objectives of the thesis could be summed up as follows:

- To set up a platform for performing the simulations. The platform could be chosen from Windows and various Linux flavors such as Red Hat, Fedora, Open Solaris and SuSe.
- To learn the platform sufficient enough to perform basic tasks.
- To install and set up appropriate software such as ns2, Tracegraph, NAM (Network Animator) on the selected platform.
- To study the existing protocols in MANETs theoretically.

- To analyze and simulate the chosen protocols with the help of the tools installed on the chosen platform. The tools required are of various types. The simulator tool could be chosen from ns2, OPNET and GloMoSim. The plotting tool could be chosen from among XGraph and Tracegraph.
- To compare the protocols based on their performance in the simulated environment.
- To conclude the results and suggest some improvements in the protocols.

Chapter 4
IMPLEMENTATION

The implementation section discusses how the three protocols i.e. AODV, DSR and DSDV were implemented and analyzed for the comparison. This includes the platform i.e. Fedora and the tools such as ns2 (Network Simulator version 2), NAM (Network Animator) and Tracegraph. Then the core implementation is discussed

3.1 The Platform

All the simulation, implementation and analysis work was done on Linux. The flavor of Linux used for this purpose was Red Hat/ Fedora. The reason for choosing this specific Operating System for Research work is that it is one of the most stable and robust platforms around. Secondly Linux systems provide more security than others and security is a very essential element in network environments. Since the platform provided the basis for doing everything, therefore it becomes essential to discuss some core features of this platform and also somewhat on how it evolved and who is actively working behind the scenes.

3.2 Insight into Red Hat / Fedora

In 2003, Red Hat introduced a desktop developer version of their Linux called Fedora Core and patented the name "Fedora". Red Hat announced that their Fedora Project was aimed to be a developer as well as a newbie oriented distribution. This was targeted mainly at the home user, in other words, for non-commercial purposes. Red Hat had to do this because its Red Hat Linux (commonly shortened to RHL) was made available only for a fee, the name being changed to Red Hat Enterprise. Red Hat claims to charge not for the software itself, but for the services they provide, such as assessing an organization's needs and setting up the systems for them. Red Hat hence could have only stable components in the Enterprise version as it was targeted at businesses. This meant there had to be a different means of ensuring support from the developer community. The Fedora project therefore came into being, where newer components, though possibly a little unstable, would be added for the enthusiast.

3.2.1 Difference between Red Hat and Fedora

Red Hat actually borrows inputs from the Fedora community. This ensured that a new and feature enhanced version would be available for free to the home user, while at the same time, the corporate user would get a stable, tried and tested version along with assured support, but for a price. It seems to work well, as one might think, but Red Hat is at the receiving end of various criticisms by open source purists for their pricing of the software or service.

3.2.2 Features of Fedora 8

The latest version of Fedora is Fedora 10 but the one used for the analysis and simulation work for this thesis is Fedora Core 8 as it is recognized to be one of the most stable versions of Fedora. It has also a 64 bit version available for download. Fedora 8 is a one DVD installer, with the total download size being close to 3.2 GB.

Fedora 8 offers the choice of setting up either a KDE or GNOME Desktop or both. If the complete installation is chosen, then about 7 GB of space is needed for the system files and applications. By installing typical components and leaving out developer tools, one can install it in about 2 to 3 GB. Installation is taken care of by Anaconda, one of the best installers available today. The Disk Druid tool, which allows one to edit the partition table, can be a little confusing for a newbie despite the on-screen instructions.

Hardware is well supported on Fedora 8. ATi/AMD and nVidia, the two largest video card manufacturers today, have their cards supported. A variety of monitor models are also recognized, and the display is adjusted accordingly. Support for card readers is also inbuilt, with the auto-mount feature.

Being a developer and enthusiast-oriented release, Fedora is scheduled for release every four to six months, while the more stable and tested Red Hat is projected for upgrades every year or longer. The biggest advantage one has with the Fedora is the seemingly infinite number of applications written to run on it. The rpm (Red Hat Package Manager) format is more or less the default standard adopted in distributing applications. Even SuSE has agreed to support the rpm format. Installing applications is a matter of just one command, making it very friendly for a newbie.

In terms of features and customizability, Fedora leaves little to be desired. One can spend hours just customizing their machine without doing any work. The biggest plus point for Fedora must be the availability of a vast pool of applications that can be installed besides the bundled ones.

3.3 ns2

After setting up the platform, a software named ns2 was set up on it which was used for all the analysis and simulation work apart from other tools used.

ns2 is the de facto standard for network simulation. Its behavior is highly trusted within the networking community. It is developed at ISI, California, and is supported by the DARPA and NSF.

ns2 is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. This means that most of the simulation scripts are created in Tcl. If the components have to be developed for ns2, then both tcl and C++ have to be used.

ns2 uses two languages because any network simulator, in general, has two different kinds of things it needs to do. On the one hand, detailed simulations of protocols requires a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important [26].

ns2 meets both of these needs with two languages, C++ and Otcl. C++ is fast to run but slower to change, making it suitable for detailed protocol implementation. Otcl runs much slower but can be changed very quickly (and interactively), making it ideal for simulation configuration.

The simulator supports a class hierarchy in C++, and a similar class hierarchy within the OTcl interpreter. The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. The root of this hierarchy is the class TclObject. Users create new simulator objects through the interpreter; these objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy. The interpreted class hierarchy is automatically established through methods defined in the class TclClass. User instantiated objects are mirrored through methods defined in the class TclObject. There are other hierarchies in the C++ code and OTcl scripts; these other hierarchies are not mirrored in the manner of TclObject [21].

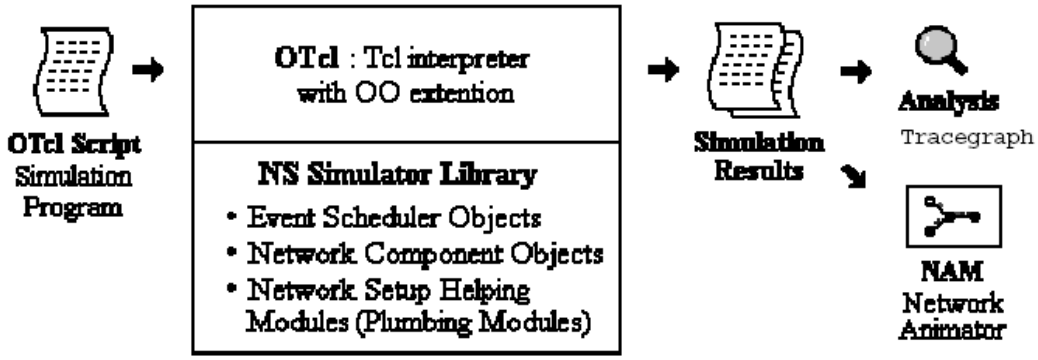

Figure 4.1: An Illustration of the Working of NS2 [22]

3.3.1 Setting up ns2

There are two methods of downloading and installing ns2. The first one is to use the bits and pieces and install each one of this and the second one is to download and install one single package i.e. 'all in one' package. Although it is quite cumbersome to install each and every piece manually, but on the other hand, it gives more flexibility to the user as the user can select which piece is needed and to install that one only. Whereas the 'all in one' package installs every component, irrespective of the fact that it is needed or not.

3.4 NAM (Network Animator)

NAM is a Tcl/TK based animation tool for viewing network simulation traces and real world packet trace data. The first step to use NAM is to produce the trace file. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Usually, the trace file is generated by ns2. During an ns2 emulation, user can produce topology configurations, layout information, and packet traces using tracing events in ns2 [21].

When the trace file is generated, it is ready to be animated by NAM. Upon startup, NAM will read the trace file, create topology, pop up a window, do layout if necessary and then pause at the time of the first packet in the trace file. Through its user interface, NAM provides control over many aspects of animation.

The main window of NAM is shown below:

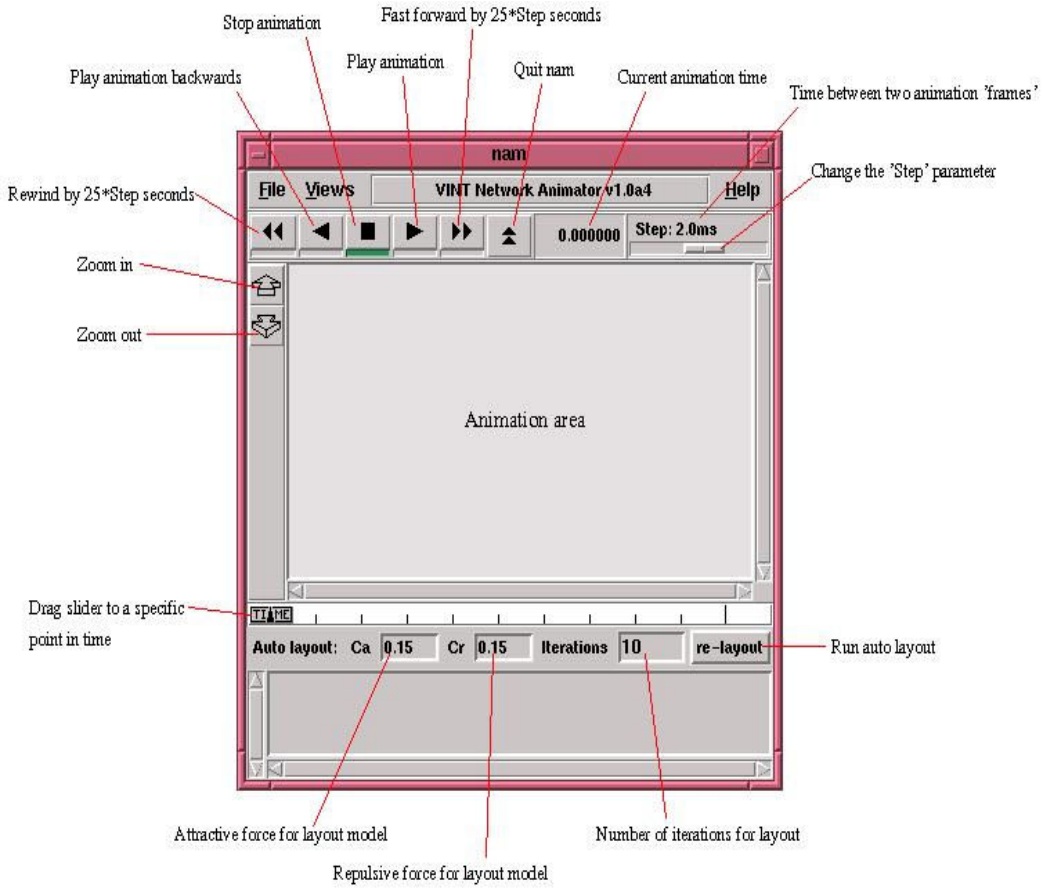

Figure 4.2: The Main Window of NAM [25]

3.5 Trace graph

Trace graph is a free tool for analyzing the trace files generated by ns2. Trace graph can support any trace format if converted to its own or ns2 trace format. Trace graph runs under Windows, Linux, UNIX and MAC OS systems [24].

Some of the program features are as follows:

- 238 2D graphs: Tracegraph supports drawing 238 different graphs depending upon different parameters in 2 Dimensional area.
- 12 3D graphs: Tracegraph supports 12 graphs in 3 Dimensions.
- Delays, jitter, processing times, round trip times, throughput graphs and statistics can be plotted with the help of Tracegraph. These are described below:
    - Delay: This is the delay encountered between the sending and receiving of the packet.
    - Jitter: This is the unwanted variation in the output.
    - Processing Time: The time it takes for a node to process the input.
    - Round Trip Time: The time required for a signal pulse to travel from a specific source to a specific destination and back again.
- Whole network, link and node graphs and statistics
- All the results can be saved to text files, graphs can also be saved as jpeg and tiff
- any graph saved in text file with 2 or 3 columns can be plotted
- script files processing to do the analysis automatically

The program does have some disadvantages though, such as it hangs or takes a very long time while trying to open large trace files. Also it some times hangs after displaying the graph in 3D. The reason why this tool was used in the simulation work is that there are not too many graph plotting tools available in the market. Further, it is free and open source and it doesn't have a steep learning curve.

3.6 Core Implementation

This section discusses how the three protocols i.e. AODV, DSR and DSDV were simulated and implemented.

First the platform i.e. Fedora 8 was set up in a virtual environment. Then ns2 was set up on the platform on which the above said protocols were implemented. ns2 requires a script file to be run on it. These script files are written in a language called TCL (Tool Command Language).

Some of the basic code to write script files is given below:
An ns2 simulation starts with this command:

```
set ns [new Simulator]
```

Creating output files (trace files or nam files)

```
#Open the trace file
set file1 [open out.tr w]
$ns trace-all $file1
#Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2
```

The termination of the program is done as follows:

```
#Define a 'finish' procedure
proc finish {} {
        global ns file1 file2
        $ns flush-trace
        #Close the trace file
        close $file1
        close $file2
        #Execute nam on the trace file
        exec nam out.nam &
        exit 0
        }
```

At the end of the ns2 program, a "finish" procedure is called which specifies the time at which the termination should occur.

e.g. **$ns at 125.0 "finish"**

Definition of a node:

set n0 [$ns node]

Definition of a link:

$ns duplex-link $n0 $n2 10 Mb 10 ms DropTail

A small program using the above commands and simulating a UDP link between two nodes is given below:

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
        global ns nf
        $ns flush-trace
        close $nf
        exec nam –a out.nam &
        exit 0
        }

#Create two nodes
set n0 [$ns node]
set n1 [$ns node]

$ns duplex-link $n0 $n1 1Mb 10ms DropTail

#Create a UDP agent and attach it to node n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

#Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
```

$cbr0 set packetSize_ 500

$cbr0 set interval_ 0.005

$cbr0 attach-agent $udp0

#Create a Null agent (a traffic sink) and attach it to node n1

set null0 [new Agent/Null]

$ns attach-agent $n1 $null0

#Connect the traffic source with the traffic sink

$ns connect $udp0 $null0

#Schedule events for the CBR agent

$ns at 0.5 "$cbr0 start"

$ns at 4.5 "$cbr0 stop"

#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Run the simulation

$ns run

Chapter 5
OBSERVATIONS AND RESULTS

5.1 Performance Analysis

The performance analysis has been done on Fedora 8 as the Operating System. NS 2.39 was installed on the platform for simulating the protocols along with necessary software such as Matlab libraries and Tracegraph which is a software for plotting graphs from the trace files. NS (version 2) is an object oriented, discrete event driven network simulator written in C++ and Otcl [1]. It implements network protocols such as TCP and UDP, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations [23].

5.2 Traffic Environment

The tests were performed on CBR traffic with 40 nodes. Packet size was set to 500 and the time interval between transferring the packets was set to 0.005 ms. Bit rate was set to 1 Mbps with a Drop Tail of 10 ms. As it is not easy to create traffic simulations for such large number of nodes manually, therefore the simulations were generated with the help of CMU traffic generator and the scenario was generated with the help of setdest, which are the tools preinstalled with the ns2. The field configuration was set to 500 by 500 m.

The simulations were performed under two protocols i.e. UDP and TCP as the base protocols. The reason for using two different protocols for analysis is that different MANET protocols behave differently under different environments because TCP and UDP use different parameters for communication such as different packet sizes, flags, different header lengths etc.

5.3 Metrics used for Analysis

The following metrics were used for the comparison of the protocols:

i. **Throughput of received packets**: This represents the number of packets received within a given Time Interval.

ii. **Throughput of dropped packets**: This represents the number of packets dropped within a given Time Interval.

iii. **End to End delays**: It represents the delay encountered between the sending and receiving of the packets.

iv. **Jitter**: It represents any unwanted variation in one or more signals generated during the packet transfer.

5.4 Comparison Results (UDP)

5.4.1 Throughput of Received Packets (TRP)

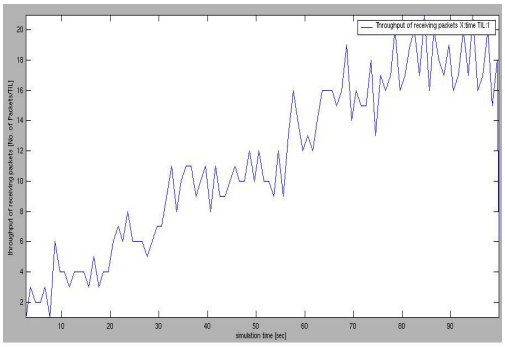

Figure 5.1(a): AOD (TRP)

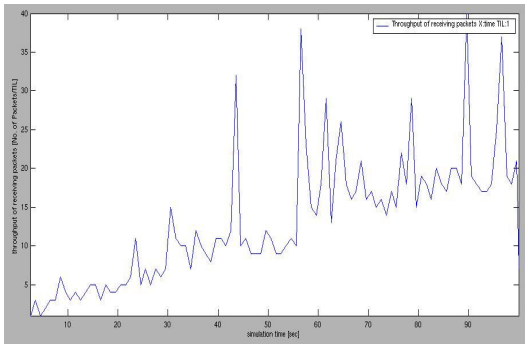

Figure 5.1(b): DSR (TRP)

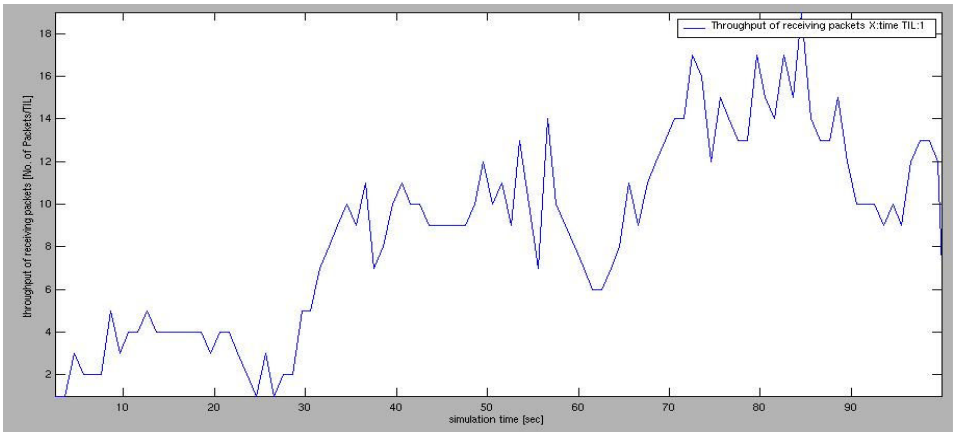

Figure 5.1(c): DSDV (TRP)

We find out from Figure 4.1(a) that the throughput increases gradually over time and that the throughput increase is consistent.

For DSR, we find out from Figure 4.1(b) that the throughput increases gradually over time but the spikes show that there is sudden increase in throughput at times which shows a lack of consistency at certain periods.

From the figure of DSDV protocol, it is clear that the throughput is constant from 10 to around 25 seconds and then it jumps suddenly. Then it remains almost constant for certain time period and then it jumps again.

5.4.1.1 Verdict

From the individual analysis of the protocols, we conclude that AODV protocol shows the most consistency of all the protocols. Although DSR is consistent at most times but a sudden jump in throughput at times shows a lack of consistency. DSDV is nowhere near the other two protocols in terms of consistency.

5.4.2 Throughput of Dropping Packets (TDP)

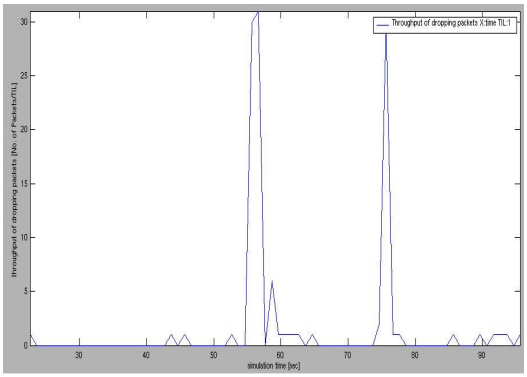

Figure 5.2(a): AODV (TDP)

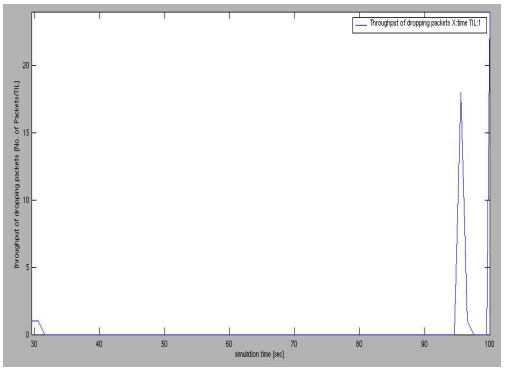

Figure 5.2(b): DSR (TDP)

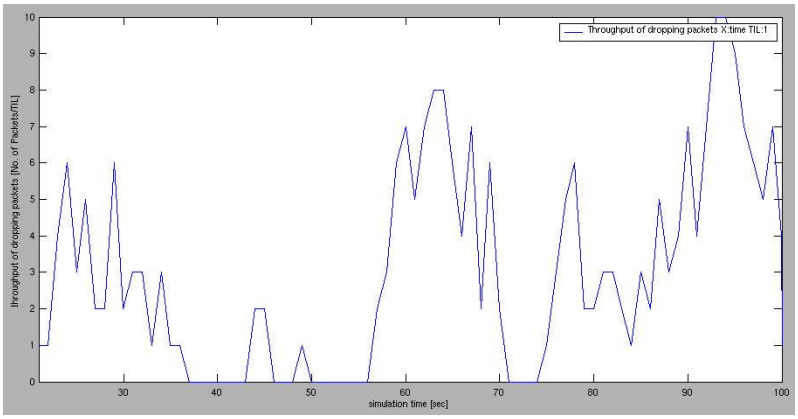

Figure 5.2(c): DSDV (TDP)

From Figure 4.2(a), we see that the throughput of dropping packets for the AODV protocol is low. Only at a couple of times, it shows a jump in the dropping rate, but nonetheless, the overall rate of packet drop is less.

From Figure 4.2(b), we find out that the packet drop throughput for the DSR protocol is also less, even less than AODV protocol.

From Figure 4.2(c) we see that the packet dropping rate for the DSDV protocol is very high. It means that it drops packets frequently.

5.4.2.1 Verdict

We conclude from the above discussion that the protocol with the least packet drop throughput is the DSR. AODV also performs very close to DSR but the performance of DSDV is the worst of all. It has the highest packet drop rate and thus performs poorly.

5.4.3 End to End Delays (EED)

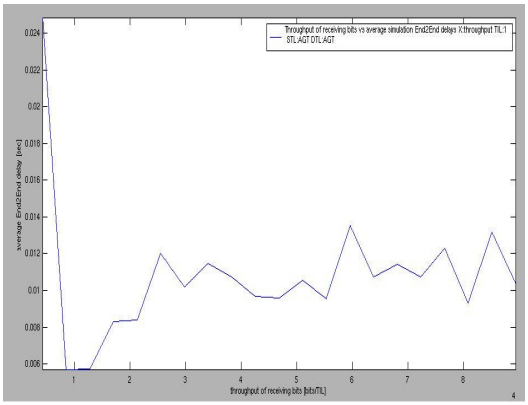

Figure 5.3(a): AODV (EED)

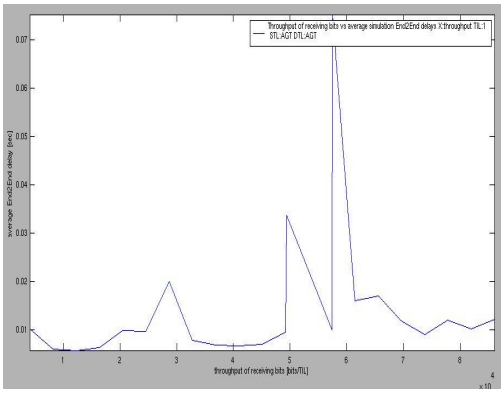

Figure 5.3(b): DSR (EED)

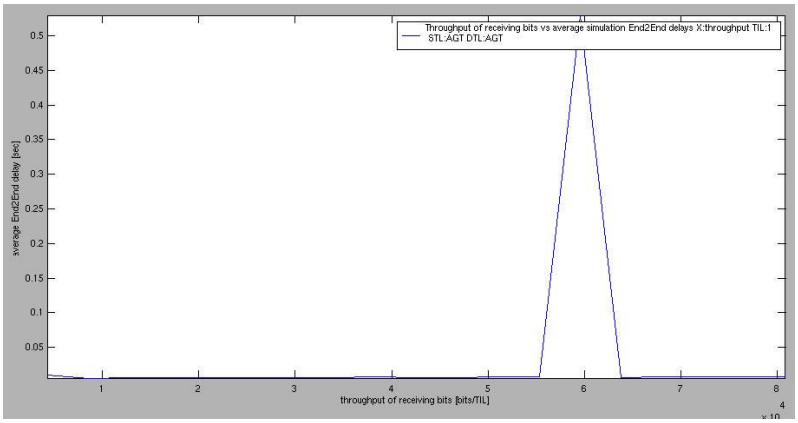

Figure 5.3(c): DSDV (EED)

From the figure 4.3(a), we find out that there is some initial delay caused in the throughput which is probably the delay caused during the route discovery process by AODV. After that, as the throughput increases, the end to end delay also increases but becomes almost constant at around 0.012 seconds.

The figure of DSR protocol shows that an initial delay of 0.01 seconds is introduced to start the throughput and then it drops somewhat and then it rises suddenly to very high levels and then again drops.

For DSDV protocol, we see from the figure that the end to end delay is almost zero. Only at a single point, it shows a sharp rise but overall it is negligible.

5.4.3.1 Verdict

From the individual discussion of the end to end delays of the three protocols, we conclude that the protocol which introduces the minimum delay in the throughput is DSDV. AODV also performs well with not much delay in the throughput whereas DSR has the maximum delay in the throughput.

5.4.4 Jitter (JT)

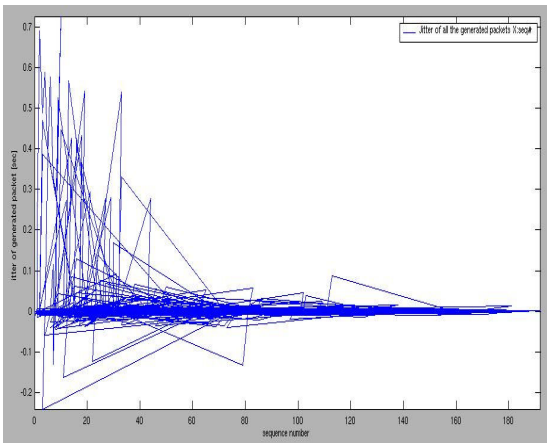

Figure 5.4(a): AODV (JT)

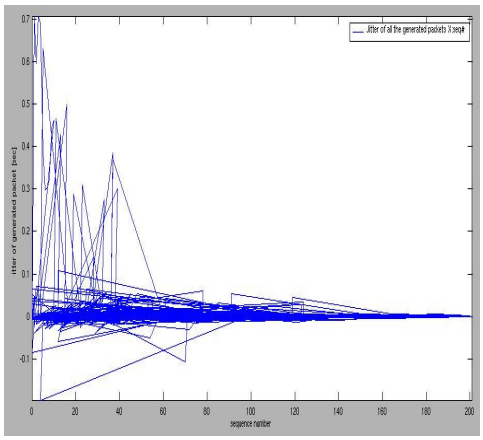

Figure 5.4(b): DSR (JT)

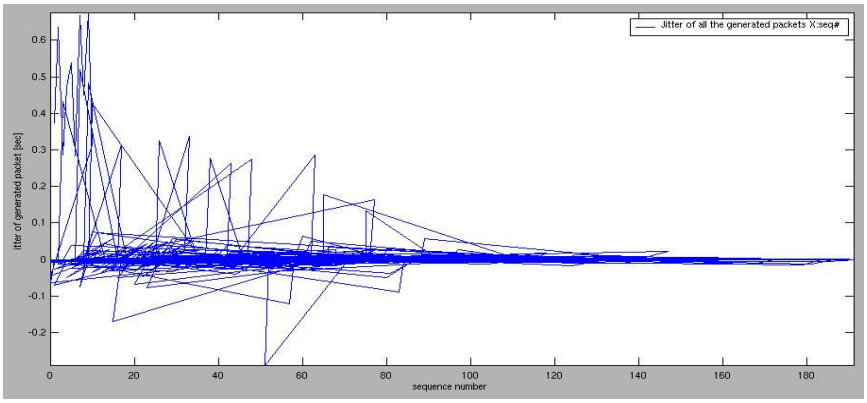

Figure 5.4(c): DSDV (JT)

We observe from the Figure 4.4(a) that the average jitter for the initial packets varies from 0 to 0.5 seconds and then it remains almost constant at around 0.05 seconds.

The jitter in case of DSR protocol is not much different from that of AODV with the initial average jitter varying from 0 to 0.4 seconds. The jitter of DSDV protocol is almost similar to AODV and DSR.

5.5 Comparison Results (TCP)

After comparing the protocols with the UDP protocol as the base, the comparisons were done under TCP. The observations done are described below.

5.5.1 Throughput of Received Packets (TRP)

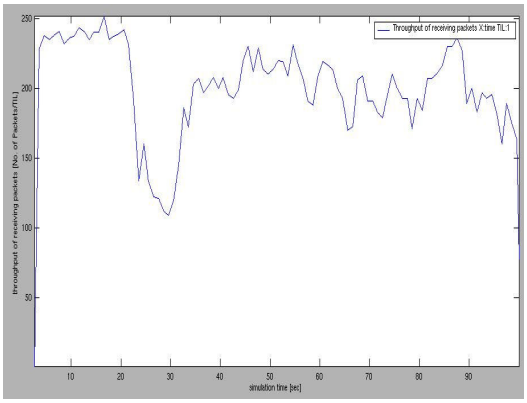

Figure 5.5(a): AODV (TRP)

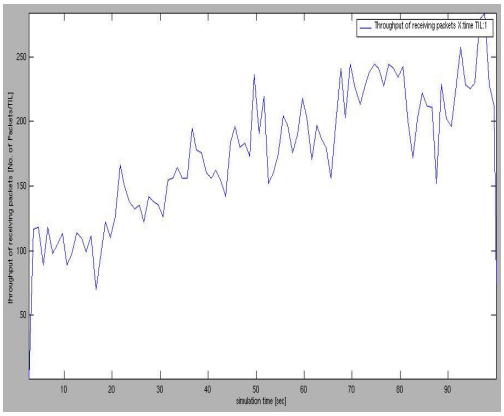

Figure 5.5(b): DSR (TRP)

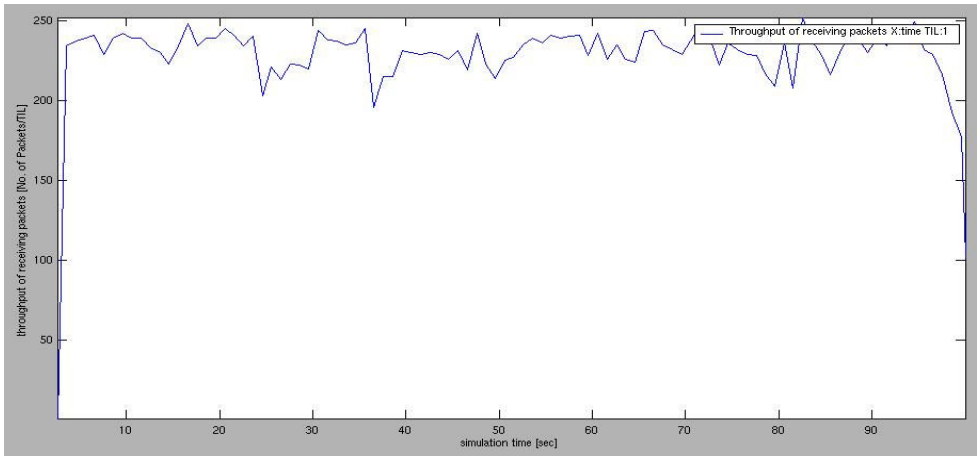

Figure 5.5(c): DSDV (TRP)

Figure 4.5(a) shows that the throughput is high initially and then it drops suddenly, then it rises and remains almost constant for the rest of the time period. It is clear from the figure that there is a lack of consistent throughput in this protocol.

Figure 4.5(b) depicts that the throughput rises constantly throughput the graph and the rise in throughput is consistent i.e. it rises in constant intervals.

From the figure 4.5(c), it is clear that the throughput is very high in the initial stage and the rise in throughput is maintained throughput the time interval.

5.5.1.1 Verdict

From above, it can be concluded that the performance of DSDV protocol is the best of all the three protocols. The throughput is very high in the beginning and it is maintained throughout. AODV protocol also performs well but is not as consistent as DSDV. On the other hand, DSR doesn't give as high a throughput as the other two protocols.

This is totally in contrast to the comparisons done under the UDP where AODV was the best performing and DSDV was the worst performing. It certainly shows that different protocols perform differently under different environments.

5.5.2 Throughput of Dropping Packets (TDP)

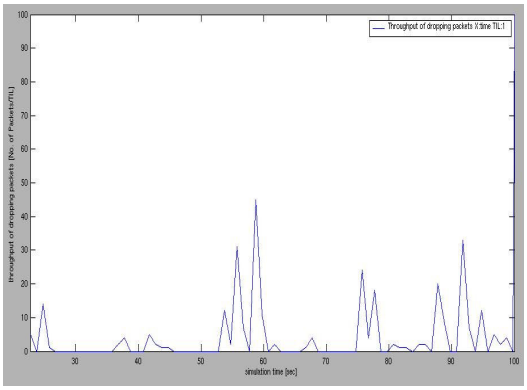

Figure 5.6(a): AODV (TDP)

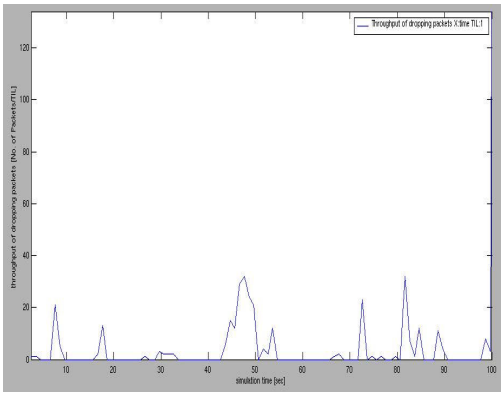

Figure 5.6(b): DSR (TDP)

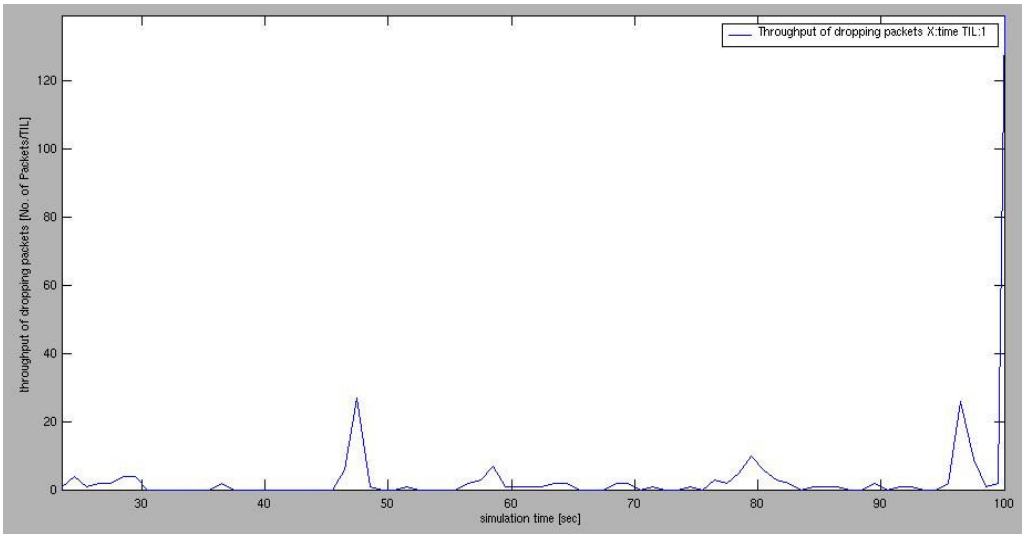

Figure 5.6(c): DSDV (TDP)

From Figure 4.6(a), it is clear that the throughput of dropping packets for the AODV protocol is pretty much high. It shows high jumps at certain places which means that the packets are frequently dropped.

Figure 4.6(b) shows that the packet drop is less than the AODV protocol although it also shows certain amount of packet drop at times.

It is clear from the Figure 4.6(c) that the packet drop for the DSDV is very low, almost close to zero.

5.5.2.1 Verdict

From the above discussion, it is clear that the packet drop is the least for the DSDV protocol which clearly indicates its high performance. The performance of DSR protocol is also good as it also drops less packets in a given time interval whereas AODV has the highest packet drop rate of the three protocols.

Again, this is in contrast to the results obtained while performing the tests under UDP in which case DSR performed the best and DSDV dropped the highest number of packets.

5.5.3 End to End Delays (EED)

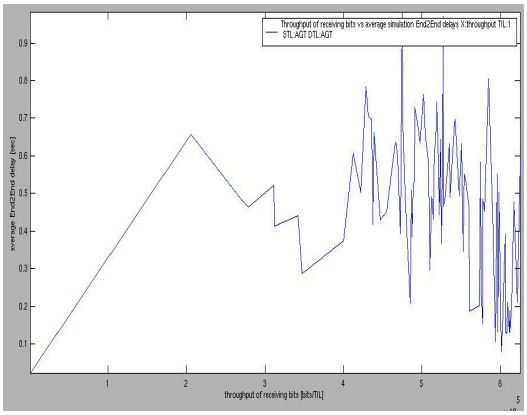

Figure 5.7(a): AODV (EED)

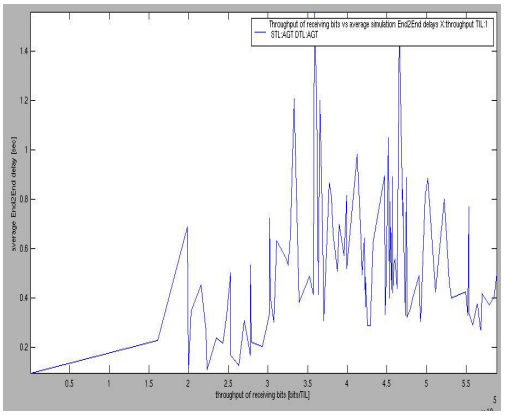

Figure 5.7(b): DSR (EED)

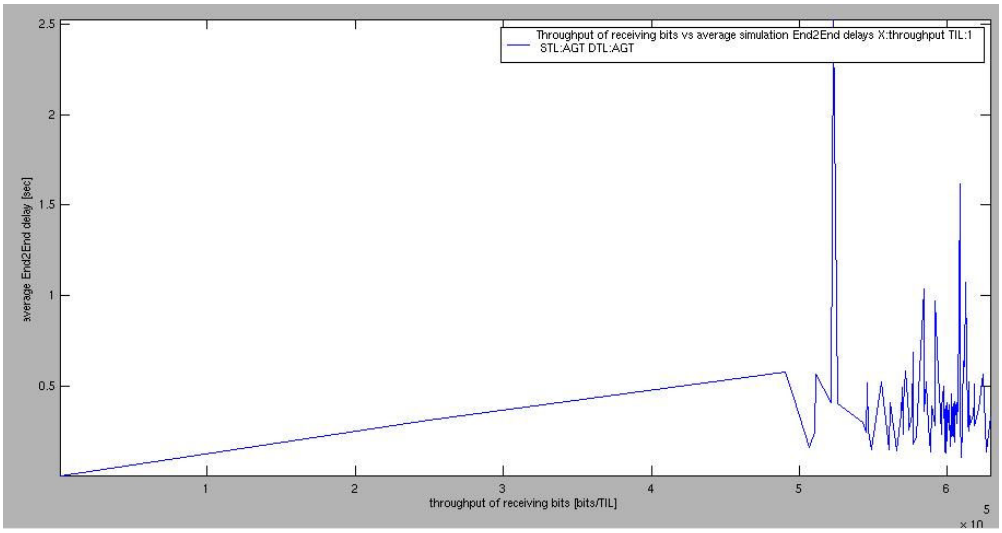

Figure 5.7(c): DSDV (EED)

Figure 4.7(a) depicts that the end to end delay increases gradually up to about 0.65 seconds and then it drops and then shows some variation in the delay. The average end to end delay remains close to 0.7 seconds. This means that AODV does introduce some delay in the throughput which is possibly caused because of the route discovery process.

Figure 4.7(b) shows that there is small delay in receiving the initial packets but after that, the delay increases gradually. After that, very high delay is introduced between

data packets. At some points, it even crosses 1.4 seconds otherwise it is close to 0.8 seconds overall which can be considered high.

The observations for the DSDV protocol reveal that the delay is less for initial some time and then it rises to high levels. Then it suddenly drops and remains almost constant at about 0.5 seconds.

5.5.3.1 Verdict

From the above results, it can be concluded that although the end to end delay cannot be considered satisfactory for any of the three protocols but the DSDV protocol is the best performing of all the protocols. On the other hand, the results for AODV and DSR protocols are poor as they introduce high delays in the packet reception.

Comparing the results to the ones under UDP, it can be depicted that the DSDV performs similarly in both the cases whereas DSR introduces high delays in both cases.

5.5.4 Jitter (JT)

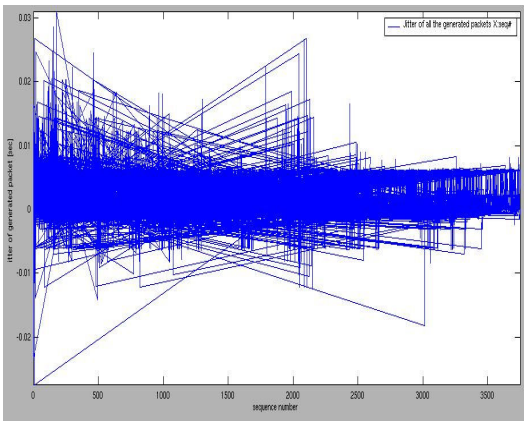

Figure 5.8(a): AODV (JT)

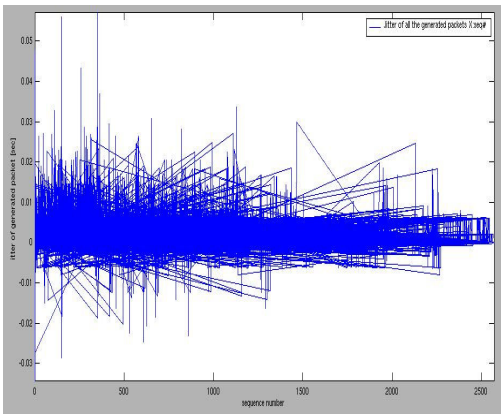

Figure 5.8(b): DSR (JT)

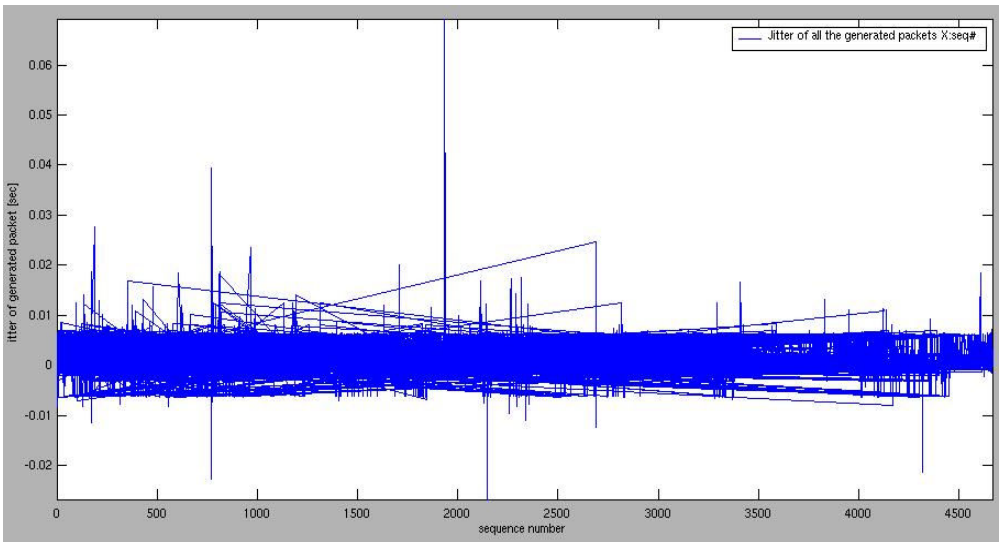

Figure 5.8(c): DSDV (JT)

The above graphs show the sequence numbers vs. the jitter of generated packets. The sequence numbers here refer to the packets only as a different sequence number is generated for every new packet created. The graph in the Figure 4.8(a) indicates the jitter for the AODV protocol. The graph shows that the average initial jitter varies from 0 to 0.03 seconds and then it varies from 0 to 0.01 seconds although at times, it goes beyond that.

The second graph is for the DSR protocol which shows that the average jitter for this protocol varies between 0 and 0.02 seconds. However, sometimes it goes up to 0.03 seconds and up to 0.05 seconds a couple of times.

The average jitter for the DSDV protocol is less than even 0.01 seconds which is the lowest of the three. However, at one time, it reaches beyond 0.06 seconds.

5.5.4.1 Verdict

Although the jitter for three protocols doesn't seem to be much different from each other, however, in network considerations, even small amount of jitter can cause significant delays. Therefore it can be said that the DSDV protocol performed better than the others. The highest jitter could be seen in the case of DSR protocol.

Comparing the results to the ones from the UDP protocol, where the jitter for the three protocols wasn't much different, it can be found out that the jitter in the case of TCP is less. The jitter is even less in the DSDV protocol.

5.6 Summary of Comparison Results

The results are given below in a tabular format for easy and quick look. There are two tables, the first one giving the results under UDP and the second one under TCP. There are a few abbreviations given in tables which are explained below:

TRP- Throughput of Received Packets

TDP- Throughput of Dropping Packets

EED- End to End Delays

	TRP (higher is better)	**TDP (lower is better)**	**EED (lower is better)**	**Jitter**
AODV	Most consistent	Packet drop similar to DSR	Maximum delay introduced	Initial- 0-0.5 Average- 0.05
DSR	Lack of consistency (sudden jumps in throughput)	Least packet drop	Delay within acceptable limits	Initial- 0-0.3 Average- 0.03
DSDV	Least consistent	Highest packet drop	Minimum delay introduced	Initial- 0-0.3 Average- 0.2

Table 5.1: Comparison Results under UDP

	TRP (higher is better)	TDP (lower is better)	EED (lower is better)	Jitter
AODV	Lack of consistency	Highest packet drop	High delay	Initial- 0-0.03 Average- 0.01
DSR	Least consistency	Packet drop acceptable	High delay	Initial- 0-0.02 Average- 0.02
DSDV	Most consistent	Least packet drop	Minimum delay introduced	Initial- 0-0.01 Average- 0.01

Table 5.2: Comparison Results under TCP

The above two tables give a comprehensive view of the performance of the three protocols i.e. AODV, DSR and DSDV and clearly show that AODV is the best performer under UDP and DSDV gives best performance under TCP.

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This thesis work presents a detailed comparative analysis of three MANET protocols i.e. AODV (Ad Hoc On Demand Distance Vector), DSR(Dynamic Source Routing), and DSDV (Destination Sequenced Distance Vector) under two different environments i.e. UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). The concluded facts are quoted below.

6.1.1 Results for the UDP Protocol

From the observations and results obtained from the previous chapter, we can conclude that the AODV protocol performs well in almost every situation (in case of UDP). The packet received throughput of AODV is good. It is consistent and increases over time. The packet drop throughput of AODV is also not high and it drops packets only at certain time intervals, which means the packets are not dropped frequently. The end to end delay of AODV is also not very high. The DSR protocol is also not far left behind AODV. It performs on par with AODV at certain times whereas it lags behind AODV in the packet received throughput test and end to end delays. The performance of DSDV protocol is not very good in comparison to AODV and DSR although in case of end to end delays, it leaves both the other protocols behind.

This means that the AODV protocol is the ideal choice for communication when the communication has to happen under the UDP protocol as the base.

6.1.2 Results for the TCP Protocol

When it comes to TCP, the results almost reverse. The protocols which performed good in UDP lagged here and the ones which performed poorly excelled under TCP. For example, DSDV gave the worst performance in the “Throughput of received packets” test under UDP but in the case of TCP, it left the other two protocols behind. In the packet drop test also, DSDV gave the best performance by dropping the least

number of packets. Although in the case of end to end delays, the performance of DSDV wasn't much different than that under UDP and the worst performance of DSR was shown in both cases too. In the case of Jitter introduction test, although there wasn't much difference in the outputs of the three protocols, however, DSDV gave slightly better performance than the other two protocols by introducing less Jitter.

6.1.3 The Comparison Concluded

Thus we can conclude that under different environments, every protocol behaves differently because there are many parameters which differ under varied situations. From the above discussion, it can be concluded that the AODV performs the best in case of UDP protocols whereas DSDV is the best performing protocol under TCP. The DSR protocol also doesn't perform poorly. Under UDP, its performance is quite comparable to AODV although under TCP, it lags behind the other two protocols. On the other hand, there are some situations where all the three protocols behave similarly. Therefore, depending upon the situation where there is a high demand of one parameter over the other, the protocol with the high efficiency in that particular parameter can be used.

6.2 Future Scope

In this thesis work, three ad hoc routing protocols i.e. AODV, DSR and DSDV have been analyzed and compared, the results of which could be useful in many situations. However there are other protocols also in MANETs such as TORA, ZRP, INSIGNIA etc. The future scope is the extensive comparisons between the above said protocols. Research on new simulation environments similar to ns2 could also be done, resulting in the development of new features such as more detailed graphs. In addition to this, improving packet delivery efficiency is the challenging area to be explored more.

REFERENCES

[1] Chenxi Zhu and M. Scott Corson. "QoS Routing for Mobile Ad Hoc Networks". In the Proc. IEEE Infocom, June 2001.

[2] Demetris Zeinalipour. "A Glance at QoS in MANETs". University of California, Tech. Rep., 2001.

[3] Jermy I. Blum, Azim Eskandarian, and Lance. J Hoffman, "Challenges of inter vehicle Ad hoc Networks", IEEE transactions on Intelligent Transportation Systems, Vol. 5 No. 4 Dec. 2004.

[4] S. Murthy and J.J. Garcia-Luna-Aceves. "An Efficient Routing Protocol for Wireless Networks". ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, pp 183-197, October 1996.

[5] David B. Johnson and David A. Maltz. "Dynamic Source Routing in Ad Hoc Wireless Networks". In Ad Hoc Wireless Networks, Mobile Computing, T. Imielinski and H. Korth (Eds.), Chapter 5, pp 153-181, Kluwer Academic Publishers, 1996.

[6] Andreas Tønnesen. "Mobile Ad-Hoc Networks"

[7] Ahmed Al-Maashri and Mohamed Ould-Khaoua. "Performance Analysis of MANET Routing Protocols in the Presence of Self-Similar Traffic". IEEE, ISSN- 0742-1303, First published in Proc. of the 31st IEEE Conference on Local Computer Networks, 2006.

[8] Imad Jawhar and Jie Wu. "Quality of Service Routing in Mobile Ad Hoc Networks"

[9] Kui Wu and Janelle Harms. "QoS Support in Mobile Ad Hoc Networks"

[10] Rafael Guimaraes, Juli´an Morillo, Llorenc Cerda, Jos´e-M Barcel´o and Jorge Garc´ia. "Quality of Service for Mobile Ad Hoc Networks- An Overview"

[11] Elizabeth M. Royer and C-K Toh. "A Review of current Routing Protocols for Ad-hoc Mobile Wireless Networks", IEEE Personal Communications, Vol. 6, No.2, pp. 46-55, April 1999.

[12] S. Corson and J. Macker. "MANET: Routing Protocol Performance Issues and Evaluation Considerations". IETF MANET, RFC 2501, 1999.

[13] Zhijiang Chang, Georgi Gaydadjiev and Stamatis Vassiliadis. "Routing Protocols for MANETs: Current Development and Evaluation"

[14] Tony Larsson amd Nisklas Hedman. “Routing Protocols in Wireless Ad-hoc Networks: A Simulation Study”

[15] P. Mohapatra, J. Li, and C. Gui. “QoS in mobile ad hoc networks. IEEE Wireless Communications, June 2003.”

[16] Aleksi Penttinen. “Research on Ad Hoc Networking- Current Activity and Future Directions”

[17] Nitin H. Vaidya,”Mobile Ad Hoc Networks: Routing, MAC and Transport Issues”, University of Illinois at Urbana-Champaign, Tutorial presented at: INFOCOM 2004 (IEEE International Conference on Computer Communication).

[18] Kwan-Wu Chin, John Judge, Aidan Williams and Roger Kermode; Sydney Networks and Communications Lab, Motorola Australia Research Centre. “Implementation Experience with MANET Routing Protocols”. ISSN:0146-4833, 2002.

[19] Zeinalipour-Yazti Demetrios; Department of Computer Science, University of California – Riverside; “A Glance at Quality of Services in Mobile Ad Hoc Networks”

[20] Ram Ramanathan and Martha Steenstrup; Advanced Networking Department, BBN Systems and Technologies. “Hierarchically-organized, multihop mobile wireless networks for quality-of-service support”. ISSN:1383-469X, pp.101-119, 1998.

[21] Kevin Fall and Kannan Varadhan. “The ns Manual (formerly ns notes and documentation)”

[22] Sandeep Gupta. “A brief guide to ns2.”

[23] Giovanni Perbellini. “An introduction to NS-2”, Università degli Studi di Verona, Facoltà di Scienze MM. FF. NN., Verona, 12/09/2005

[24] Jaroslaw Malek. “Trace graph - Network Simulator NS-2 trace files analyzer” http://www.tracegraph.com

[25] Tutorial for the network simulator “ns”. http://www.isi.edu/nsnam/ns/tutorial/

[26] Carnegie Mellon. Computer Science Department. 15-744 Spring 2007 “Problem Set 2”

[27] Suresh Singh. Department of Computer Science, Portland State University, Portland, OR 97207. “Challenges: Wide-Area wireless NETworks (WANETs)”. ACM MOBICOM, San Francisco, CA, September 14-16, 2008.

[28] Luc Hogie, Pascal Bouvry and Fr´ed´eric Guinand. Laboratoire d'Informatique, Universit´e du Havre, France. "An Overview of MANETs Simulation". In Proceedings of the First International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005), Vol. 150, No. 1, pp. 81-101, 9 March 2006.

[29] C. R. Dow, P. J. Lin, S. C. Chen, J. H. Lin and S. F. Hwang. Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. "A Study of Recent Research trends and Experimental Guidelines in Mobile Ad-Hoc Networks". Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on Volume 1, vol. 1, pp. 72 – 77, 28-30 March 2005.

[30] Jeonghoon Park, Sungkyunkwan University. "NS Simulator for Beginners"

[31] Theodore S. Rappaport. "Wireless Communications: Principles and Prentice". New Jersey, Prentice Hall. ISBN 0-13-375536-3.

[32] M. Scott Corson and Anthony Ephremides. "A Distributed Routing Algorithm for Mobile Wireless Networks". Proceedings of IEEE INFOCOM'97, March 1996.

[33] C.E. Perkins, E.M. Belding-Royer and S. Das. "Ad hoc on-demand distance vector (AODV) Routing". RFC 3561, July 2003.

[34] Royer, E.M. and Chai-Keong Toh. "A Review of current routing protocols for ad hoc mobile wireless networks Personal Communication". IEEE, April 1999.

[35] Charles E. Perkins and Pravin Bhagwat. "Highly Dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers". In Proceedings of the SIGCOM'94 Conference on Communications Architecture, protocols and Applications, pp 234-244, August 1994.

[36] S. Murthy and J.J. Garcia-Luna-Aceves. "An Efficient Routing Protocol for Wireless Networks". ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, pp 183-197, October 1996.

[37] H. Deng, W. Li and Dharma P. Aggarwal. "Routing Security in Ad Hoc Networks. In IEEE Communications Magazine, Special Topics on Security in Telecommunication Networks, Vol. 40, No.10, pp 70-75, October 2002.

[38] Sonja Buchegger and Jean-Yves Le Boudec. "Nodes Bearing Grudges: Towards Routing Security, Fairness and Robustness in Mobile Ad Hoc Networks. In

Proceedings of the Tenth Ecurfomicro Workshop on Parallel, Distributed and networks based Processing, pp 403-410, January 2003.

[39] D.P. Aggarwal and Qing-An Zeng. “Introduction to wireless and Mobile Systems”. Brooks/Cole, 2005.

[40] S. Chen and K. Nahrstedt. “Distributed Quality-of-Service Routing in Ad-Hoc Networks”. IEEE Journal on Special Areas in Communications, Vol. 17, No. 8, August 1999.

PAPER ACCEPTED/COMMUNICATED

1. Darshan Singh Kali Rai, Ms. Ashima Singh. "Comparative Analysis of Ad Hoc Routing Protocols". ISAN-2009, National Conference Cum Workshop on Information Security and Networks, Chitkara Institute of Engineering & Technology, Chandigarh, Punjab, 19-20 June, 2009. [Accepted]

2. Darshan Singh Kali Rai, Ms. Ashima Singh. "Analysing AODV, DSR and DSDV w.r.t. UDP and TCP". 7th International Conference on Information Technology: New Generations, April 12-14, 2010, Las Vegas, Nevada, USA [Communicated]