Multipath Routing Protocol For Multimedia Applications In Ad Hoc Wireless Networks

By

Gamit Shailesh V.
(05MCE004)

Department of Computer Science and Engineering

Institute of Technology

Nirma University of Science and Technology

Ahmedabad 382481

May 2007

Multipath Routing Protocol For Multimedia Application In Ad Hoc Wireless Networks

Major Project

submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

By

Gamit Shailesh V.
(05MCE004)

Guide
Prof. Sharada Valiveti

Department of Computer Science and Engineering

Institute of Technology

Nirma University of Science and Technology

Ahmedabad 382481

May 2007

This is to certify that Dissertation entitled

Multipath Routing Protocol For Multimedia Application in Ad Hoc Wireless Networks

Submitted by

Shailesh Gamit

has been accepted toward fulfillment of the requirement
for the degree of
Master of Technology in Computer Science & Engineering

Prof. (Dr.) S. N. Pradhan
Professor In Charge

Prof. D. J. Patel
Head of The Department

Prof. A. B. Patel
Director, Institute of Technology

CERTIFICATE

This is to certify that the Major Project entitled “Multipath Routing Protocol for Multimedia Application in Ad Hoc Wireless Networks” submitted by Mr. Shailesh Gamit (05MCE004), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science & Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Prof. Sharada Valiveti
Guide
Asst. Professor,
Department of Computer Engineering,
Institute of Technology,
Nirma University,
Ahmedabad

Date:-

ACKNOWLEDGEMENT

Although it has come with share of frustrations, doubts, regrets and tradeoffs, this process has been rich with rewards. Although achieving a new milestone in my career with master's is a personal achievement, I did not get here on my own Hence, it is important to recognize and thank the very special people – family, friends, peers, colleagues and academics who have helped to make this achievement possible.

First and foremost I would like to express my humble and sincere thanks to my guide – **Prof. Sharada Valiveti** to continuously guide me in my project area and to provide me with all the necessary resources and material which have been extremely useful in my dissertation. I am very much grateful to him for his enthusiastic encouragement and guidance.

Secondly a very special thanks to – **Dr. S. N. Pradhan** (Professor and M.Tech. Co-coordinator) who always inspired to put the maximum of our efforts into dissertation work. He has always been helpful to provide the necessary facilities and also to solve our problems.

I would like to thank **Prof. A. B. Patel** (Director, NIT) and **Prof. D. J. Patel** (H.O.D., CSE Dept.) for supervising the entire dissertation program and organizing meetings in order to receive feedback from students as well as the staff-in-charge regarding the problems faced in the program and their efforts to solve them to their best.

Last, but not the least, I would like to thank all my colleagues at the Nirma University, all active mailing list members of NS2 for all their help and support.

Shailesh Gamit
(05MCE004)

ABSTRACT

For any Multimedia application the Quality of Service (QoS) parameters are the Bandwidth, Jitter and Delay. The task for QoS provisioning becomes more challenging when it comes for the ad hoc wireless networks due to the dynamically changing topology. The Routing protocol available such as AODV, DSR and DSDV does not provide any guarantee to be used for multimedia application. AODV protocol provides single path and is on-demand. It requires periodic updates of the adjacent neighbors. DSDV protocol provides single path and periodic updates whenever there is change in routing table. DSR protocol is based on source routing, suffers from the scalability problem.

The main focus of this new protocol is to reduce the delay that incurred in the AODV and DSR routing protocol and to decrease the packet loss ratio that incurred in the DSDV routing protocol. This protocol is based on on-demand basis. Delay is reduced by establishing more than one path to destination and transmitting the packets to the path where the hop count is less than other available paths. The packet loss ratio is decreased by controlling on the dissemination of control information into the network, which has higher priority than the data packets.

This new protocol is implemented in NS-2. This protocol discovers multiple paths towards destination and the packets are transmitted to path where hop count to reach the destination is less. Delay is also reduced in terms of time needed to find a new path to the destination when the link breaks which occurs in other routing protocols like AODV and DSR.

CONTENTS

LIST OF FIGURES

List of Tables

Abbreviations and Acronyms

ACK	Acknowledgment
AGT	Agent
AODV	Ad Hoc On-Demand Vector
ARP	Address Resolution Protocol
AVG	Average
CTS	Clear-To-Send
DST	Destination
DSDV	Destination Source Distance Vector
DSR	Dynamic Source Routing
ID	Identification
MAC	Medium Access Control
MANET	Mobile Ad Hoc Networks
MAX	Maximum
MIN	Minimum
NS2	Network Simulator 2
QoS	Quality of Service
RTR	Routing
RTS	Request-to-Send
SRC	Source
TCP	Transmission Control Protocol
TTL	Time-to-live

1 INTRODUCTION

The ability to communicate from anywhere and at any time is mankind's dream for a long time. Wireless is the only medium that can fulfill this need. With the recent advances in the technologies and the mobility of the wireless systems it is possible for "anyone, anywhere, anytime" paradigm of mobile ad hoc network to become reality.

1.1 BACKGROUND: WIRELESS MOBILE NETWORKS

Based on the hop distance of packet transfers, wireless networks can be classified into two types: single-hop and multi-hop. The single-hop network generally requires preconfigured, fixed infrastructures. The multi hop network, on the other hand, does not rely on a fixed infrastructure, thus can provide a more flexible service, for example, in a rural area. These two different wireless networks are detailed below.

1.1.1 The Single-Hop Wireless Network

In a single-hop wireless network as shown in Figure 1.1, the whole service area is divided into several smaller service regions called cells [4]. In each cell, at least one base station is allocated to provide network service to mobile hosts in the cell.

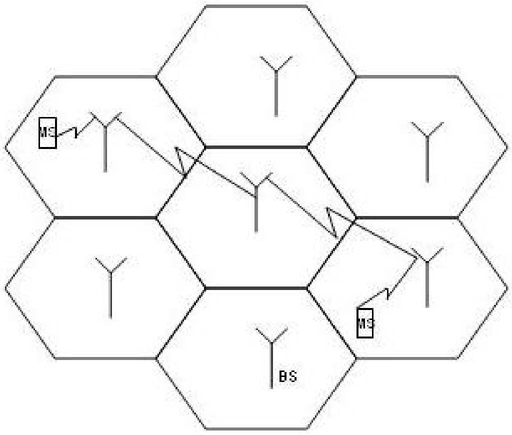


Figure 1.1: Single-Hop Network

The mobile host connects to the network by establishing a wireless connection to the base station. The connections among base stations are usually provided by high speed wired backbone.

1.1.2 The Multi-hop Wireless Network

The drawback of a single hop network is that it requires a pre-established communication backbone, which is infeasible under certain circumstances for example, battle-field, disaster (flood, fire, and earthquake) recovery, search and rescue, or exploration of an unpopulated area, etc [4].

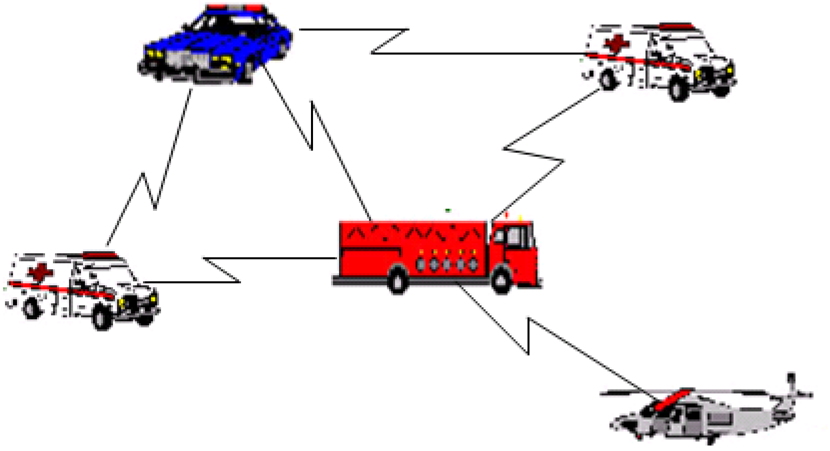


Figure 1.2: Mulithop Wireless Network

Shown in Figure 1.2 Multihop wireless networks are the networks where applications require an instant infrastructure to carry multimedia information. The multi-hop wireless mobile network, also called “ad hoc” network, serves this need because it relies merely on the wireless communication and allow host mobility.

1.2 MOBILE AD HOC NETWORKS

Mobile Ad hoc Networks can be defined as “An autonomous system of mobile routers connected by wireless links. 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 stand-alone fashion, or may be connected to the larger Internet”

Mobile Ad Hoc Networks (MANETs) are self-organizing, and highly dynamic networks formed by a set of mobile hosts connected through wireless links [4] [5]. These networks can be formed on the fly, without requiring any fixed infrastructure or central coordinator that will manage the whole network. Even though ad hoc network works in absence of any fixed infrastructure, recent advances in wireless network architecture reveals interesting solution that enables it to function in the presence of infrastructure. As these are infrastructure less wireless networks, each node should act also as a router. 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 where all network activities including discovering the topology and delivering messages must be executed by the nodes themselves. As a router, the mobile host represents an intermediate node which forwards traffic on behalf of other nodes. If the destination node is not within the transmission range of the source node, the source node takes help of the intermediate nodes to communicate with the destination node. Tactical communication required on battlefields, among a fleet of ships, or among a group of armored vehicles is some of the military applications of these networks. Civilian applications include peer-to-peer computing and file sharing, collaborated computing in a conference hall, and search and rescue operations.

With the evolution of Multimedia Technology, Quality of Service in MANETs became an area of great interest. Besides the problems that exist for QoS in wire-based networks, MANETS impose new constraints. This is due to the dynamic behavior and the limited resources of such networks.

1.2.1 Efficient Wireless Routing

In MANETs it is critical for a routing protocol to consider all the reachability, scalability and connection quality for multimedia services. Therefore, the goals of our research in wireless routing are: firstly, the routing scheme has to be efficient and it should provide continuous connection to the destination node even if the link breaks, and secondly the delay parameter which is required QoS parameter to be considered carefully when focusing on multimedia applications.

1.2.2 Support of Renegotiable QoS

Multimedia applications place stringent requirements on networks for delivering multimedia content in real-time. Compared to the requirements of traditional data-only applications, these new requirements generally include high bandwidth availability, low packet loss rate, and a low variation in packet delivery time [9]. Unfortunately, in a wireless environment, no guarantee on these requirements can be safely made in the fact of mobility. Therefore, in order to maintain same level of acceptable quality over such networks, needed is to take a new look at QoS support.

1.3 ASSUMPTIONS

The assumptions that are to be considered in this thesis are the following

- The nodes are the portable devices with the limited battery life. This feature can impose restriction on the computation and the communication (transmitting and receiving) at every node in the MANET.
- All the nodes have the equal capability of processing power, transmission range and the other features of the device.
- Connection between the nodes is not transitive.
- Nodes are identified by fixed ID's.
- Nodes are free to move in the network without any restriction and can leave or join the network at any time.
- The route availability defined in the thesis is limited to the range of transmission of all the nodes that comes in the path from source to destination.
- All the nodes trust each other by using predefined keys or because it is known that there are no malicious intruder nodes.
- Packets may be lost or corrupted during transmission on the wireless network.
- The routing protocol is tested with the assumption that the mobility is not very high.

1.4 PROBLEM STATEMENT

The transmission of real-time, multimedia or any type of data to the wireless medium introduces many technical obstacles. The protocols available for the wired medium cannot be easily migrated to the wireless network because of the error-prone medium and the mobility of the devices. This is true for Mobile Ad Hoc Networks (MANETs) where mobile devices move in an unpredictable manner and at arbitrary time with random mobility.

Video or multimedia transport over wireless ad hoc networks is a challenging subject, since the wireless links are unreliable and have limited bandwidth. Typical multimedia applications, such as streaming, may require higher reliability connections than that provided by a single link. In a network consisting of mobile nodes, the connection between a source and destination may break down and has to be updated regularly. Although, when a path fails, one could switch over to an alternative path; this may take an unacceptably long period of time, causing a temporary disruption in the multimedia signal. Instead of transporting a multimedia stream through a "single" communication pipe, the stream is split up into multiple sub-streams, each of which takes a separate route through the network. At the destination all sub-streams received properly are merged in a clever way.

In this thesis the network layer protocol for the transmission of multimedia data is focused more. The main idea is to provide continuous connection to the destination even when the path breaks. Continuous connection is provided by establishing multiple paths from source to destination, so that when one of the paths breaks than also other paths are available through which the data can be transmitted.

1.5 THESIS CONTRIBUTION

The Quality of Service is a challenging task for any type of network whether it is wired or wireless. The task becomes even more challenging when it comes to the mobile ad hoc network. The thesis contribution is to provide at least minimum QoS requirements for the multimedia application. Although all the layers are equally responsible for the QoS provisioning for any type of service, the thesis contribution is towards the network layer i.e. the routing protocol of mobile ad hoc network for multimedia application. The brief introduction to the new routing protocol for the multimedia application is given below.

When stared the thesis, the routing protocols that were commonly used were AODV, DSR and DSDV. With the studies of these routing protocols and looking at the advantages and the disadvantages a new routing protocol for multimedia application is to be designed. As the DSR lacks in scalability, and the AODV with the control packets dissemination into the network, a new Multipath routing using the AODV approach for the routing table update mechanism along with route discovery and maintaining the continuous path from source to destination.

1.6 OUTLINE OF THESIS

The remaining chapters are organized as follows. The second chapter starts with the literature survey that consists of design issues and challenges, routing fundamentals and different types of routing used for the MANETs. After this is the evaluation of different routing protocols used for the MANETs are discussed. Next is the performance comparison of the routing protocol with their testbed implementation and results and then at last the concluded part of these routing protocols.

Chapter third includes the various control packet formats that are used in Multipath routing protocol. The new routing protocol implementation, the routing table building up and the timers used in the protocols is defined in the fourth chapter.

The fifth chapter introduces about the NS2 tool that is used for the simulation which covers the background of NS2 and about how to add a new routing protocol in ns2. It also includes the trace format that is used in ns2.

At last the simulation results with the new Multipath routing protocol is studied and the results compared with the AODV, DSR and DSDV routing protocol.

2 LITERATURE SURVEY

This chapter introduces different Issues and challenges in MANETs followed by design choices for QoS support. Next section is about comparison of different multimedia application and than discussed about how routing is done in ad hoc networks. It also shows how the routing is more challenging and different as compared to the static network. After that the different routing protocols that are used for the ad hoc network are described and the difference among these routing protocols is outlined. At last the implementation of these routing protocols with the implementation in NS2 is shown followed by the conclusion.

2.1 ISSUES AND CHALLENGES IN MANETS

Providing QoS support in Ad Hoc Wireless Networks (AWNs) is an active research area. AWNs have certain unique characteristics that pose several difficulties in provisioning QoS [5]. A detailed discussion on how the characteristics of AWNs affect QoS provisioning are given below:

Dynamic topology

The nodes in mobile ad hoc wireless network do not have any restriction on mobility and so the network topology changes dynamically. The admitted QoS sessions may suffer due to frequent path breaks, thereby requiring such sessions to be re-established over new paths. The new path may suffer from the QoS due to the delay incurred in establishing a new path.

Imprecise state information

In most cases, the nodes in an ad hoc wireless network maintain both the link-specific state information and flow-specific state information. The link-specific state information includes bandwidth, delay, delay jitter, loss rate, error rate, stability, cost, and distance values for each link. The flow specific information includes session ID, source address, destination address, and QoS requirements of the flow the state information is inherently imprecise due to dynamic changes in network topology and channel characteristics. Hence routing decisions may not be accurate, resulting in some of the real-time packets missing their deadlines.

Lack of central coordination

Unlike wireless LANs and cellular networks, AWNs do not have central controllers to coordinate the activity of nodes. This further complicates QoS provisioning in AWNs.

Error prone shared radio channel

The radio channel is a broadcast medium by nature. During propagation through the wireless medium the radio waves suffer from several impairments such as attenuation, multi-path propagation, and interference (from other wireless devices operating in the vicinity).

Hidden terminal problem

The hidden terminal problem is inherent in AWNs. As shown in figure 2.1 this problem occurs when packets originating from two or more sender nodes, which are not within the direct transmission range of each other, collide at a common receiver node. Here in the Figure 2.1 the nodes A and C both are in range of B, but, the node B can't here the transmission of A and C that of A. So, when A is transmitting the packets for node C the medium appears to be free so he will also start the transmission causing collision [7].

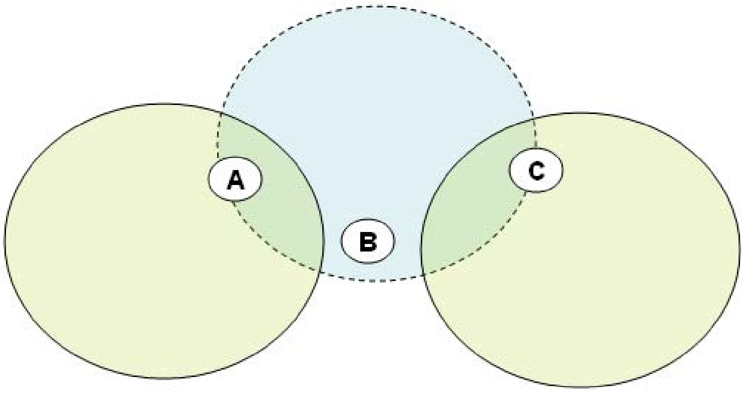


Figure 2.1: Hidden Terminal Problem

It necessitates retransmission of packets, which may not be acceptable for flows that have stringent QoS requirements. The RTS/CTS control packet exchange mechanism, proposed and adopted later in the IEEE 802.11 standard reduces the hidden terminal problem only to a certain extent.

Limited resource availability

Resources such as bandwidth, battery life, storage space, and processing capability are limited in AWNs. Out of these, bandwidth and battery life are very critical resources, the availability of which significantly affects the performance of the QoS provisioning mechanism. Hence efficient resource management mechanisms are required for optimal utilization of these scarce resources.

Insecure medium

Due to the broadcast nature of the wireless medium, communication through a wireless channel is highly insecure.

2.2 DESIGN CHOICES FOR QOS SUPPORT

Since the absolute QoS is very difficult task as far as the MANET is considered, the next subsection introduces what the QoS and the framework for the QoS in MANET.

Hard state vs. soft state resource reservation

QoS resource reservation is one of the very important components of any QoS framework. It is responsible for reserving resources at all intermediate nodes along the path from the source to the destination as requested by the QoS session. QoS resource reservation mechanisms can be broadly classified into two categories, hard state and soft state reservation mechanisms. In hard state resource reservation schemes, resources are reserved at all intermediate nodes along the path from the source to the destination throughout the duration of the QoS session. Due to problems in hard state, soft state resource reservation mechanisms, which maintain reservations only for small time intervals, are used. These reservations get refreshed if packets belonging to the same flow are received before the timeout period. The soft state reservation timeout period can be equal to packet inter-arrival time or a multiple of the packet inter-arrival time. If no data packets are received for the specified time interval, the resources are de-allocated in a decentralized manner without incurring any additional control overhead [6].

Stateful vs. stateless approach

In the stateful approach, each node maintains either global state information or only local state information, while in the case of stateless approach no such information is maintained at the nodes. State information includes both the topology information and the flow-specific information [4][5]. If global state information is available, the source node can use a centralized routing algorithm to route packets to the destination. The performance of the routing protocol depends on the accuracy of the global state information maintained at the nodes. Significant control overhead is incurred in gathering and maintaining global state information. On the other hand, if mobile nodes maintain only local state information (which is more accurate), distributed routing algorithms can be used. Even though control overhead incurred in maintaining local state information is low, care must be taken to obtain loop-free routes. In the case of stateless approach, neither flow-specific nor link specific state information is maintained at the nodes. Though the stateless approach solves the scalability problem permanently and reduces the burden (storage and computation) on nodes, providing QoS guarantees becomes extremely difficult.

Hard QoS vs. soft QoS approach

The QoS provisioning approaches can be broadly classified into two categories, hard QoS and soft QoS approaches. If QoS requirements of a connection are guaranteed to be met for the whole duration of the session, the QoS approach is termed as hard QoS approach. If the QoS requirements are not guaranteed for the entire session, the QoS approach is termed as soft QoS approach [5].

Keeping network dynamics of AWNs in mind, it is very difficult to provide hard QoS guarantees to user applications. Thus, QoS guarantees can only be given within certain statistical bounds. Almost all QoS approaches available in the literature provide only soft.

2.3 COMPARISON OF MULTIMEDIA APPLICATIONS

As the thesis is concerned about the multimedia traffic in the Mobile Ad Hoc Network the key parameters that should be satisfied are specified below in the Table 2.1.

Audio

Conversational voice

Influence by the delay parameter which case echo so tighter delay required. Human ear is tolerant to certain amount of information loss, so packet loss is acceptable but delay not.

Table 2.1 Requirements of different Multimedia Applications [10]

Multimedia	Application	Typical Data Rate	Delay
Audio	Conversation voice	4-64 kbps	<150 ms
Audio	Voice messaging	4-32 kbps	<1s for playback & <2s for recording
Audio	High quality streaming audio	16-128 kbps	<10 s
Video	Video phone	16-384 kbps	<150 ms preferred <400 ms limit
Video	One way	16-384 kbps	<10 s
Data	Web browsing	~10 kbps	Preferred 2 s per page Acceptable 4 s per page

Voice messaging

For voice messaging the information loss is somewhat acceptable as that of conversational voice and also the delay is tolerable since no direct conversation is involved.

Streaming audio

If better quality is expected than packet loss should be low and the delay is even more tolerable than that of voice messaging.

Video

Videophone

Same delay requirements as for conversational voice, i.e. no echo and minimal effect on conversational dynamics. Human eye is tolerant to a certain amount of information loss.

One-way video

No conversational element involved, meaning that the delay requirement will not be so stringent, and can follow that of streaming audio.

Data

From a user point of view, the prime requirement for any data transfer application is to guarantee zero loss of information. Delay variation is not generally noticeable to the user, although there needs to be a limit on synchronization between media streams in a multimedia session

2.4 ROUTING

The bandwidth reservation and real-time traffic support capability of MAC protocols can ensure reservation at the link level only, hence the network layer support for ensuring end-to-end resource negotiation, reservation, and reconfiguration is very essential [5] . QoS is very difficult in ad hoc network because of the dynamic network topology and the wireless medium that is shared by many nodes. In this section various solutions for providing Quality of Service in ad hoc wireless network is defined.

QoS routing protocol

The target of QoS routing protocol is to find the path in the network from source to destination which satisfies the QoS requirement for each of the connection in the network and optimizes the use of the network resources. The information regarding the feasible path resides within the routing protocol and by that information best path from source to destination which meets the required QoS in identified.

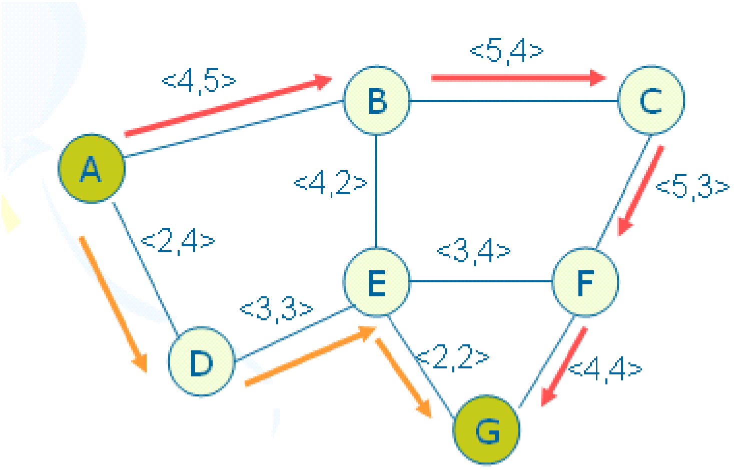


Figure 2.2 Network with <BW, Delay> tuple

In the Figure 2.2 the suppose the Quality of Service requirement is to be maintained for the connection having minimum bandwidth of 4 then the path from the source A to the destination G is A-B-C-F-G. If the QoS requirement for the particular connection is having bandwidth requirement of less than 4 but the delay constraint is to be meet with maximum delay of 10 than the path from source to destination is A-D-E-G. Thus the QoS in ad hoc network is based on the requirement of the particular connection.

The network is the layer in ad hoc network that do all the functions regarding the reservation of bandwidth and the other parameters like end-to-end delay negotiation and depending on the requirement the path is establish in the network.

To assist QoS routing, the topology information can be maintained at the nodes of AWNs. The topology information needs to be refreshed frequently by sending link state update messages, which consume precious network resources such as bandwidth and battery power. Otherwise, the dynamically varying network topology may cause the topology information to become imprecise. This trade-off affects the performance of the QoS routing protocol. As path breaks occur frequently in AWNs compared to wired networks where a link goes down very rarely, the path satisfying the QoS requirements needs to be recomputed every time the current path gets broken. The QoS routing protocol should respond quickly in case of path breaks and recomputed the broken path or bypass the broken link without degrading the level of QoS. The different routing protocols that can be used for providing QoS in ad hoc network are described below.

Mobile Ad Hoc Networks are characterized by constantly changing network topology and the absence of central coordinator. The absence of central coordination results in the node to be acted as the router and all other functions required from transmission to receiving. The nodes maintains the routing information into its table and on its basis route the packets to its destination. Since each node in the ad hoc wireless network has limited range of transmission so in order to communicate with the other nodes that are not within the direct range, it needs to enlist the aid of the other nearby nodes in transmitting the packets. Thus routing is done in hop-by-hop manner i.e. the communication is done by multiple nodes that acts as the intermediate node from any source to destination.

Routing is the key to efficient operation of MANETs. For this efficient operation routing is the one of the main operation that can handle different challenges of the MANETs such as the mobility pattern, time varying topology, imprecise state information, bandwidth constrained links, scalability of the network and many more.

The routing even becomes more challenging when the network grows in size, and the when the problems such as the increasing node density and large number of nodes which increased the network size. The node density causes to excessive transmission of control packets in the network and the network size causes the routing table maintenance problems. It is not true that the routing protocol which is more scalable is more efficient, it may not work well with the mobility pattern and the one which is susceptible to mobility may not be successful with the scalability.

Thus for the efficient operation of the MANETs the following features are required in the routing protocol.

- It should be scalable to increase the reliability and availability i.e. to reduce the chance that any node is isolated from rest of the network.

- Adaptive routing algorithm should be used for adjusting with the frequent changes in the topology, radio propagation and network conditions [11].
- The routing protocol should be of low overhead because of the scarce resources of the wireless network.

2.5 ROUTING METHODS

Each node in the mobile ad hoc networks maintains a preferred neighbor for the destination through whom it transmits the data packets. The forwarding of these packets continuous until the packet reaches the destination. The manner in which the routing table are constructed, updated or deleted of the entry differs from one routing method to another. The thesis mainly concerns about the next hop routing mechanism so the methods for the next hop routing methods are described.

2.5.1 Link-State

The link state routing method each node maintains a view of network topology and the associated cost of each link. To keep this information consistent each node periodically broadcast its link conditions information to other nodes using a protocol called flooding [11]. The receiving node uses this information and finds a shortest path to the destination.

2.5.2 Distance Vector

In distance vector routing method each node x maintains a routing table for the entire destinations d in the network. The other information is the cost to the destination and the next hop to reach the destination. To maintain the information up-to-date the nodes continuously broadcast its information into the network. This method is the simplest one and more efficient and required less storage space than other routing methods. The only problem which it faces in case of MANETs is the dynamic topology. Thus, it is required that the network information is to be continuously broadcasted into the network so the node maintains and updates the consistent information.

2.6 ROUTING PROTOCOLS

In this section three different routing protocol AODV, DSDV and DSR are studied and compared the performance. All these routing protocols are for the MANETs. The operation of these routing protocols differs in their routing algorithm and the maintenance of the routing table. This protocol has been proposed for solving the multi-hop problem in ad hoc wireless network. There are two types of routing differ in routing table building for the ad hoc network.

Table Driven

In this type of routing protocol consistent routing information of the entire destination is stored in the routing table. This type of routing protocol does not scale to large network because of the routing table limitation and updating of the routing table in the network where nodes continuously change the position [5] e.g. DSR.

On-demand

This type of routing protocol the route to the destination is updated only at the time of need. In this the source node request for the destination and when the request reaches to the destination the node reply back. When the reply reaches the source node update its routing table and packets are transmitted on that route [5] e.g. AODV

2.6.1 AODV

AODV (Ad Hoc On-demand Routing Protocol) uses the hop-by-hop routing. As shown in Figure 2.2 the node that wants to know a route to a given destination generates a ROUTE REQUEST. The route request is forwarded by intermediate nodes that also create a reverse route for itself from the destination. When the request reaches a node with route to destination it generates a ROUTE REPLY containing the number of hops requires reaching destination. All nodes that participate in forwarding this reply to the source node create a forward route to destination. This state created from each node from source to destination is a hop-by-hop state and not the entire route as is done in source routing.

The AODV routing algorithm is a source initiated [12], on demand routing algorithm Therefore a route is discovered only if and when a source wants to send data to a specific destination. Once the route is established between the source and the destination, it remains as long as it's needed for further communication.

One of the main features of AODV routing protocol is that it uses traditional routing tables to maintain the routing information. It maintains one entry for each destination in the network it requests. The routing information is update on on-demand basis. For each entry in the routing table it has stored has the sequence number that prevents it from the looping problem.

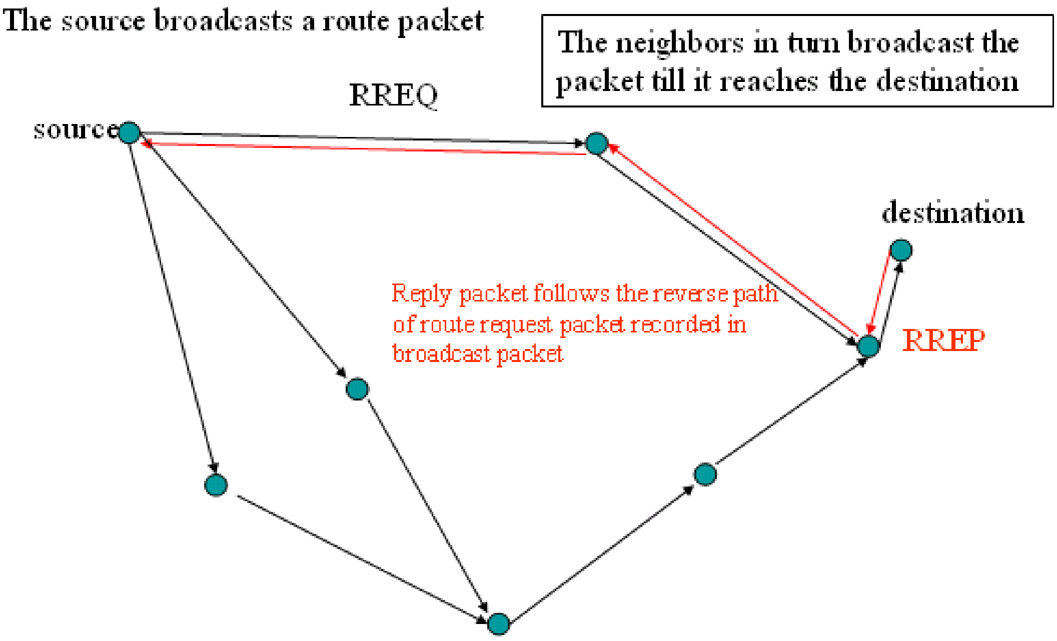

The source broadcasts a route packet
The neighbors in turn broadcast the packet till it reaches the destination
RREQ
source
destination
Reply packet follows the reverse path of route request packet recorded in broadcast packet
RREP

Figure 2.3 AODV Routing Protocol

For preventing this looping problem requesting node always chooses a route with the greatest sequence number to communicate with its destination node. Once a new path is found, a RREP (Route Reply) is sent back to the requesting node. AODV also has an important feature that informs nodes of any possible link breaks that might have occurred. The routing protocol maintains a time-base state in each node, regarding utilization of individual routing table entries. A table entry is deleted if not used recently. A set of predecessor nodes is maintained for each routing table entry, indicating the set of neighboring nodes, which use that entry to route the data packets. These nodes are notified when the next-hop link breaks with RERR (Route Error) packets. On the receiving of these packets each predecessor node, in turn forwards the RERR packets to its own predecessors, thus erasing all routes with broken links AODV is designed to inform all sources using a given route when link failure occurs

In AODV routing protocol the source first find the destination and than sends the packets with the same route even if the node is first five nodes away reaches to it until the route breaks. The drawback found is that even if the shortest path is available after the route request i.e. when the destination node comes nearer to the source node the packets are transmitted through the longer route.

2.6.2 DSDV

In DSDV (Destination Source Distance Vector) routing protocol routing messages are exchanged between neighboring mobile nodes (i.e. mobile nodes that are within range of one another). DSDV is a proactive or table driven routing protocol. That is the protocol maintains a correct route to any node in the network. The DSDV routing algorithm is based on the idea of the classical Bellman-Ford Routing [14], with some major improvements to make it suitable for wireless schemes and specifically solve the count-to-infinity problem. The main idea for the routing table updates is based on the time at which the routing information is to be sent to the other nodes. In this protocol the information is disseminated based on the time at which the routing table was updated since previous update. The protocol uses a sequence number for each routing table entry to distinguish stale routing information from new routing information, and thus avoids looping. The nodes communicate with each other to update their routing tables. The update is both time-driven and event-driven. That is, the nodes periodically transmit their routing tables to their neighbors. A node also transmits its routing table if a significant change has occurred in its table since the last update was sent. Any routing updates may be triggered or routine. Updates are triggered in case routing information from one of the neighbors forces a change in the routing table. The received routing information is also broadcasted by the nodes after adding one hop count more to the already hop count number. This is so because the receiving nodes know the cost to reach the destination. A packet for which the route to its destination is not known is cached while routing queries are sent out. The packets are cached until route-replies are received from the destination. There is a maximum buffer size for caching the packets waiting for routing information beyond which packets are dropped.

2.6.3 DSR

The Dynamic Source Routing protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. The protocol is composed of the two mechanisms of Route Discovery and Route Maintenance, which work together to allow nodes to discover and maintain source routes to arbitrary destinations in the ad hoc network [14]. As shown in Figure 2.4 the use of source routing allows packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use. All aspects of the protocol operate entirely on-demand, allowing the routing packet overhead of DSR to scale automatically to only that needed to react to changes in the routes currently in use.

DSR, like AODV falls under the family of On-Demand routing protocols. That is, it also discovers routes only when needed by the source. Unlike AODV, DSR doesn’t use traditional routing tables to maintain routing information. The key characteristic of DSR is the use of source routing and route cache. That is, the sender knows the complete hop-by-hop route to the destination. When the packets are transmitted the complete route information is stored in the packets so, at the receiving the nodes do not have to finds the route for this packet, the packet are normally transmitted by looking into the routing table. These routes are stored in the route cache. The data packets carry the source route in the packet header. When one host sends a packet to another host and does not know the route to the destination, it broadcasts a route request packet to dynamically discover the route to the destination.

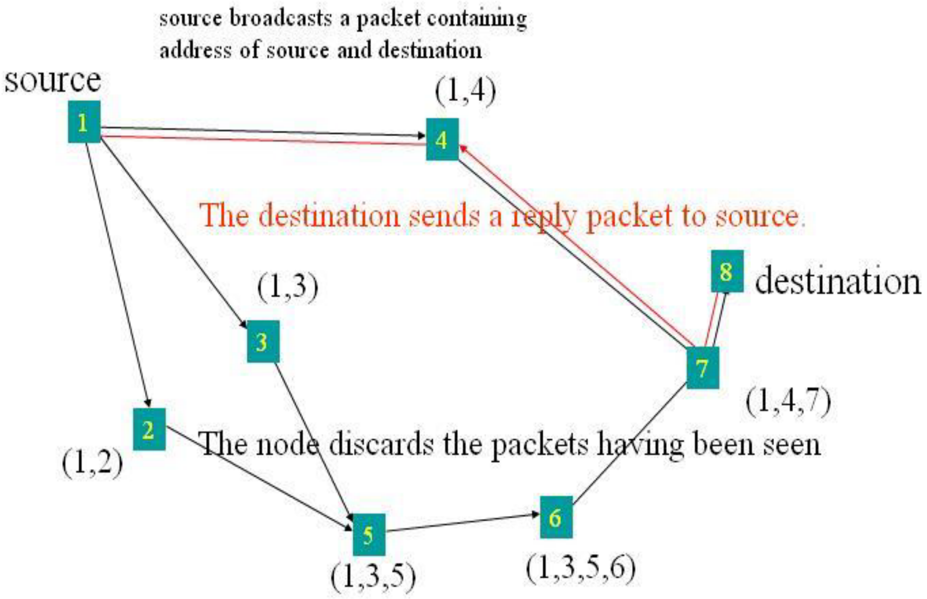

source broadcasts a packet containing address of source and destination
source
1
(1,4)
4
The destination sends a reply packet to source.
(1,3)
3
8
destination
7
(1,4,7)
2
(1,2)
The node discards the packets having been seen
6
5
(1,3,5)
(1,3,5,6)

Figure 2.4: DSR Routing Protocol

The route discovery mechanism works by flooding the network with RREQ packets. When a node receives a RREQ it forwards it, unless the node is the destination or it has knowledge of a route to the destination. If the node happens to be the destination or knows the route to the destination, it replies with a RREP packet that is routed back to the original source. In this process RREQ and RREP packets are both source routed. Unlike the AODV no special mechanism is needed to detect the routing loops.

2.7 COMPARISON OF PROTOCOLS

The difference between the three protocols can be seen in the results:
The scenario for the network is given below:

- Area = 670 x 670 meters
- Node movement = Random
- Packet size = 512 bytes for constant bit rate
- Number of connection = 8

Table 2.3: Comparison of AODV, DSR and DSDV Routing Protocols

Parameters	**AODV**	**DSR**	**DSDV**
No. of Nodes	30	30	30
Mobility Max	20	20	40
Packets Type	CBR	CBR	CBR
Packets Sent	3577	3585	3578
Packets Received	3566	3577	2103
Packets Dropped	11	7	1467
Efficiency(%)	99.69	99.77	58.77
Avg Delay(ms)	27.4615	112.706	15.4584
Min Delay(ms)	5.449	5.4495	5.449
Max Delay(ms)	1985.16	7963.78	216.192

As shown in Table 2.3 the comparison is mainly on the throughput and the delay incurred during the packets transmission. The simulation testbed is of area 670 x 670 and the number of nodes used during the simulation is 31. There are eight (8) different connections starting and ending at different interval of time. The nodes mobility during simulation ranges with time but the maximum speed that is defined is 20 m/sec. The packet type used is CBR (constant bit rate). For the multimedia application if we want to compare than the packet type must be VBR (variable bit rate). Since at the tool used was the NS2 (see chapter 5) which at present does not support for the VBR data, so, CBR data rate was used in simulation.

From the results it can be clearly seen that the maximum packets are dropped in DSDV protocol and than comes the AODV followed by DSR. In DSDV the packet dropped are 1467, in AODV with 11 and DSR with 7 only. So it can be concluded that the application where packet loss constraint is present it is better to use DSR or AODV protocol than using DSR protocol. The packet drop in DSDV routing protocol is due to the control information of the network that is transmitted more than that of the AODV and DSR routing protocol.

In case of AODV and DSR routing protocol the control information is transmitted periodically i.e. at regular interval of time. So the control information is less that that in DSDV routing protocol. This control information at the intermediate node dropped the data packets and so the efficiency is less in DSDV routing protocol.

Delay

In case of delay variation in DSDV routing protocol is because the routing table is the most updated in this protocol. The control information that is transmitted regularly in the network for the DSDV routing protocol contains the routing table information. This information is sent to all the nodes in the network and so the path to the destination is the shortest path form the source to destination. Due to this the delay is less in DSDV routing protocol.

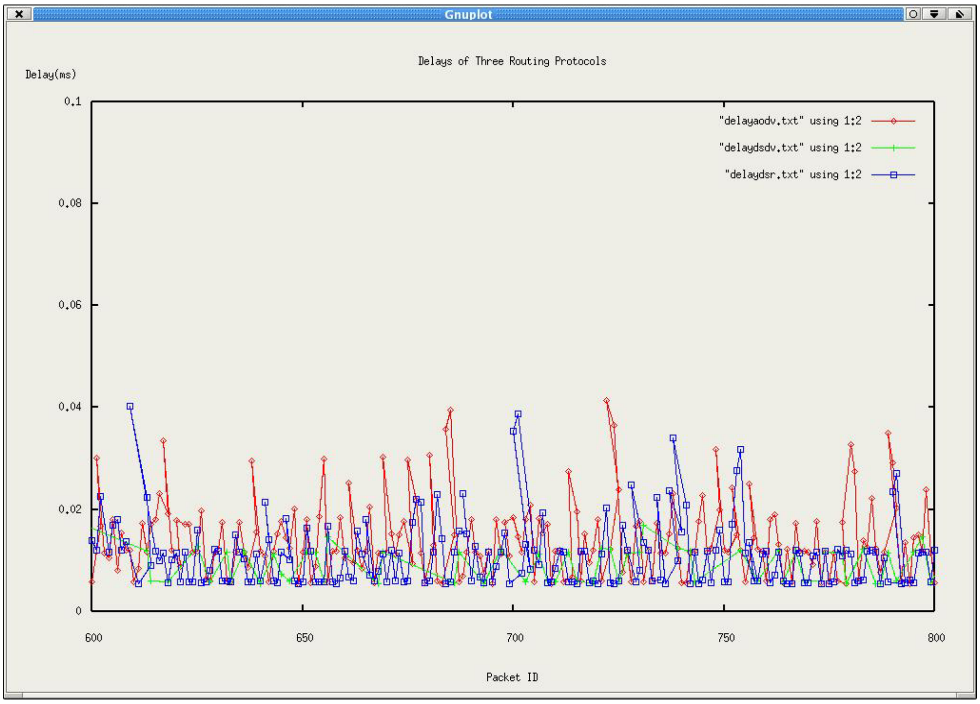


Figure 2.5 Delay of all AODV, DSDV and DSR routing protocols
Note: Red: AODV, Green: DSDV and Blue: DSR

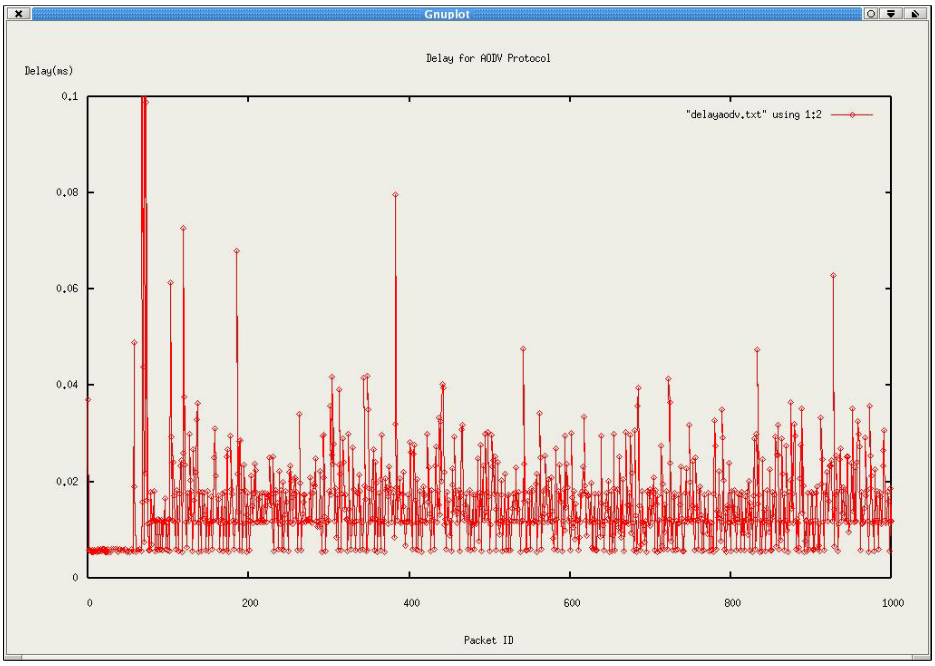


Figure 2.6: Delay of AODV Routing Protocol

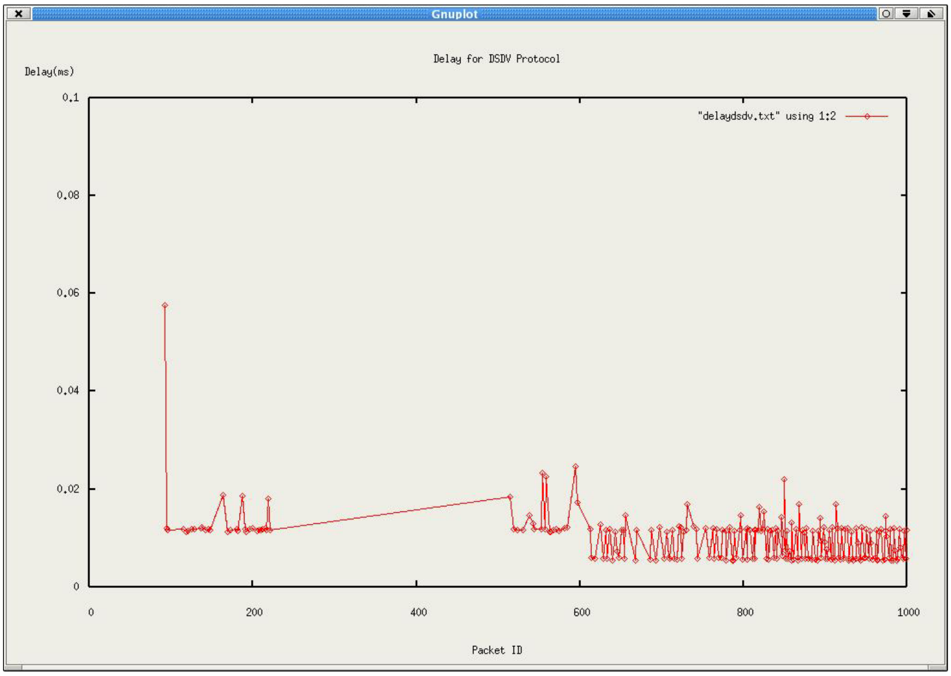


Figure 2.7: Delay of DSDV Routing Protocol

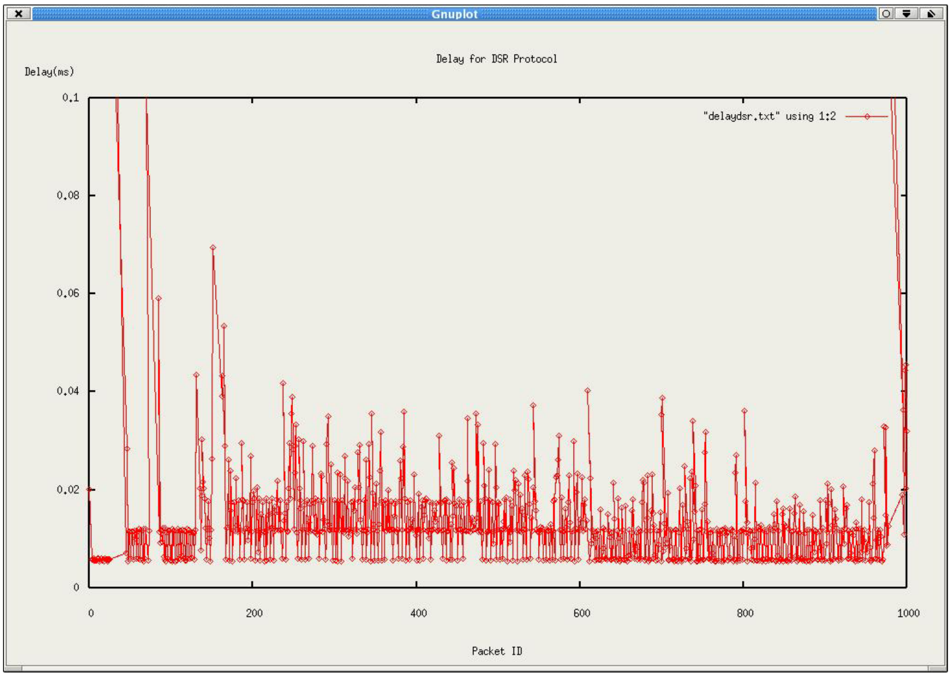


Figure 2.8: Delay of DSR Routing Protocol

2.8 CONCLUSION

From the results it can be concluded that the application whose packets loss ratio are more stringent to QoS than DSDV cannot be used. In that case either AODV or DSR can be used where packet loss is very less. In case if the application that are more stringent to delay for example conversation voice, where some of the packet loss doesn't make any difference than DSDV protocol can be used. From the conclusion it is required to have a new routing protocol that is having the packet loss ratio less than that of DSDV routing protocol and the Delay parameter to be less than that of AODV and DSR routing protocol.

The delay and the packet loss that is seen in these protocols is due to the control information disseminated into the network. AODV and DSR routing protocols have less number of control information disseminated into the network and the DSDV has more number of control information. This control information has more priority at the intermediate node that acts as the routers so the data packets are dropped at these nodes. In case of delay, the routing table of DSDV routing protocol is updated whenever there is change in routing table at any node. So the path for any node to reach the destination is the shortest.

3 PROJECT DETAILS

3.1 BACKGROUND

Any multimedia application to transport over ad hoc wireless network is a very challenging task because of the unreliable links. There is no guarantee for the link to remain stable for the required time period. In case of wired networks the links are fixed so we can trust on the links. The unreliability is due to the frequent path breaks due to the dynamic mobility of the nodes and the dynamic topology created.

Typical multimedia applications such as streaming require higher reliability connection than that provided by the single link. In a network consisting of mobile nodes, the connection between a source and destination may break down and has to be updated regularly. When the links breaks the time is needed to setup a new path for the destination. This time requirement is unacceptable in cases where destination is far away from the source node and it will temporarily disrupt the multimedia transmission. Instead of this single path if multiple paths are maintained than the time needed to setup a new path is in parallel to the packets transmission. Only the time is requiring during the first search. So the time for the source to wait for the data transmission is reduced.

Unlike traditional protocols that will choose a single path through the network, the protocol used must deliver multiple paths from a source to a destination. Also, the Quality of Service (QoS) aspects of each path (e.g. delay, bandwidth, and cost) must be taken into account. Finally, care must be taken that the different routes do not share nodes other than the source and destination nodes.

The advantages of this are given below;
First, Multipath transport distributes traffic load in the network more evenly. For example, a large burst of data, e.g., an Intra or I video frame, can be partitioned into several smaller bursts, each transmitted on a different path. A high rate video can be partitioned into several sub-flows, each with a lower rate and sent on a different path. Such balanced load results in less congestion inside

the network. Thus the video packet losses caused by router buffer overflow can be effectively reduced.

Second, Multipath transport provides a larger aggregate capacity for a multimedia session. In an ad hoc network, since the available link bandwidth may be limited and time varying, a high rate flow may not find enough available capacity on a single path. With Multipath transport, the flow can be partitioned into several thinner sub-flows, each of which can be accommodated by a path.

Third, if a set of disjoint paths are used in Multipath transport, losses experienced by the sub-flows may be independent to each other. When a path is down because of a link failure, which happens more often in an ad hoc network than in a wired network, it is likely that some other paths are still in good condition. Thus the receiver can always receive some data during any period.

To summarize, the use of Multipath transport for real-time multimedia applications in ad hoc networks can effectively reduce packet losses, provide better scalability, and provide un-interrupted display of video even with the presence of frequent link breaks.

The protocol must provide multiple, loop-free, (preferably) node-disjoint paths from a source to a destination. Since multimedia streaming is primary goal, the multiple routes need to be used simultaneously.

3.2 PROTOCOL OVERVIEW

The protocols for ad hoc wireless network are different from that of the wired or static network. The main difference is due to the dynamic topology. There is no central coordinator or administration that maintains all the nodes in the network. The routing is to be done by the node itself and the information related to the network is also to be transmitted and maintained by the nodes. So the node itself acts as the router and transmits the packets towards the destination by looking into the routing table.

The new Multipath routing protocol is based on the approach used in AODV i.e. on-demand basis and it reactive routing protocol. In this protocol different control messages are transmitted before the actual transmission takes place. The control information transmitted by the nodes are the REQUEST, REPLY, ERROR and the HELLO packets and all the packets are used for different purpose.

On-Demand Routing

In the on-demand routing scheme, a node builds up a route by flooding a query to all nodes in the network. The request packet "picks up" the IDs of the intermediate nodes and stores them in a path field. On detecting the query, the destination or any other node that has already learned the path to destination answers the query by sending a "reply" response packet back to the sender. Since multiple responses may be produced, multiple paths may be computed and maintained. After the paths are computed, any link failure will trigger another query/response so the routing can always be kept up to date. They introduce excessive control overhead since they require frequent flooding, especially when mobility is high and traffic is dense and uniformly distributed. As a result, on-demand routing protocols are only suitable for wireless network with high bandwidth, small packet transmission delays.

3.3 PACKET FORMATS

The different packet format used in the Multipath routing protocol are describe below with their field types. These packets are called the control packets and are not used in the actual data transmission. These packets are used only for the management of the network at each node. All these packets have different formats and functions are different for all of them. In this section all the packets are described briefly.

3.3.1 Hello Packet

Hello packets format is as shown in Figure 3.1. These packets are broadcasted by every node in the network at regular interval of time. The Hello packets are usually used to find out the neighbors in the networks by the nodes. The neighbors are used during the Request for any destination. The hello packet contains the information of the sent node address and the time to live field. Hello packets have the lifetime of only 1, so as soon as it is received by the neighbors it extracts the required information and than the hello packet is dropped. The hello packet when received by the node it first finds the information about which node has sent the packet, after that it adds the address of the sent node in the neighbor list of the receiving node. If the information is already in the receiving node it simply updates the information of expire time of the neighbor. When expire time reaches its limit, the neighbor is automatically deleted which is handled by the other timer. Expire timer is required in the ad hoc network because of the node mobility.

```
 0               1             2               3               4
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |           Reserved            |   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator IP address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Lifetime                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.1: Hello Packet format

The following are the fields that are used in the hello packet:

- Type: - The type field specifies the type of packet i.e. whether is it hello, request, reply or other control packets.
- Hop Count: This field specifies the number of hop it has to traverse in the network. It is set to 1, so that it is received only by the neighboring node and by receiving this hello packet it updates its neighbor list entry.
- Sequence Number: This field is set so that the node receives the update packets. When the node receives the hello packet, it compares it with the latest sequence number it has received. If the sequence number of the received packet is less than the previously received sequence number than the information is not valid.
- Originator IP address: This field specifies the originator of the hello packet. By this the receiving node identifies the neighbor and adds in the neighbor list.
- Lifetime: This field is set to 1 so that the packet does not travel in the network.

3.2.2 Request Packet

The protocol is on-demand routing protocol, so the path from the source to destination is searched only when there is data to send. Whenever any node wants to send data packets it first waits until the path is found to reach the destination.

```
 0                   1                   2                   3                   4
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |            Reserved           |   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            RREQ ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination Sequence Number                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator IP Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Originator Sequence Number                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.2: Request Packet format

Figure 3.2 shows the packet format of Request packet. The fields of the request packet are described below:

- Type: - The type field specifies the type of packet. For this it is "Request" type
- Hop Count: This field specifies the number of hop it has to traverse in the network. It is set to 1, so that it is received only by the neighboring node and by receiving this hello packet it updates its neighbor list entry.
- RREQ ID: This field describes the ID that is unique for the entire route that is request. .
- Destination IP address: The destination IP address for which the request is to be made.
- Destination Sequence Number: The latest sequence number received in the past by the originator for any route towards the destination.

- Originator IP address: This field specifies the originator of the hello packet. By this the receiving node identifies the neighbor and adds in the neighbor list.
- Originator Sequence Number: The current sequence number to be used in the route entry pointing towards the originator of the route request.

When path to the destination is found by the REQUEST packet, the reply packet is sent by the destination node towards the source node. The source node sends the request packet by broadcasting the request packet to all its neighbors. The number of request is dependent on the number of neighbors it has in the neighbor list, which is filled by the hello packet.

When the request reaches to the neighbor, they broadcast this request into the network. All the intermediate nodes that come towards the destination also broadcast the request until it reaches the destination.

When the request comes to the intermediate node, the node adds the reverse path into the routing table of its own. This reverse route can then be used by the reply message that will come from the destination. The intermediate node if received the request from the same source, it will discard the request and drop the packet.

3.3.3 Reply Packet

REPLY packet as shown in Figure 3.3 is sent when the REQUEST packet reaches to destination node of the source request. Reply packet replies to all the neighbors from which it has received the route request with taking into consideration that the reply is not to be sent for the request ID whose reply has been sent.

```
  0                   1                   2                   3                   4
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |          Reserved               |   Hop Count   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                            RREP ID                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                   Destination Sequence Number                   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     Originator IP address                       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP address                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                            Lifetime                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.3: Reply Packet Format

- Type: - The type field specifies the type of packet. For this it is "Reply" type
- Hop Count: This field specifies the number of hop it has to traverse in the network. It is set to 1, so that it is received only by the neighboring node and by receiving this hello packet it updates its neighbor list entry.
- RREP ID: This field is the ID for the route request for whom the reply is sent.
- Destination IP address: The destination IP address for which the request is to be made.
- Destination Sequence Number: The latest sequence number received in the past by the originator for any route towards the destination.
- Originator IP address: This field specifies the originator of the hello packet. By this the receiving node identifies the neighbor and adds in the neighbor list.
- Life time: The time for which nodes receiving the Route Reply consider the route to be valid.

If the reply packet is received by the intermediate node it simply forwards the packet by looking into the routing table which it had developed during the route request. The route reaches the destination; the destination node sends the data that it has queued.

3.4 ROUTING TABLE

Multipath routing protocol deals with the routing table management. The routing table shown in Figure 3.4 is to be updated when even the packets reaches to the node. The different fields that are used in the routing table are described below.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Dest_IP |  ID |  Seq_No |  Hops   | Next_Hop  | Expire_Time | Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.4: Routing Table Format

- Dest_IP: - Destination IP Address. This is 32-bit unsigned integer which identifies the destination address and the port number for the routing table entry. For the ad hoc network it is set to 255 for the ad hoc network.
- ID: - Broadcast ID which is unique to every route. It is 32-bit unsigned integer which identifies the unique route from the many routes available to the destination
- Seq_No: - Sequence number of the route, which is used for update. It is 32-bit unsigned integer. This field is used to identify the updated packet that is received by the nodes.
- Hops: - This field identifies the number of hops to reach the destination. It is unsigned 8-bit integer.
- Next_Hop: - This field identifies the next hop to where it has to forward the packet when the destination of address of the packet is the first field i.e. the Destination_IP in the network to reach the destination. It is 32-bit address and the port number and port address is always 255.
- Expire_Time: - Expire time of the route after which the route is no longer valid. It is 64-bit floating-point number. When this time reaches the routing table entry is deleted.
- Flags: - This is used for whether the route is UP or DOWN; If the route is UP than the packets can be forwarded to this route else if DOWN than can't. It is 8-bit unsigned integer.

Multipath routing protocol deals with the routing table management. The routing table is to be updated whenever the packet reaches to the node.

3.5 ID MANAGEMENT

The ID table shown in Figure 3.5 is maintained at each node. This table is update\added at each time of request\reply packet received at the node. The fields used this table are described below.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Source    |    Destination      |      ID     |    Reply Sent    |
|-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.5: ID table format

- Source: Source address of the request
- Destination: Destination address of the request
- ID: Unique ID that is sent at each request by the source node
- Reply Sent: a Boolean variable that is set false when the request is sent and update to true when the reply is received by the nodes. The nodes here are the source, intermediate and the destination.

4 IMPLEMENTATION DESIGN FOR PROTOCOL

The protocol in on on-demand basis and so the path to the destination is not already available when there is data to send. For each route the nodes that actively participates into the forwarding of the data packets maintains the ID table and the routing table. The ID table has the format as show in Figure 3.5. The protocol is routing protocol so the main idea of the protocol is how to forward the packets to the destination without knowing where it is.

This chapter will cover about how will the source node know where the destination node is and the how the packets are transmitted to the destination node. The request is made and the processing is done at all the nodes. It will also cover how the ID table entries are to be made and how the routing tables entries are that are made upon receiving the request and the reply packet.

4.1 MAINTAINING THE NEIGHBORS

In the distance vector routing algorithm the neighbor's list are maintained and the packets are transmitted by multi-hop using the neighbor information. This neighbor information is not permanent because of the dynamic nature of the mobile ad hoc network [4]. So the information that is in the neighboring table is to be updated periodically. The periodic updates of these neighbors are done with the hello packets. The hello packets are broadcast by every node in the network with the ttl set to 1. The ttl value of 1 indicates that the hello packet should not travel in the network for more than 1 hop. As shown in the figure 4.1 the hello packets are transmitted to all the neighbors which are in the direct range of the node 1 and than the table for the neighbors at different nodes is shown in the figure 4.1

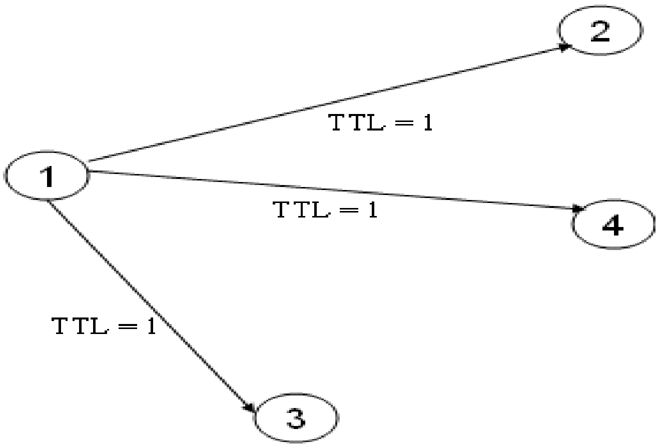


Figure 4.1: Hello packet transmission

Below is shown the neighbor information that is maintained at different nodes when each of the neighbors broadcast the hello packets.

As shown in Figure 4.1 the hello packets when broadcast by all the neighbors the Neighbor list are updated at each receiving node. As shown in Figure 4.1 the hello packets with TTL field 1 is broadcast so that only neighbors can receive the hello packets. After receiving the hello packet the receiving node extracts the source of the hello packet and updates the entry in its neighbor list. The update is done with expire time if the neighbor is already in the receiving node or else if not than the neighbor along with expire time is added in the neighbor list.

Table 4.1 Neighbor Table for node 1 when receives hello packets form all the nodes

Neighbors	**Expire**
2	Expire time
3	Expire time
4	Expire time

4.2 REQUEST FOR ROUTE

The request is made by the node whenever the node does not find any entry or the routing table entry for the destination to which it has to send the packets are not more than one so, we can have more than one route at any time. This topic is about how to handle the request at the source, intermediate and the destination node when the request packet is to be sent, the routing table and ID table entries made at these nodes. As shown in the Figure 4.1, the source node is the node-1, the intermediate nodes are the nodes 2, 3, 4, 5, 6 and the destination node is the node-7.

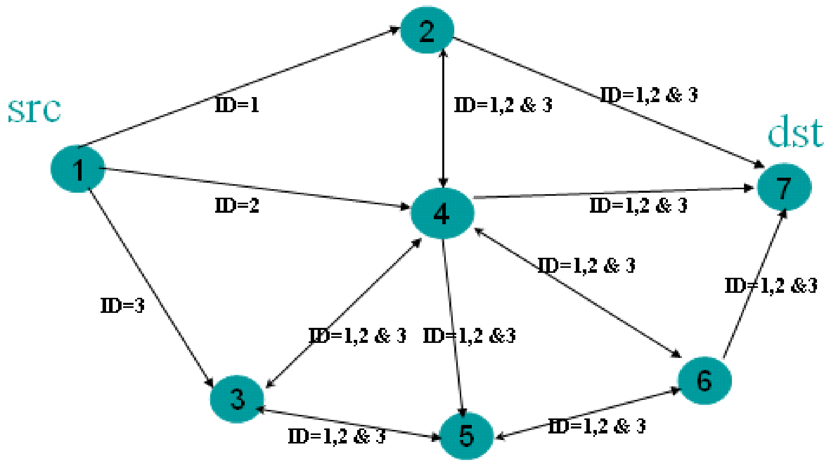


Figure 4.2: Request made by the nodes

At Source Node

As shown in Figure 4.2 when the source node (node 1) receives the data, it broadcast a route request message to toward the destination and the data are temporarily queued into the queue. When it is the source node than the request is more than one otherwise a single request is done by the intermediate node. The number of request depends upon the number of neighbors that the source node has, so, the number of paths towards the destination are never more than the number of neighbors. As shown in Figure 4.2 the source node sends three different requests with different ID's. The request from the source is followed by the unique ID which identifies the route through which the packets has to traverse. The ID is unique for the nodes only i.e. the same ID can be for the other nodes. The ID field is identified by the source and destination pair. The ID table is updated by the source address, destination address, the unique ID that was generated by the source node and reply_sent variable that is temporarily set to false until the reply is received.

Table 4.2: ID management at node 1 when the first request sent

Source	Destination	ID	Reply_received
1	7	1	False
1	7	2	False
1	7	3	False

Table 4.3: Routing Table at node 1 when the first request made

Destination	No. of Hops	Next hop	Expire Time	Sequence Number	ID	Flags
7	Infinity	2	Route time	Current Seq_No	1	Down
7	Infinity	4	Route time	Current Seq_No	2	Down
7	Infinity	3	Route time	Current Seq_No	3	Down

In addition to the ID table, the routing table entry is also made into the routing table. The destination address of the request packet is added into the routing table destination address with the next_hop is set to the IP_BROADCAST and the hop_count is set to INFINITY. The routing table is again updated when the reply is received back sent by the destination node.

Intermediate Nodes

As shown in Figure 4.2 when the intermediate nodes (nodes 2-6) receive the request packets, the packets are broadcast into the network, if it is not the destination address. This packet is broadcast until the TTL of the IP header field reaches to zero. The TTL field is set to the maximum number of nodes it has to travel to reach to the destination. When the TTL field reaches zero the packet is dropped. The forwarding is done based on the ID that each node stores into its ID table. When the node receives the request packet it extracts the ID field from the request packet, the source and destination address of the request and the recv_reply is set to false. The recv_reply is the Boolean variable is set to true only when the reply is received for the same request. By this recv_reply when true the intermediate node does not accept any request from the source whose recv_reply is true by looking into its ID table, so that intermediate node will not become the bottleneck for this source-destination pair.

Table 4.4: ID management at node 2 when the first request received and forwarded

Source	**Destination**	**ID**	**Reply_received**
1	7	1	False

Table 4.5: Routing table at node 3 when the request received

Destination	**No. of Hops**	**Next hop**	**Expire Time**	**Sequence Number**	**ID**	**Flags**
1	1	1	Route time	Current Seq_No	1	UP
7	Infinity	IP_BROADCAST	Route time	Current Seq_No	1	Down

The intermediate nodes after receiving the request checks if it is the destination node, if it is the destination node than it will reply by the reply packet or otherwise not the destination it will simply forward the packet.

The intermediate node in addition to receiving the request packet it also adds the routing table entry in the routing table. The routing table entries made are two. The one is for the destination nodes as the original request destination with the flags set to DOWN and the other entry is the packet from where the request comes. For the second entry the flags are set to UP because the route is available as the packet comes from that source.

Destination Node

The destination node (node 7), shown in Figure 4.2 after receiving the request identifies itself as the destination node. The destination node than looks for the ID field in the ID table entry for the same source-destination pair that it has received in the request. If it finds the entry in the ID table with the reply sent as true then it will discard the request and drop the packet. If no entry is found in the ID table with the source-destination pair than it will send a reply back to the source of the request.

The routing table entry is made for the node from which it received the request and the flags are set to UP.

From all the route request at the source, intermediate and the destination nodes the number of ID's that reach to the destination are three and the different paths available to the source are not always three because a single node cannot be selected for the more than one route from the same source-destination pair. This is better explained in the reply section.

4.3 REPLY FOR REQUEST

The reply packet is sent by the destination node of the request packet. This topic will cover the part which concerns about the reply packet. It will cover about how reply packet is generated and processed at the different node i.e. the destination (node 7), Intermediate and at the source of the requesting node.

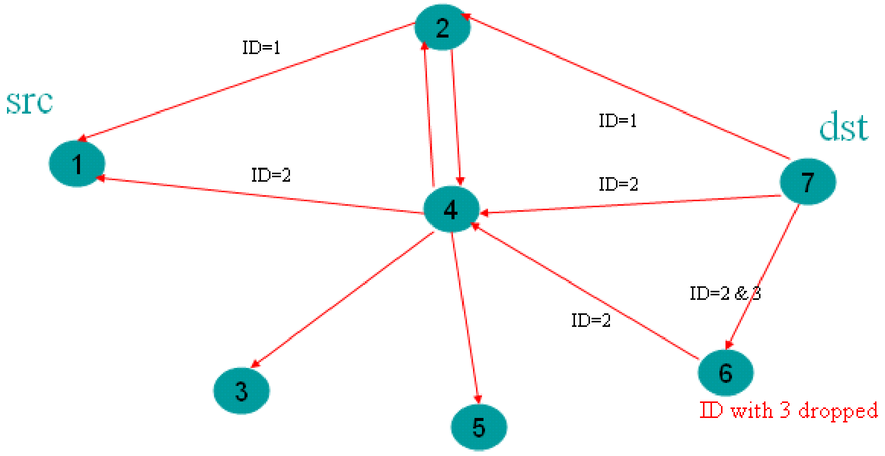


Figure 4.3: Reply of the Request packet

Destination node

The destination node (node 7) here is the node that will receive the request and sends the reply. When the destination node receives the request packet it will either reply with the reply packet or will discard the request. This decision is based on where the reply is sent for the ID received with the same source-destination pair. If the reply has been sent than the packet will be dropped or otherwise the reply is sent by the destination node.

Table 4.6: ID management at node 7 when the first request received and forwarded

Source	Destination	ID	Reply_received
1	7	1	True
1	7	2	True

Table 4.7: Routing table at node 7 when the reply sent

Destination	No. of Hops	Next hop	Expire Time	Sequence Number	ID	Flags
1	2	2	Route time	Route_seq_no	1	UP
1	2	3	Route time	Route_seq_no	2	UP

On receiving the request the routing table entries are made with the flags set to UP i.e. the route is active and the packet can be transmitted by this route. In addition to this the hop count entry is also made into the routing table.

Intermediate node

As shown in Figure 4.3, when the intermediate nodes (nodes 2-6) receive the reply packet it will update two table entries. One is for the ID table and the other is for the routing table. The ID table update is for the recv_reply Boolean variable to be set to true for this source-destination pair. This is to ensure that this node will not accept other requests from the same source-destination pair. The other updates are the routing table updates for the route flag to be set to UP, this will tell that the route is active and the packets can be transmitted by this route.

Table 4.8: ID management at node 3 when the first request received and forwarded

Source	Destination	ID	Reply_received
1	7	1	True
1	7	2	True

Table 4.9: Routing table at node 3 when the reply sent

Destination	**No. of Hops**	**Next hop**	**Expire Time**	**Sequence Number**	**ID**	**Flags**
1	1	1	Route time	Updated	1	UP
7	2	7	Route time	Updated	1	UP

Source node

The source node (node 1) here is the node that has sent the request. When the reply is received by the source node the source node first will update the ID and routing table entries. After that the queue is checked for the packet with the destination with these entries. If the packets are found with the destination address as that in the routing table then they are dequeued and forwarded by the route in the routing table.

Table 4.10: ID management at node 1 when the first request sent

Source	**Destination**	**ID**	**Reply_received**	**Expire**
1	7	1	True	Updated
1	7	2	True	Updated
1	7	3	False	Expire time

Table 4.11: Routing Table at node 1 when the first request made

Destination	No. of Hops	Next hop	Expire Time	Sequence Number	ID	Flags
7	2	2	Updated	Updated	1	UP
7	2	4	Updated	Updated	2	UP
7	Infinity	3	Route time	Current Seq_No	3	Down

Here the destination node (node 7) is receiving the request with ID's 1, 2 and 3 from three different nodes 2, 4 and 6. Here the when the request was made at the source node, three different ID's were selected with different neighboring nodes. At the destination the nodes 2, 4 and 6 were selected with ID's have 1 and 2 and 3, but the nodes 6 will be selected for the ID equal to 2 because the

request is first received for the ID 2 at node 6 and also the reply is first received first by the same node.

4.4 FORWARDING PACKETS

Whenever there is data to be sent and the route is available, the data is forwarded. During the request when forwarding the packets, it will change the source IP address of the packets so that the node receiving the packets can update the routing table for the next hop address. This is only added when the reverse route is to be built up. The other fields update is increase the hop-count by one, decrease the TTL field, etc.

When the data packets are forwarded, it looks for the number of paths available to the destination. If the source node is the current node and the path available to the destination are less than two than it will send a request for the other route. This will find the route other than the already established route. So next time when the route is found out by the route request and the currently available link is lost, the packets can be transmitted to the other route.

4.5 MAINTAINING LOCAL CONNECTIVITY

Each node should continuously keep track of its continued connectivity to its active next hops. The node maintains the local connectivity by the network or the link layer mechanism.

Here the link layer is used for the connectivity of the next hop. This is provided by the IEEE 802.11[5, 8]. The absence of the ACK or failure to get CTS after sending the RTS even after the maximum number of retransmission attempts indicates the loss of link to the next hop.

The node not getting the RTS will now send error message to the network about the failure of the link. All the nodes receiving will now delete the entries from the routing table whose next hop is the source address of the error message. The node after deleting the entry will now forward the message to other nodes and so on until it reaches to the source node of the data packets that is transmitting the packets.

4.6 ACTIONS AFTER REBOOT

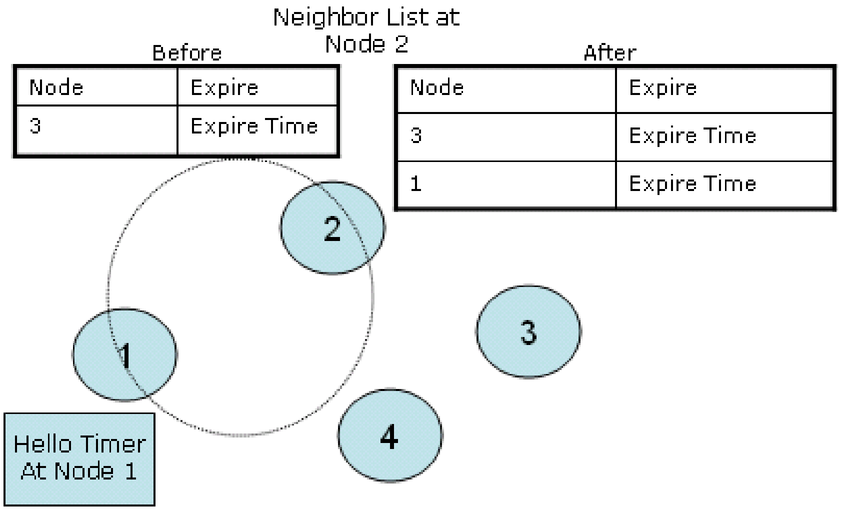
A node participating in the ad hoc network must take certain action after reboot as it might lose all sequence number records for all destinations, including its own sequence number. However, there may be neighboring nodes that are using this node as an active next hop. This might create routing loops. To prevent this, the node does not transmit any control packets until the fixed interval that is DELETE_PERIOD. When the node receives any control packets like hello, request or the reply packet, the will update its routing table entries. When it receives any data packet, it will broadcast an error message to the network, so the source node knows about the failure of the link.

4.7 TIMERS

Timers are used when any regular execution of certain event or function is desired. The timers used in the protocol are the hello timers, Neighbor timers, Route Cache timers, and Broadcast ID timers.

4.7.1 Hello Timer

Hello timer is executed at every node at regular interval of time. This timer is used to continuously send hello packets so that the receiving node can know who the neighbors are. As shown in Figure 4.4 when the hello timer event at node 1 happens, node 1 broadcast hello packet into the network with hop count set to 1. This is to prevent the hello packet to disseminate into the network. Figure 4.4 also shows before and after of the hello packet received at node 2 from the node 1. Before the hello packet is received at node 2 there was only one entry of node 3 and after the hello packet received from node 1-second entry for node 1 is made with the time of expire.



Node	Expire
3	Expire Time

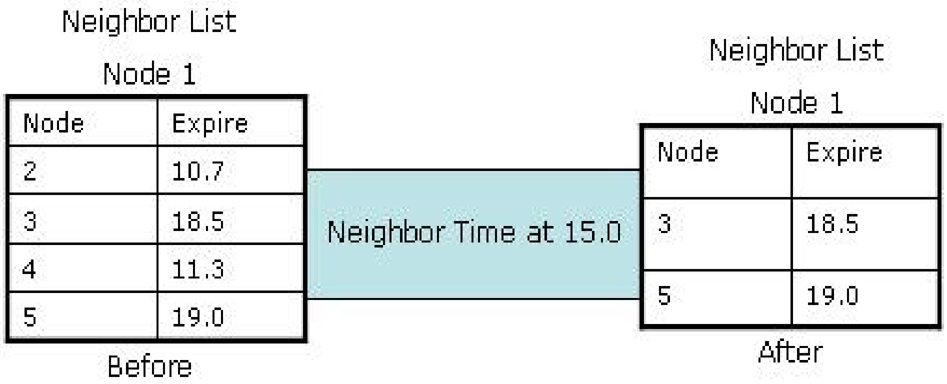
Node	Expire
3	Expire Time
1	Expire Time

Figure 4.4: Hello Timer event

The interval for the hello interval is set to random number. The timer is also used to know whether the neighbor has moved to other location and is out of range. This is done by knowing that the hello message is not received from the particular node within the particular interval of time. If this is so then the node knows that the neighboring node has moved to the other location and so the deletion of that node is required. The deletion is done in both, the ID table as well as well as in the routing table. All the table entries that have the moved node address are updated. In the ID table management the entry is deleted and in the routing table the route flag is set to DOWN.

4.7.2 Neighbor Timer

The neighbor timer is used to delete the entry of the neighbor when the neighbor timer expires. This timer will execute the neighboring purge function and all the entries of the neighbor whose expire time is less than the current time are deleted.



Node	Expire
2	10.7
3	18.5
4	11.3
5	19.0

Node	Expire
3	18.5
5	19.0

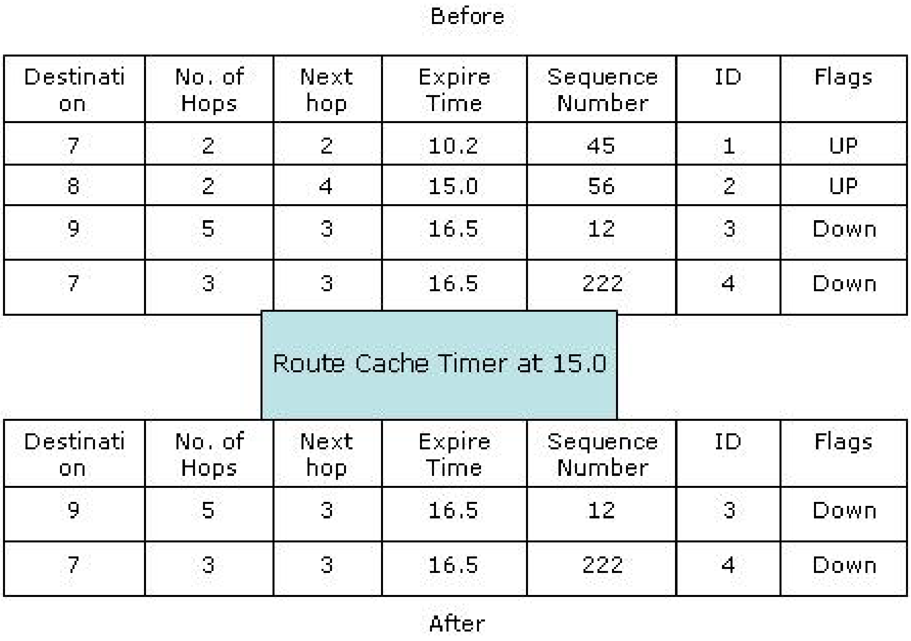
Figure 4.5: Neighbor Timer event execution

The Neighbor interval is fixed for all the nodes i.e. at periodic intervals the neighbors are purged form the entries. The neighbors are again added to the neighbor list when hello packet from that node is broadcast and the current node is in range and one hop away. As shown in Figure 4.5 the neighbor timer event at node 1 is executed at time 15.0. Before the timer event has occurred there are 4 entries and after the event the two entries, whose expire time is less than 15.0 has been deleted.

4.7.3 Route Cache Timer

The route cache timer is executed at regular interval of time. The route cache timer will delete the entries in the routing table whose expire time is less than the current time.

In Figure 4.6 the routing table entries before and after the route cache timer event is occurred is shown. Before the route cache timer there were 4 entries and after the route cache time event 2 entries whose time is less than the route cache execution time are deleted.



Before

Destination	No. of Hops	Next hop	Expire Time	Sequence Number	ID	Flags
7	2	2	10.2	45	1	UP
8	2	4	15.0	56	2	UP
9	5	3	16.5	12	3	Down
7	3	3	16.5	222	4	Down

Route Cache Timer at 15.0

Destination	No. of Hops	Next hop	Expire Time	Sequence Number	ID	Flags
9	5	3	16.5	12	3	Down
7	3	3	16.5	222	4	Down

After

Figure 4.6: Route Cache Timer

The route cache timer is essential because the route are made on the fly when the node in the range of some source to destination. In Ad hoc network the node move at any time, so to maintain the route previously established is of no worth. So at periodic interval of time the route cache timer is to be executed and the route entries, which are expired are deleted.

4.7.4 Broadcast Timer

The broadcast timer is executed at regular interval of time. The timer will execute the ID purge function. This function will delete all the ID's in the ID table whose expire time has reached i.e. expire time is less than the current time.

4.8 PROTOCOL IMPLEMENTATION

In this section the complete protocol implementation is described starting from when the data packets arrived and the initiation of the request for the destination is handled. The flow chart of this protocol is shown in the figure below.

The flow chart as shown in Figure 4.3 shown the implementation of the protocol from the start when the data comes at the nodes and the node sends the request to the destination.

When the data comes to the node, if it is the source node than it first sends the route request to the destination. Before forwarding is done by the source node it adds the routing table entry for the destination, Broadcast_id, expire time and the other information that are required in the routing table. The ID management table is also added with the required information. In case of the routing table the flags which are used to indicate whether the route to the destination is active i.e. it can transmit the data packets to the destination by this route are set to DOWN. DOWN here means that the route is not available currently only the routing table entry is made. In case of ID management table the information for the source and destination are added along with ID and the recv_reply flag is set to false i.e. the reply has not been received for this source-destination request packet.

When the request reaches to the intermediate node the node first adds the reverse route to the source i.e. the routing table entry for the path towards the source. It than adds the ID table entries with the recv_reply Boolean as false, this says that the reply is not received for this entry. The intermediate node then broadcast the request packet into the network with hop count incremented by one. The intermediate node also adds the entry in the routing table with the destination address and hop count to Infinity. Along with all these entries there are also entries made for the expiration of the route. This expiration field is only used for the ad hoc network since the node move now and then so the topology changes every time. So for any entries made in the database of the node is associated with the expire field.

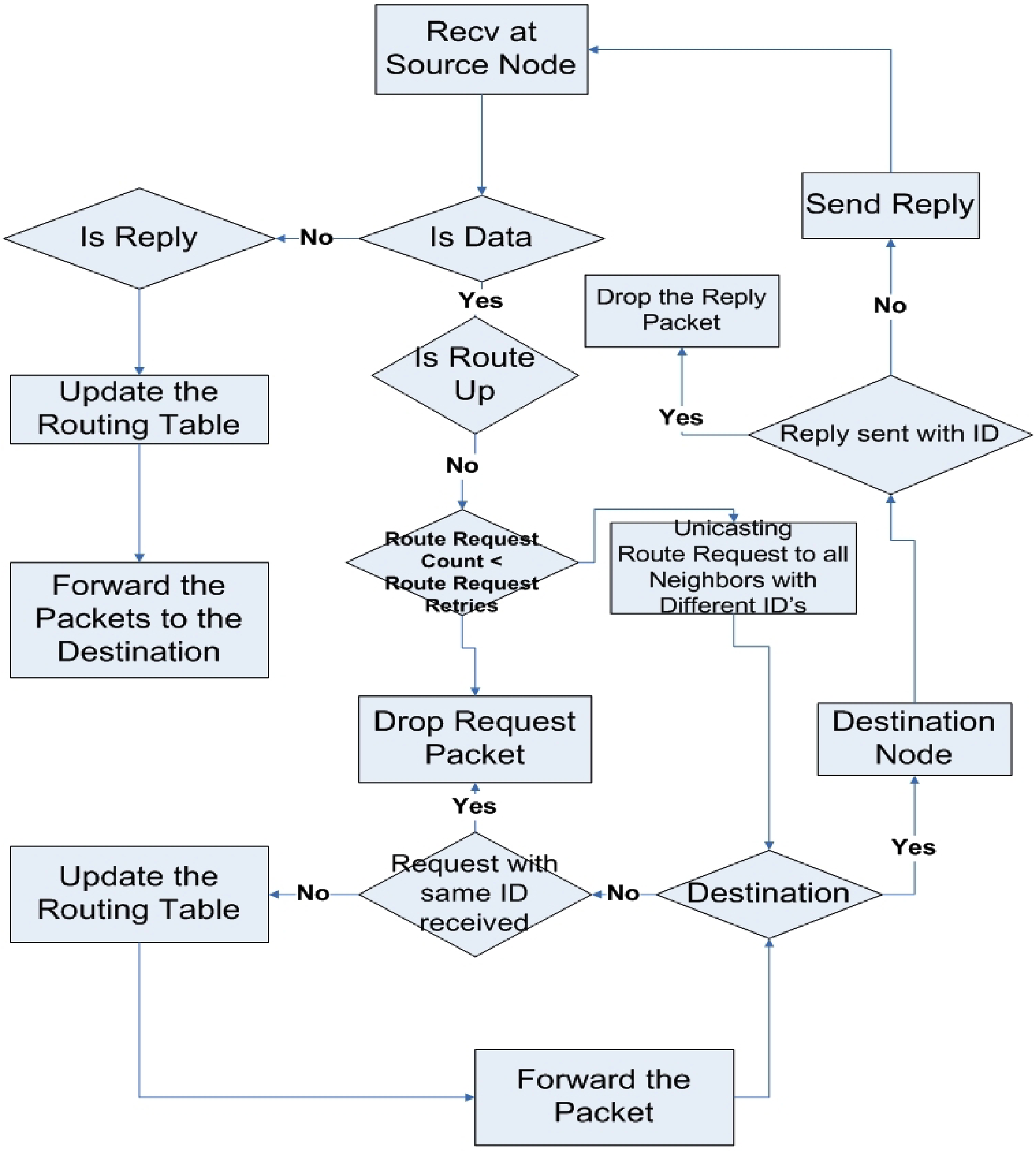


Figure 4.4: Protocol flow chart

When request reaches to the destination the destination node sends single reply for each id that is associated with the request packet i.e. if two requests come with the same id, the destination node replies only for one request. This is to maintain the routing table information consistent. The destination node replies with the reply packet with the fields fill with the information. This information includes the request source, id, hop_count, sequence number, etc. The destination also adds into the ID management table with the reply sent equal to true.

When the reply is received at the intermediate node the intermediate nodes update the routing table as well as the ID table from the reply packet. The ID table entries for the recv_reply are set to true so that the other entries from the same source is not to be handled by this intermediate node. The routing table entries are also updated with the routing flag set to UP if it is DOWN. When the routing flag is UP than the packets can be transmitted by this route because now the route is active.

When the reply reaches the source node of the request packet, the node first updates it ID table and than the routing table. The ID table is similarly updated as the intermediate node i.e. the recv_reply flag which was set false when the request was made is set true. The routing table is updated by setting the route flag to UP, after the update if there are packets in the queue waiting for the transmission whose destination is in the routing table and the route flag is UP are transmitted. The route is now valid until the source, intermediate and the destination node are within the range of each other.

This protocol searches for the multiple routes for the destination. All the routes are with the different ID's. The ID is the only field that can differentiate the routes towards the destination. The ID's are unique for all the nodes. Means there is no central administration that maintains the ID's. When the reply is received by the source node the packets are transmitted to all the routes or can be transmitted to the single route whose hop count is less than all the routes that are available. The packets can also be transmitted by considering the timestamp field that is used in the reply field. By this field we can know the time required for the packet to reach the destination and accordingly we can send the packets of any multimedia application.

5 SIMULATION ENVIRONMENT

For the experiment purpose of this thesis NS2 (Network Simulator 2) has been used. NS2 is a discrete event simulator targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

5.1 NETWORK SIMULATOR

NS2 is an object oriented simulator, written in C++, with the OTcl interpreter as a front-end. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the OTcl interpreter (also called the interpreted hierarchy) [2]. As shown in Figure 5.1, the two hierarchies are closely related to each other; from the user's perspective, there is one-to-one correspondence between the class in the compiled hierarchy and the class in the interpreted hierarchy. The root of this hierarchy is called the OTclObject. User creates a new simulator object through the interpreter; these objects are instantiated within the interpreter and are closely mirrored by a corresponding object in the compiled hierarchy.

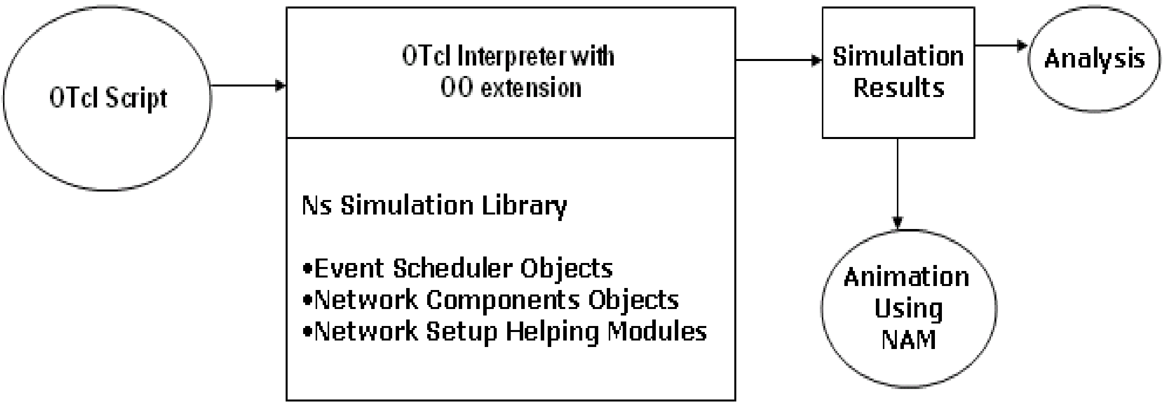


Figure 5.1: Simplified user view of Network simulator

The interpreted 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 the OTcl scripts; these hierarchies are not mirrored in the manner of TclObject.

NS use two languages because simulator has two different kinds of things it needs to do. On one hand, detailed simulations of protocols require 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. NS 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. NS provides glue to make objects and variables appear on both languages.

5.2 ADDING A NEW ROUTING PROTOCOL IN NS2

In this section implementation of new routing protocol is explained step-by-step. The first step for adding a new MANET routing protocol in NS2 is to create a directory of the protocol name in the ns2 base directory [3].

The basic files used for the protocol are then added. In the Multipath routing protocol the files used are protocolname.cc and its header file, routing table with its header file and the queue with the header file and the packet format file which we use to define various packet formats.

From all this files the actual implementation is done in protocolname.cc file which implements all the timers, routing agents and the Tcl hooks. The Tcl hook is the interface by which the cc files interacts with the Tcl files. The Tcl script is used for the simulation purposes. As stated in Section 5.1 all the protocolname.cc files are the compiled hierarchy and the Tcl file is the interpreted hierarchy. Thus there is one-to-one correspondence between the compiled hierarchy and the interpreted hierarchy.

In short the different files used and their actual contents are described below:
Protocol.h This is the header where all the necessary timers and the routing agents are defined which performs protocol functionality.
Protocolname.cc The actual implementation of timer and Tcl hooks are done
Protocol_packet.h All the packet formats are defined here
Protocol_routing.h Header file for the routing table
Protocol_routing.cc In this the actual implementation of the routing table functionality is defined.

For any protocol to be implemented have some sort of functions that should be periodically executed. In case of Multipath routing protocol the hello packets are sent by all the nodes periodically, the other is the routing purge and the broadcast id purge that are periodically executed. This functionality is executed by the timers only. These timers are defined in the header file of the main protocol.

Packet.h

In this the various packets that are used in the protocol are defined. For example in the Multipath routing protocol, the Request packets, Reply packets, etc. are used were defined in the files. Types used in the packet format i.e. 16-bit integer, 8-bit integer, etc. are defined in the config.h file. The packet format is defined within the structure. These packets are only the control packets. The IP packet and the common packet are defined in the packet.h file in the ns2 directory. By this one is able to use the created control packets in the simulation and is received by the nodes.

As the packet received by the node is the ip packet we need to define the offset field where this control packet starts, this is also defined in the packet class.

Protocolname.cc

To use our packet with the Tcl interface we need to bind it with the Tcl interpreter. This binding is done in protocolname.cc file.

Protocolname.h

In this the class for the protocol used is defined which contains the attributes and the functionality needed by the protocol in doing its job. For all this functions to execute the needed header files are the timer-handler.h, trace.h, random.h, classifier-port.h, etc.

The command function is built in the ns2 from where the arguments passed are defined along with this is the Tcl_OK and the Tcl_Error at the end of each of the case of argument count.

The basic function of the protocol are the recv (), where it is defined in the code about what the node has to do when it receives any packet whether it is control or the data packet. The other functions are the forwarding (), Reply (), etc.

All the tracing fields are defined in the cmu-trace.cc file where the tracing of the parameters are defined. We can also define the trace parameters from the Tcl script what we want to trace.

In the routing table file all the function of the routing table are defined. The common function of the routing table are addition (), deletion () and update(). The routing table field is defined in the header file of the routing table.

5.3 CHANGES NEEDED AFTER ADDING NEW PROTOCOL IN NS2

When we have added all the files and their function into the respective files the other things are also there to be changed into the ns2 directory for the successful compilation and to run the protocol in the simulation.

The changes required are given below:

Common/packet.h

In the common folder into the ns2 base directory the packet.h file has to be modified. In this file the protocol packet type is to be added e.g. if the new protocol is protocolname than the line PT_ptotocolname has to be added into the enmu_packet_t field.

The other changes are into the p_info () function where the protocolname has to be added.

Trace/cmu-trace.h

To support for the tracing, changes is also required into the cmu-trace.h where the function format_protocolname has to be added. The functionality of this is to be defined into the cmu-trace.cc file.

The new wireless trace format that is introduced in NS documents 2006[2] is explained below. This is only for the wireless network and is the only means by which we get the complete network information.

Explanation of new trace format

The new trace format as seen above can be can be divided into the following fields.

Event type In the traces above, the first field (as in the older trace format) describes the type of event taking place at the node and can be one of the four types:

Table 5.1: Trace of event Type

s	send
r	receive
d	drop
f	forward

General tag The second field starting with "-t" may stand for time or global setting

Table 5.2: Trace for General tab

-t	Time
-t *	(global setting)

Node property tags This field denotes the node properties like node-id, the level at which tracing is being done like agent, router or MAC. The tags start with a leading "-N" and are listed as below:

Table 5.3: Trace for the node property tag

-Ni:	node id
-Nx	node’s x-coordinate
-Ny	node’s y-coordinate
-Nz	node’s z-coordinate
-Ne	node energy level
-Nl	traces level, such as AGT, RTR and MAC
-Nw	Reason for the event.

The different reasons for dropping a packet are given below:

Table 5.4: Trace for Packet Dropping Event

"END"	DROP_END_OF_SIMULATION
"COL"	DROP_MAC_COLLISION
"DUP"	DROP_MAC_DUPLICATE
"ERR"	DROP_MAC_PACKET_ERROR
"RET"	DROP_MAC_RETRY_COUNT_EXCEEDED
"STA"	DROP_MAC_INVALID_STATE
"BSY"	DROP_MAC_BUSY
"NRTE"	DROP_RTR_NO_ROUTE i.e. no route is available.
"LOOP"	DROP_RTR_ROUTE_LOOP i.e. there is a routing loop
"TTL"	DROP_RTR_TTL i.e. TTL has reached zero.
"TOUT"	DROP_RTR_QTIMEOUT i.e. packet has expired.
"CBK"	DROP_RTR_MAC_CALLBACK
"IFQ"	DROP_IFQ_QFULL i.e. no buffer space in IFQ.
"ARP"	DROP_IFQ_ARP_FULL i.e. dropped by ARP
"OUT"	DROP_OUTSIDE_SUBNET i.e. dropped by base stations on receiving routing updates from nodes outside its domain.

Packet information at IP level The tags for this field start with leading "-I" and are listed along with their explanations as following:

Table 5.5: Trace at the IP level

-Is	source address.source port number
-Id	dest address.dest port number
-It	packet type
-Il	packet size
-If	flow id
-Ii	unique id
-Iv	ttl value

Next hop info This field provides next hop info and the tag starts with a leading "-H".

Table 5.6: Trace for the Next hop Information

-Hs	id for this node
-Hd	id for next hop towards the destination

.

Packet info at MAC level This field gives MAC layer information and starts with a leading "-M" as shown below:

Table 5.7 Trace for Packet Information at MAC level

-Ma	duration
-Md	destination ethernet address
-Ms	source ethernet address
-Mt	ethernet type

Packet info at "Application level" The packet information at application level consists of the type of application like ARP, TCP, the type of adhoc routing protocol like DSDV, DSR, AODV etc being traced. This field consists of a leading "-P" and list of tags for different application is listed as below:

-P arp Address Resolution Protocol. Details for ARP is given by the following tags

Table 5.8: Trace for the packet information at Application Level

-Po	ARP Request/Reply
-Pm:	src mac address
-Ps:	src address
-Pa:	dst mac address
-Pd:	dst address

-P dsr This denotes the adhoc routing protocol called Dynamic source routing. Information on DSR is represented by the following tags:

Table 5.9: Trace for the packet of DSR Routing protocol

-Pn	How many nodes traversed
-Pq	routing request flag
-Pi	route request sequence number
-Pp	routing reply flag
-Pl	reply length
-Pe	src of src routing->dust of the source routing
-Pw	error report flag ?
-Pm	number of errors
-Pc	report to whom
-Pb	link error from linka->linkb

-P cbr Constant bit rate. Information about the CBR application is represented by the following tags:

Table 5.10: Trace for the CBR data

-Pi	sequence number
-Pf	how many times this packet was forwarded
-Po	optimal number of forwards

-P tcp Information about TCP flow is given by the following subtags:

Table 5.11: Trace for the TCP flow

-Ps	sequence number
-Pa	ack number
-Pf	how many times this packet was forwarded
-Po	optimal number of forwards

Tcl Library

Now some changes in Tcl files is required. Actually we are going to add our packet type, give default values for binded attributes and provide the needed infrastructure to create wireless nodes running our protoname routing protocol.

Ns-packet.tcl

In this file needed changes are that we have to add our protocol name into the foreach_prot{} function.

Ns-default.tcl

In this files are the default values for the variable that are sent by the user from the Tcl script. The default values are required because in case if the user does not enter any values to the binding variables than the default values are to be taken. This default values are taken from the ns-default.tcl files

Ns-lib.tcl

The procedure node calls to the create-wireless-node procedure. This last one, among other tasks, is intended to set the routing agent for a node. It is need to hack this procedure to create an instance of our protoname protocol. The other script for the creation of protoconname agent with the node address is also to be defined here.

Queue/priqueue.cc

Priority queues treats routing packets as high priority packets, inserting them at the beginning of the queue. The PriQueue class mentions that protoname packets are routing packets and therefore treated as high priority. So the priorities for the packets are defined here.

Makefile

The last step after adding all the required function into the files, it should be compiled. The compilation of file is done by the "make" command. Before making the "make", the object files is to be added into the object list of the "Makefile". After adding and running the make command, if no errors comes than the protocol is successful and can be used for the simulation purpose.

6 RESULTS OF SIMULATION

In this chapter the implementation part for the new Multipath routing protocol is discussed. The next section will introduce about the testbed for the implementation and than the next section shows the results and than at last the conclusion part of the implementation.

6.1 TESTBED

The simulation is done in NS2. The complete guidance about the simulation environment can be seen in [1] [2]. The simulation with 31 nodes was done with the new Multipath routing protocol. Below is given the simulation environment

Scenario

The scenario for the network is given below:

Area = 670 x 670 meters
Node movement = Random
Packet size = 512 bytes for constant bit rate
Interface Queue length = 50 packets
Interface Queue = Tail drop for AODV, DSDV and Multipath and
Priqueue for DSR
Simulation length = 140 seconds
Antenna Type = Omni directional
Transmission range = 250 meters
Transmitting power and Receiving power = 281.8 mille Watt
Propagation Type = Two-ray ground Reflection

Table 6.1: Comparison of different routing protocols with max mobility 10m/s

Parameters	AODV	DSR	DSDV	Multipath Routing
No. of Nodes	30	30	30	30
Mobility Max	10	10	10	10
Packets Type	CBR	CBR	CBR	CBR
Efficiency (%)	100	100	100	87.68
Avg Delay(ms)	33.4912	9.8416	8.26218	25.7392
Min Delay(ms)	5.469	5.45034	5.44978	5.469
Max Delay(ms)	61.1047	399.78	20.3764	691.963

Table 6.2: Comparison of different routing protocols with max mobility 20m/s

Parameters	AODV	DSR	DSDV	Multipath Routing
No. of Nodes	30	30	30	30
Mobility Max	20	20	20	20
Packets Type	CBR	CBR	CBR	CBR
Efficiency (%)	99.84	100	75.39	92.96
Avg Delay(ms)	15.5204	19.1148	11.8156	14.0406
Min Delay(ms)	11.2043	11.8337	11.1637	11.2035
Max Delay(ms)	61.1047	399.78	20.3764	691.963

Delay and Throughput

From the Table 6.1 and 6.2 it shows that the efficiency of the protocol is decreased when the mobility increases. In case of Multipath routing protocol the case is not true. In Multipath routing protocol when the mobility is 10m/s the efficiency i.e. the ratio of packets sent to the packets received is not 100% because in this the packets are dropped when there is link breakage.

Figure 6.1, 6.2, 6.3 and 6.4 shows delay graph for different routing protocol. From the graph it is shown that the minimum delay is for the DSDV routing protocol than is for Multipath followed by AODV and DSR. DSDV protocol is best suited for the application where packet loss is tolerable for e.g. conversational voice, voice messaging etc. but the delay is not tolerable. Like wise AODV and DSR are more applicable where delay is preferable but packet loss are tighter, for e.g. data transfer.

The packet loss in DSDV routing protocol is due to the control information that is disseminated into the network for the up-to-date routing table. This control information has more priority than that of the data packets as they contain the network related information. So, due to this the data packets are dropped at the intermediate node which acts as the router. As the routing table contains the route to the entire available destination and the path to that destination is the lowest the delay is less in this routing protocol.

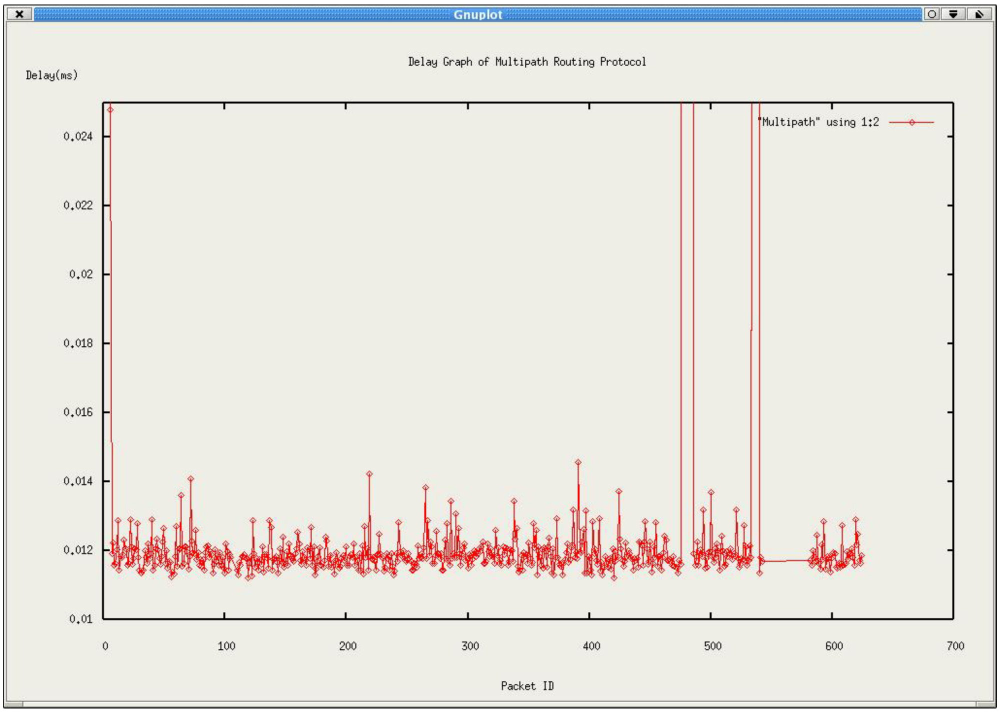


Figure 6.1: Delay of Multipath routing protocol with maximum mobility 20m/s

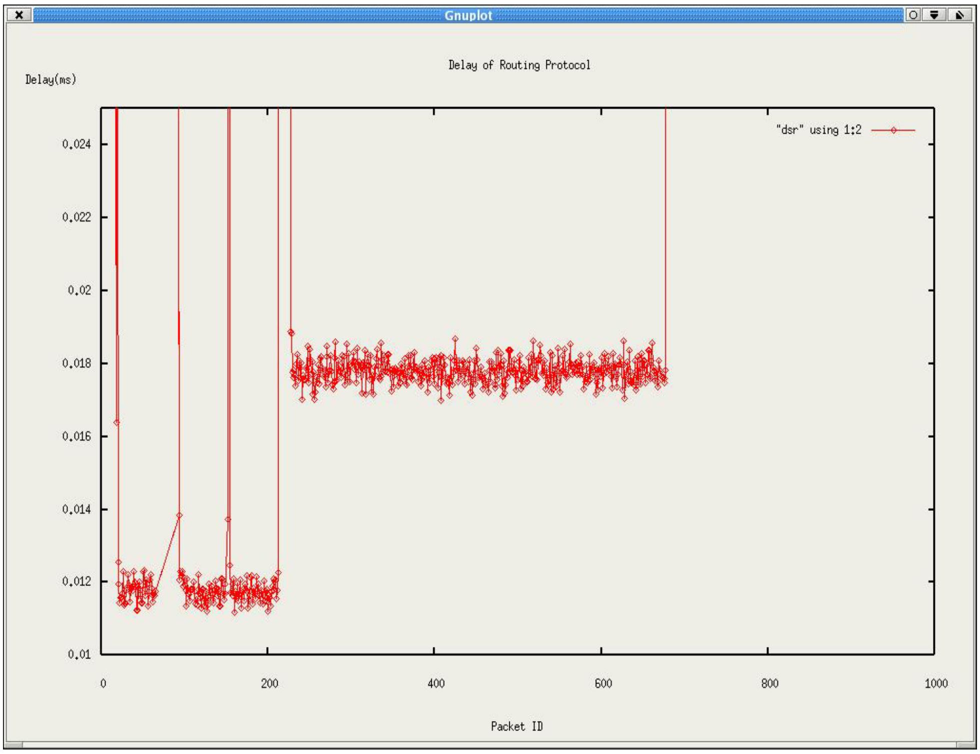


Figure 6.2: Delay of DSR Routing Protocol with maximum mobility20m/s

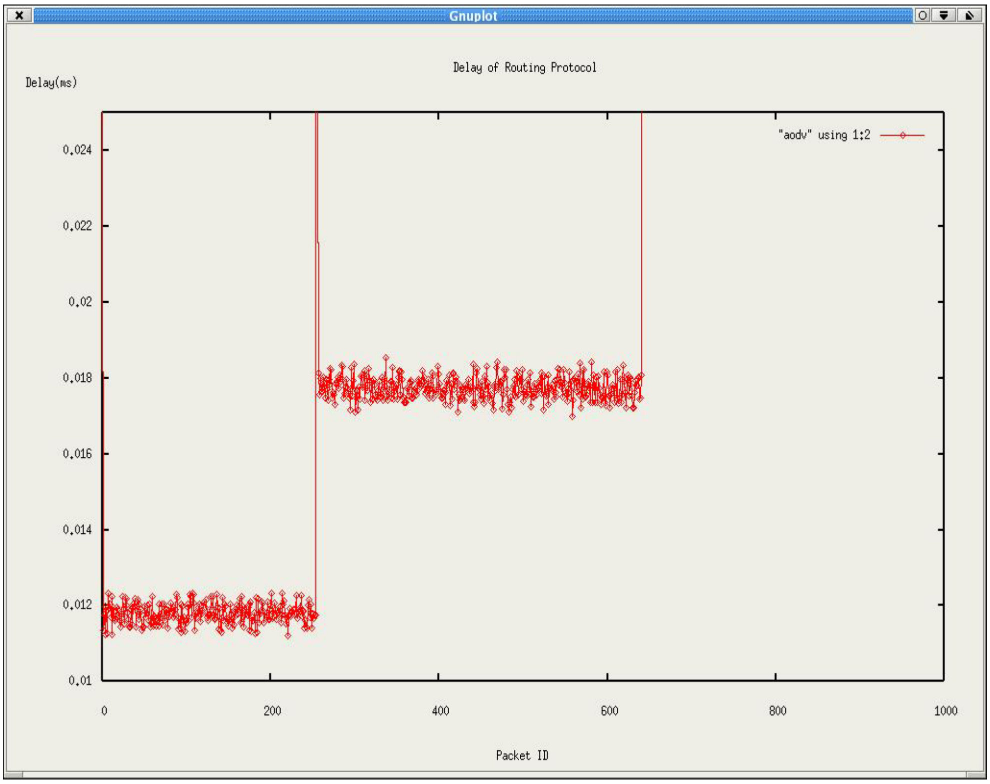


Figure 6.3: Delay of AODV Routing Protocol with maximum mobility 20m/s

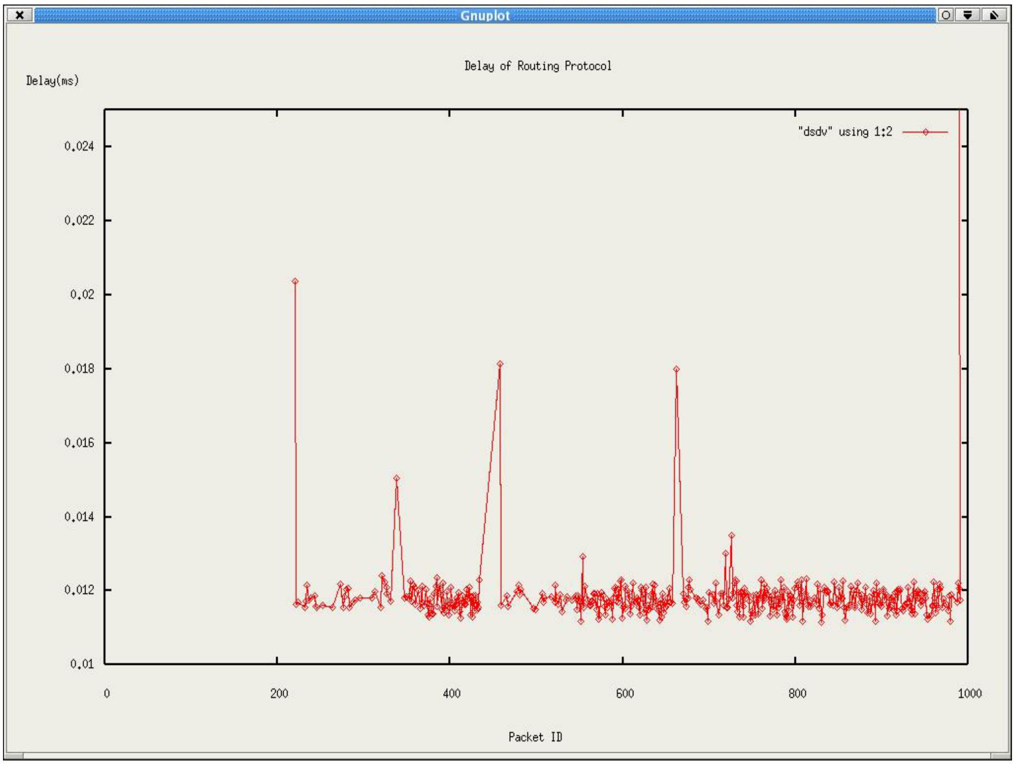


Figure 6.4: Delay of DSDV Routing Protocol with maximum mobility 20m/s

As from the results of this routing protocol the required routing protocol was where the delay is less than AODV and DSR routing protocol and the packet loss is less than that of DSDV routing protocol. The new Multipath routing protocol suites best at that place where the packet loss is less than DSDV and delay is less than DSR and AODV routing protocol.

In Multipath routing protocol multiple paths are discovered for the source. All the paths are having different number of hop count to the destination. So when the source has the data to transmit it selects the best path from the available paths. So as seen from the graph the delay is somewhat same. As the routing protocol searches for the multiple paths the control information are more than that of AODV and DSR routing protocol and so suffers from the packet loss. In case of AOD and DSR routing protocol the control information is transmitted periodically i.e. at fixed interval of time, even if the network topology changes. Also the routing table is updated for the source is updated only when there is link breakage. So, in this type of routing table the control information are less and so the packet loss are less. As the routing table is update only when there is link breakage the delay is more.

7 CONCLUSION

It can be clearly seen from the table 6.1 that the delay is less for DSDV than other protocols. This delay variation is so because the routing tables in DSDV protocol is updated as the topology changes. During the topology change the control information about the node is to be broadcasted. This broadcasting of the routing table is dependent on the timing interval of how frequent the routing table is updated. This updated information is to be transmitted to the other nodes in the network. So the packet loss is more due to the control packets processing at the intermediate nodes. As the routing tables are the updated regularly the source gets the best path to the destination.

In case of DSR and AODV routing protocol the route for the packets from the source once established is updated only when there is path break. So even if the destination node is far away during the first packet transmission and comes to a single hop distance during the other packet transmission the path do not change as long as the path is active. This is the main reason for more delay in AODV and DSR routing protocol. The other parameter is the packet loss in the AODV and DSR routing protocol, in this the control packets are transmitted on timely basis i.e. at fixed interval of time the control information is disseminated into the network. So due to less number of control packets the node which acts as the router handles the data packets.

The new Multipath routing protocol searches for multiple paths to the destination. The maximum number of paths is limited to four paths because of control to dissemination of control packets in the network. This dissemination of control packets is increased with respect to the network size. So the delay difference in this protocol is due to restriction made on number of paths to destination. Also the control information transmitted is on timely basis as in case of AODV and DSDV routing protocol. In case of DSDV routing protocol, single path is available; this is the best one to reach the destination.

Future Work

The packet loss ratio is high in Multipath routing protocol than that of AODV and DSR routing protocol. The drawback of this protocol is that whenever there is a link break, the packet is dropped. This is mostly occurring at the intermediate nodes. So, the next work direction is towards reducing these packet losses.

REFERENCE

[1] Marc Greis. Tutorial for the Network Simulator NS.
http://nsnam.isi.edu/nsnam/index.php/

[2] Kevin Fall and Kannan Varadhan, ”The NS Manual (formerly NS Notes and Documentation)” *The VINT Project Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.* September 16, 2006

[3] Francisco J. Ros and Pedro M. Ruiz, *“Implementing a New Manet Unicast Routing Protocol in NS2”*, Dept. of Information and Communications Engineering, University of Murcia, December 2004.

[4] C.Siva Ram, Murthy and B.S.Manoj, *“Ad Hoc Wireless Networks: Architecture and Protocols,” India: Pearson Education Asia pte. ltd., 2005.*

[5] T. Bheemarjuna Reddy, I. Karthigeyan, B.S. Manoj, and C. Siva Ram Murthy, *“Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions”,* Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600036, India.

[6] Meri Hyytinen Helsinki, *“Resource Reservation Protocol (RSVP) Research Seminar on IP QoS,”* Department of Computer Science, University of Helsinki, September 2000.

[7] A.Talukder and Yavagal, *“Mobile Computing”,* Tata McGraw Hill, 2005.

[8] Shiann-Tsong Sheu and Tzu-Fang Sheu, “A Bandwidth Allocation/Sharing/Extension Protocol for Multimedia Over IEEE 802.11 Ad Hoc Wireless LANs”, IEEE Journal, Volume 19, Issue 10, Oct 2001 pp. 2065 – 2080.

[9] Yihan Liy, Shiwen Maoz, and Shivendra S. Panwar, “The Case for Multipath Multimedia Transport over Wireless Ad Hoc Networks”, Department of

Electrical and Computer Engineering Polytechnic University, Brookly, Pages: 486 - 495, 2004 IEEE Computer society Washington, DC, USA

[10] International Telecommunication Union (ITU_T) Study Group 12: *"Performance and quality of service – Lead Study Group on Quality of Service and performance",* http://www.itu.int/ITU-T/studygroups/com12/index.asp.

[11] Tsu-Wei Chen, "Efficient Routing and Quality of Service Support for Ad Hoc Wireless Networks", University of California, Los Angeles, 1998.

[12] Ian D. Chakeres and Elizabeth M. Belding-Royer, "AODV Routing Protocol Implementation" Design Dept. of Electrical & Computer Engineering University of California, Santa Barbara.

[13] C.E. Perkins, T.J.Waston, and Pravin Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing [DSDV] for mobile networks", Pages: 234 - 244, 1994. ACM Press New York, NY, USA

[14] David B. Johnson, David A. Maltz, and Josh Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," Computer Science Department Carnegie Mellon University Pittsburgh, PA 15213-3891 http://www.monarch.cs.cmu.edu/

APPENDIX A INSTALLING NS-2

NS-2 is designed to run from on most UNIX based operating systems [1]. It is possible to run NS-2 on Windows machines using Cygwin. If you don't have a UNIX install, you can also use a virtual linux machine and run that under Windows. VMWare has a free VMWare Player that allows you to download linux systems like Ubuntu and run them on your computer. You will need to make sure you have standard development packages like 'make' and 'gcc'.

Installing NS2 on UNIX Based System

First, download a copy of ns-allinone-2.29.3.tar.gz. Then from the command prompt there, execute the following:

```
tar -xzf ns-allinone-2.29.3.tar.gz
cd ns-allinone-2.29
./install
```

After a long wait and a whole lot of text, you should see the installation finish up with text like the following:

```
Nam has been installed successfully.
Ns-allinone package has been installed successfully.
Here are the installation places:

    • tcl8.4.11:    /home/pcraven/ns-allinone-2.29/{bin,include,lib}
    • tk8.4.11:     /home/pcraven/ns-allinone-2.29/{bin,include,lib}
    • otcl:         /home/pcraven/ns-allinone-2.29/otcl-1.11
    • tclcl:        /home/pcraven/ns-allinone-2.29/tclcl-1.17
    • ns:           /home/pcraven/ns-allinone-2.29/ns-2.29/ns
    • nam:          /home/pcraven/ns-allinone-2.29/nam-1.11/nam
    • xgraph:       /home/pcraven/ns-allinone-2.29/xgraph-12.1
    • gt-itm:       /home/pcraven/ns-allinone-2.29/itm, edriver, sgb2alt,
                    sgb2ns, sgb2comns, sgb2hierns
```

put

/home/myusername/ns-allinone-2.29/bin:/home/myusername/ns-allinone-2.29/tcl8.4.11/unix:/home/myusername/ns-allinone-2.29/tk8.4.11/unix

into your **PATH environment**; so that you'll be able to run itm/tclsh/wish/xgraph.

(1) You MUST put

/home/myusername/ns-allinone-2.29/otcl-1.11,
/home/myusername/ns-allinone-2.29/lib,

into your **LD_LIBRARY_PATH** environment variable.

If it complains about X libraries, add path to your X libraries into **LD_LIBRARY_PATH.**

If you are using csh, you can set it like:

setenv LD_LIBRARY_PATH <paths>

If you are using sh, you can set it like:

export LD_LIBRARY_PATH=<paths>

(2) You MUST put

/home/myusername/ns-allinone-2.29/tcl8.4.11/library

into your **TCL_LIBRARY** environmental variable. Otherwise ns/nam will complain during startup.

(3) [OPTIONAL] To save disk space, you can now delete directories tcl8.4.11 and tk8.4.11. They are now installed under /home/myusername/ns-allinone-2.29/{bin,include,lib}

After these steps, you can now run the ns validation suite with

cd ns-2.29;
./validate

At this point, you should follow the advice here and update your environment variables. You should also add ns-allinone-2.29/bin to you path. This has links to all the executables created by NS-2. Since the Tcl scripts may call these executables (like nam or xgraph), it is a good idea to have them in the path.