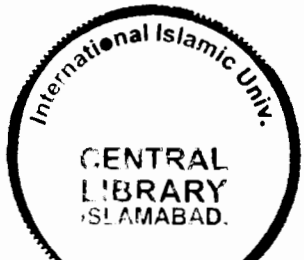
EFFICIENT AODV ROUTING BASED ON TRAFFIC LOAD AND MOBILITY OF NODE IN MANET

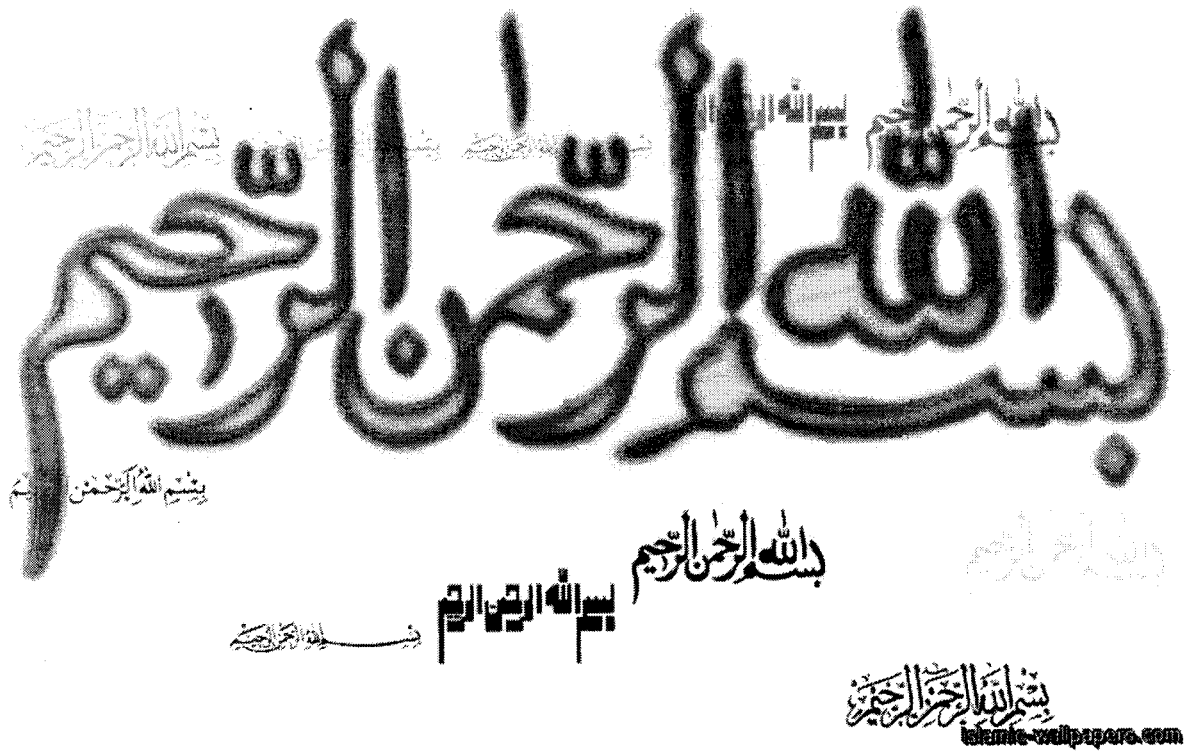
Iftikhar Ahmad
446-FBAS/MSCS/F08

Supervised By

Muhammad Mata ur Rehman

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University Islamabad
(2010)





Efficient AODV Routing based on Traffic Load and Mobility of node in MANET

I am dedicating my thesis to my Grand Mother, Father and to the people of Kashmir

A dissertation Submitted To

Department of Computer Science,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

As a Partial Fulfillment of the Requirement for the Award

of the Degree of MSCS.

Declaration

We hereby declare that this Thesis ***"Efficient AODV Routing based on Traffic Load and Mobility of node in MANET"*** neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this research with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor *Mata ur Rehman.* If any part of the system is proved to be copied out from any source or found to be reproduction of any project from any of the training institute or educational institutions, we shall stand by the consequences.

Iftikhar Ahmad

446-FBAS/MSCS/F08

Acknowledgement

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project. Special thanks to my Holy Prophet Muhammad (SAW) who enable us to recognize Allah the Almighty and Who is the real role model for me in every aspect of life.

I am thankful to my honorable Sir Mata ur Rehman for their support and encouragement during the process of my research work. His motivation led me to this success. Without his sincere and cooperative nature and guidance it would have been difficult for me to complete my thesis.

It is surely due to my parent's moral and financial support during my entire career that enables me to complete my work. I am also thankful to my loving bothers, friends who mean most to me.

Iftikhar Ahmad

446-FBAS/MSCS/F08

Project In Brief

Project Title:	Efficient AODV Routing based on traffic load and Mobility of node in MANET
Undertaken By:	Iftikhar Ahmad 446-FBAS/MSCS/F08
Supervised By:	Muhammad Mata ur Rehman
Start Date:	September 2009
Completion Date:	November 2010
Tools & Technologies	Network Simulator 2, AWK Scripting, Xgraph
Documentation Tools	MS Office ,MS paint
Operating System:	Fedora 8, windows 7 ultimate N
System Used:	Pantium IV

Abstract

Mobile ad hoc network is wireless network of mobile nodes, with no centralized management and control. We proposed a protocol with an enhanced route discovery mechanism that avoids congestion in the route. Our proposed protocol selects route on the basis of traffic load on the node and resets path with mobility. Instead of transmitting entire data through one route, new efficient paths are discovered from time to time during transmission. Our technique focuses to maintain efficient and less congested path for longer transmissions.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Introduction

More and more advancements in wireless communication technologies and availability of less expensive, small, portable computing devices led to mobile computing and its applications. A "mobile ad hoc network" (MANET) consist of mobile nodes connected by wireless links. The union of which forms an arbitrary graph. The nodes are free to move randomly and organize themselves arbitrarily; thus, the network's topology may change rapidly and unpredictably.

MANET may operate in a standalone fashion, or may be connected to the other larger network like Internet. Applications of MANETs include military use in battle fields, where a centralized command center is not only infeasible but also undesirable; and disaster management scenarios, where on the run, communication between various rescue teams is required in the absence of any existing communication infrastructure. Example of mobile adhoc network is shown in Figure-1.

A key challenge in MANETs is to devise efficient methods to ensure reliable route availability by utilizing minimum energy of each node while incurring minimal packet drop. MANET routing protocols are of two kinds: proactive (table-driven) and reactive (on-demand). Proactive class of protocols always have routes to any destination in the network, while reactive protocols class need to discover routes as needed. Proactive protocols suffer from excessive control overhead associated with maintaining routes to destinations even when not required, while reactive protocols experience higher end-to-end packet delays when compared to proactive protocols.

Examples of proactive protocols are Destination Sequenced Distance Vector (DSDV) and Optimized Link State Routing (OLSR) while some examples of reactive protocols include Dynamic Source Routing (DSR), Ad hoc On-demand Distance Vector

Ad hoc on-demand distant vector routing protocol is descendent of DSDV and it is especially design for mobile ad hoc networks [2]. AODV discover routs through flooding of a message throughout the network but this flooding is controlled through TTL. Overall there are four types of messages in AODV which are RREQ, RREP, RERR and Hello message. PREQ is the message for rout discovery, RREP message is routed back whenever the rout is discovered by the

destination or any intermediate node in the path. In case of link failure the RERR message is used by the node which faces link failure to the next node. RERR message send back to the source node then source node starts rout discovery again. AODV uses sequence number to control looping. In rout discovery process AODV uses different flags and fields in RREQ message. These fields include source and destination node addresses, source and destination sequence numbers, a broad cost ID and other validation flags. RREQ ID with source address uniquely identifies a rout request and is used to prevent repetition of same message.

Basic AODV protocol does not take parameters like congestion on node, power consumption, and link state of the route into account in the route discovery process. Different improvements in AODV have been made which have shown improvement in performance in typical network scenario.

In basic AODV once a rout has been identified the entire data is transmitted on that route until unless failure of link or transmission of data completed. This technique is useful for shorter period transmission when transmission needs a link for shorter period of time.

For longer period transmissions, typical mobility feature in MANET can be useful. Instead of transmitting on pre selected path, new efficient path can be found and utilized to improve overall through put.

Route discovery algorithm

Algorithm used in route discovery mechanism of AODV routing protocol for the route discovery

Throughput

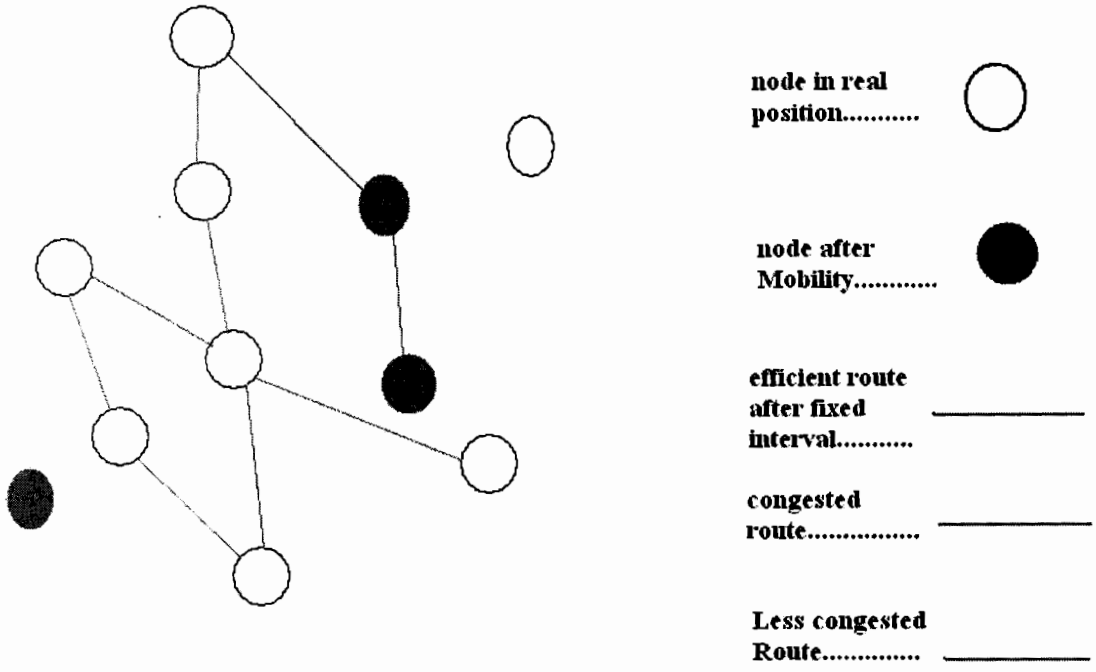
Throughput is the total number of packets transmitted across the network in the given time.

Packet delivery ratio

It is the ratio of the number of packets received successfully to the total number of packets transmitted.

Packet loss

It is the packet loss rate of the transmission.



A Typical MANET scenario with Molility

Figure 1: A MANET Scenario

1.2 Problem statement

In the mobile Ad hoc networks route discovery algorithm of existing AODV [10] protocol is not suitable for the longer transmissions that require less congested and efficient path during entire transmission time.

Existing route discovery algorithms may selects congested and longer routes, or the route may get congested during transmission and resulting into more packet drops and lesser throughput.

We proposed modifications in the route discovery algorithm and route selection criteria of existing AODV routing protocol. The performance of the protocol can be increased by modifying the existing route discovery algorithm of AODV routing protocol.

1.3 Motivation

Ideally, AODV routing protocol that produces packet drop due to congestion and MANET life is critical. In the near future, we can expect a significant amount of multimedia traffic in ad hoc networks. All existing ad hoc protocols and all existing versions of AODV are not capable of handling congestion in the selected route well.

The work described here is an attempt to design a routing protocol that offers lesser packet drop and more end-to-end throughput and increase in delivery ratio. This can only be achieved by taking in to account traffic load information together with mobility of mobile nodes which is generally ignored in conventional AODV protocol. Existing versions of AODV protocols do make use of link state information, node status, buffer status and etc in different ways and hence do not offer a great performance advantage for transmissions that require a link for longer period of time; each protocol performs better in certain scenarios.

1.4 Research Overview

The scope of this research is to enhance an existing reactive ad hoc routing protocol AODV in the form that produce greater performance over the existing versions of AODV in the given network. Further, the protocol is modified into an load preventing mode suitable for congestion avoidance by preventing a busier node not to take part in the route selection during the route selection procedure and taking other nodes into account that can take part in the route by expiring route after a time duration in .We call the resulting protocol load and mobility based AODVLM. The following presents a summary of the total work done:

1. Developed an enhanced algorithm for the traffic load calculation and to predict the congested route during the rout selection in AODV.

2. Enhanced algorithm for the selection of route from the given routes.

2. Used of a mechanism to take advantage of the node mobility

3. Combined implementation of enhanced algorithm and mobility mechanism in AODV
4. Implemented the above mentioned approaches on given network scenario especially designed to cheek the performance of these approaches.

5. Implemented the above enhancements to AODV in the ns-2 simulator and ran simulations for thorough performance evaluation studies that compare AODV and load and mobility based AODVLM .

6. Simulation results and performance analysis of both protocols are provided.

7. For TCP traffic, achieved a significant increase in overall throughput.

8. Finally, the conclusion of the research work.

1.5 Organization of Thesis

This thesis is organized as follows: Chapter (1) is an introduction to mobile ad hoc networks, a typical network scenario and problem statement. Chapter (2) is a detail description of background and problem domain of research work.

Chapter (3) is about related work in the field (literature review) and problem formulation with example and at the end problem definition.

Chapter (4) is solution and design methodology of the problem. Mechanism for route expiration and route discovery on the basis of load is provided (presents the enhanced route selection algorithm in detail.)

Chapter (5) gives a brief introduction to the network simulator (ns-2) and deals with implementation details of the enhanced algorithm and mechanism for mobility of nodes. Also gives details of the performance metrics chosen, simulation results and performance analysis.

Chapter (6) discusses conclusions drawn from simulation experiments and scope for future work.

Chapter 2
Background and Problem domain

2.1 Problem Domain

Mobile ad hoc networks are temporary networks useful for rescue, emergency and military operations. MANET is infrastructure less wireless network with no centralized management and control. Due to the mobile nature of nodes the network topology changes rapidly. Every node is autonomous in its operations and can rout data in the network received from other nodes. Every node is mobile and relies on batteries, so, power of nodes is another constraint [9].

The following are the salient features of MANETs:

- Dynamic Topologies: All nodes in the MANET generally move with different velocities and the network topology changes dynamically. Frequent link breaks are very common. Nodes may join the network or any nodes can leave the network. The dynamic changes in the network topology are the biggest challenge for routing in Ad Hoc Networks.

- Asymmetrical Communication: Each node in the Ad Hoc Network may have antennas with different transmission characteristics soothe symmetrical, bi-directional communication over the same link is not always possible. Sometimes, only unidirectional communication is possible.

- Bandwidth limitations: Since the nodes communicate via wireless links, the realized throughput in these networks as compared to a wired network of similar size is very small. The relatively lower capacity of the wireless links does not facilitate transmission of real-time or multimedia traffic. Moreover, the wireless links are quite error-prone, which may further degrade throughput due to upper layer retransmissions.

- Energy limitations: The nodes in the MANET are generally rely on battery .Hence, energy conservation techniques and energy-aware routing in MANETs become necessary.

Existing routing protocols in wired networks both link state and distance vector are not suitable for MANETs. These routing protocols distribute topological information across the network to update other nodes of topological changes. This mechanism is not suitable for MANETs, because there are frequent topological changes as the nodes move randomly causing frequent link breakages.

2.2 Routing Protocols for MANETs

Due to all aforementioned factors, routing in MANET is difficult. Lot of protocol has been defined for MANET routing to enhance performance. These protocols are classified as proactive protocols, reactive protocols and hybrid. Reactive protocols are source initiated on demand protocols. Proactive are table driven protocols. Most important and basic reactive protocols are AODV [10] and DSR [16]. Examples of proactive protocols are DSDV [17] and OLSR.TORA is hybrid protocol.

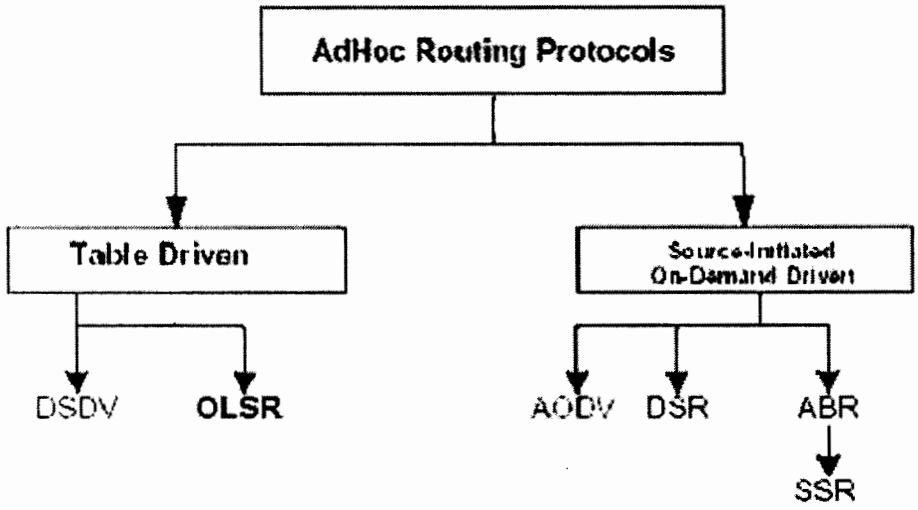


Figure 2: Classification of existing MANET protocols.

2.3 Table Driven or Proactive Protocols

Table-driven protocols or proactive protocols generate frequent updates of network topology information to maintain a consistent view of the network at all nodes. These nodes are maintaining tables containing topology information, so that any node wishing to communicate with any other node may do so by computing a route to the destination node from the table. It is fairly costly in terms of table size and control overhead to maintain a table of topological information of all nodes. The main disadvantage of this method is that the nodes may be maintaining topological information about nodes with which it may never communicate.

2.3.1 Destination Sequenced Distance Vector Routing Protocol

DSDV protocols control looping of message through the use of sequence. Each node maintains sequence number that is originated by the destination node and other related information.

There are two types of updates, periodic and triggered routing updates to maintain table consistency. When a link to the next hop is broken, any route through that next hop is assigned infinity metric and an updated sequence number. To prevent fluctuations due to simultaneous routing table updates DSDV uses settling time.

The DSDV give loop-free paths to the destination and detects optimal routes. Due to the periodical routing updates that are broadcast throughout the network more band width is required. When the number of nodes in the MANET increases the size of the routing table is also increases and the bandwidth required for updating them also increases. That's lead to more communication overhead in the network.

2.4 On-Demand or Reactive Protocols

Reactive protocols discover routes only when needed. When a node want to communicate with another node it checks with its existing information for a valid route to the destination. If one exists, the node uses that route for communication with the destination node. If no route to the destination is present then source node initiates a route request procedure, to which either the destination node or one of the intermediate nodes sends a reply back to the source node with a valid route. A soft state is maintained for each of these routes, if the routes are not used for some period of time, it is removed from the table and if it is used before it expires, then the lifetime of the route is increased

2.4.1 Dynamic Source Routing

DSR is source routing protocol. It avoids periodic routing advertisements instead; routes are dynamically determined based on temporary information in table or on the route discovery process results. Main advantage of source routing is that intermediate hops do not need to maintain routing information because the packets themselves contain all the necessary routing information in it.
The DSR protocol does not periodically transmit route updates, hence reducing control overhead, control overhead is more reduced when movement of nodes is less. When a mobile node wants to communicate with some destination, it will check its table for valid route. If the node does not have a route, it will initiate route discovery by broadcasting a RREQ packet, which contains the addresses of the source node and the destination, and a unique sequence number.

On receiving a RREQ, a node checks to see if it has already received a request with the same (source address, request id) pair. If there is no loop, it appends its own address to the route record in the route request packet and re-broadcasts to its neighbors. At the destination, the destination node then sends a route reply packet to the source with a copy of the route.
Route Maintenance is used to detect link failure due to network topology change. Each node along the route which detects failure uncast a Route Error packet (RERR) to the source of the packet.

The DSR [18] protocol is suitable for networks in which the mobility is less and latency as well. There is less power consumption but for larger networks

the data packet size become larger and there is need to expire routes those are no more needed.

2.4.2 Ad-hoc On-Demand Distance Vector Routing (AODV)

AODV removes the disadvantages of both DSR and DSDV and uses the advantages of both. It is loop-free, self-starting, and well scalable.
When a node want to communicate with any other node in the network it will start route discovery process and broad cost the RREQ message to its neighbors. The packet finally reached the destination; the destination will check the sequence number and number of hops it has passed through.
If it gets a RREQ message with fresher sequence number then it will send reply back to source. Each node remembers only the next hop only. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ packet with the same sequence number it already has in its table then it will discard the RREQ and do not forward it. Once the source node receives the RREP, it may begin to forward data packets to the destination. RREP is forwarded back to source node on the path established by the RREQ; AODV requires bidirectional links to perform its operation.
The main benefit of AODV over DSR is reduction of routing protocol overhead.
Comparing the features of on-demand and table-driven protocols, it is felt that reactive protocols offered more scope for improvement and, in general, offered better performance than their proactive counterparts. Among on-demand protocols, DSR and AODV were considered. Considering the relative merits and demerits of AODV and DSR as discussed in this chapter and given the relative ease of implementation of enhancements in AODV as compared to DSR, AODV is chosen over DSR as the routing protocol to be modified.

3.1 Literature Review

AODV is an easiest and most widely implemented protocol in comparison with other MANET routing protocols. Elizabeth, Santa and Chi-keno [15] discussed different MANET protocols. In the basic AODV there are many great features; those made AODV as most widely used protocol in MANET. These features are; it is especially design for MANET. It discovers route on-demand. AODV free of loop with quick convergence and scale well. It can easily fits into the existing protocol stack.
Lot of enhancements has been made to improve the performance of AODV. These enhancements are made on the basis of different MANET parameters. Most improvements are made by modifying route discovery mechanism by taking into account different network parameters like; Node status, power consumption, link status, packet overhead, congestion and others.

Ian and Luke [8] gave a lighter version of AODV called as AODVjr and shown that AODVjr perform well as basic AODV is performing normally. AODVjr is without sequence numbers, hop count, hello massage, RERR and precursor. The removal of these items made AODV easier to implement.

In [11], Lee and Gerla proposed a Dynamic Load-Aware Routing (DLAR) protocol. DLAR defines the network load of a mobile node as the number of packets in its interface queue.

In [12], Hassanein and Zhou proposed a Load-Balanced Ad hoc Routing (LBAR) protocol. They define network load in a node as the total number of routes passing through the node and its neighbors.

In [13], Wu and Harms proposed a Load-Sensitive Routing (LSR) protocol. In LSR, the network load in a node is defined as the summation of the number of packets being queued in the interface of the mobile host and its neighboring hosts. Even though the load metric of LSR is more accurate than those of DLAR or LBAR, but it does not consider the effect of access contentions in the MAC layer .Therefore, LSR produce contention delay.

In consideration of this contention delay, Sheu and Chen proposed a Delay-Oriented Shortest Path Routing protocol [14] and analyzed the medium access delay of a mobile node in IEEE 802.11 wireless network.
The above observations motivate us to develop a new load aware on-demand routing technique for better performance.

Yasser, Ghadeer and Asmahan [1] extend the ideas of both AODV and VON schemes. The main goal of the scheme is to build a stable route based on the hop-based AODV, velocity of nodes and traffic status of the nodes. Their scheme depends on the velocity and the traffic load of the nodes to decide which nodes will rebroadcast; to build a stable route.VON scheme divides the nodes into high speed nodes and low speed nodes. High speed nodes do not participate in the route discovery phase, since they produce unstable (volatile) routes. Unstable routes lead to broadcast more messages that will increase the congestion in the route.

S. Santtosh and B. Narasimhan [2] proposed a protocol for heterogeneous MANET. By using congestion aware routing which employ combined weight value as routing metric based on data rate, queuing delay, link quality and MAC overhead a cost factor is calculated and among the discovered routes, the route with minimum cost is chosen. Load status is calculated by sending Dummy RREP those are stored in the node to destination to estimate delay. Route with delay within the bound of application requirement is selected.

Hao, Zhao and Li [3] presented an AODV protocol with enhancements. In route discovery the route selection is made on the basis of node power which is power required for processing of packets, load status which is defined as a ratio of maximum length of queue and number of buffered packets and link state between nodes forward mechanism is developed which is used whenever a node receives a route request and then rebroadcast according to this mechanism. This mechanism decreases routing overhead and also results in higher packet delivery ratio but not suitable for congestion in transmissions of longer duration.

Li TANG and Ji [4] combine the shortest route selection criteria of AODV with real network status including link quality, the remaining power capacity and traffic load. This paper contributed by giving status adoptive routing which improves network life and delivery ratio.

Yu-Doe, II-Young and Sung-Joan [5] considered the node mobility and rediscover route after a fixed time. Path is resettled after rediscovery during the mobility of the nodes. But it does not take other factors like node power consumption, load on node and congestion into account. But if a node in the selected path is part of multi-path route then load on that node increases and greatly influence the performance negatively.

Duc and Harish [6] proposed a protocol CBR which is congestion adaptive routing protocol for MANET.CBR tried to prevent congestion in first attempt rather dealing with it reactively. A by-pass is designed as a sub path connecting the node with next non congested node. If congestion is expected on the next node then by-pass path is adopted.

Sanjeev, Vankateswarla and Bhupendra [7] proposed a protocol which includes load parameter in the RREQ message that helps in the selection of route with low congestion during route discovery process. This decreases the end to end delay for real time traffic. Route maintenance is done through RODV by not sending the error information back to the source results a decrease in routing overhead.

In all above mentioned protocols different techniques are used to increase the performance of the network .we have seen different load and congestion reduction methods in the literature, most of the literature proposed enchantments to reduce the congestion. Instead of reduction in congestion, avoidance is more suitable by decreasing load on the nodes that have greater impact on the performance of MANET. We have proposed such enhancement in combination with enhanced route discovery mechanism that is suitable for transmissions that requires routing connection for longer period of time.
The study of literature give us an insight of different routing approaches that has been made for MANET to achieve optimal performance in the given network scenarios. After deep literature understanding it is quite clear that because of the

different nature of MANET, not any single routing protocol work well with all network scenarios.

Congestion in the network is one of the biggest problems, it not only decreases the network performance but in our point of view also have negative impact on the network life as well. Different congestion aware techniques in the literature have impact on the performance of network only. We could use a mechanism in such a way that will have affect not only on performance but also on overall network life. And that is also suitable for transmissions that require longer time. Hence the performance of protocol will be optimized to a significant level.

3.2 Problem formulation in detail

In our enhancement we are actually avoiding the node with more loads not to take part in the route, because the discovered route will be used for transmissions that require longer period of time. Our proposed route discovery mechanism will take into account the load on each node and the selected route will expire after fixed interval of time. After an interval of time discovery mechanism run again to find new route because due to mobility it is most likely that more node come closer to the route which can provide new efficient route. In route discovery the load on node is considered, the route which is having less number of load nodes is selected. Because if a node with more load on it is selected in the route, the node becomes the part of multiple route paths we can say that a multi-path node.lot of processing will be on this node and it remains busy for longer period of time and as mobile nodes heavily rely on batteries the node will use more power and will die more quickly and made the some sort of network partitioning. When multi-path node dies it greatly impact the performance.
Our proposed protocol avoids this problem and the selection of node on the basis of load and path resetting after fixed interval of time, made other nodes those can take part in the MANET partially sharing the load of node which is multi-path. With these enhancements our proposed protocol not only increases the overall throughput of network as compare to the basic AODV routing protocol, but also increases the life of nodes which directly improves the network life and performance.

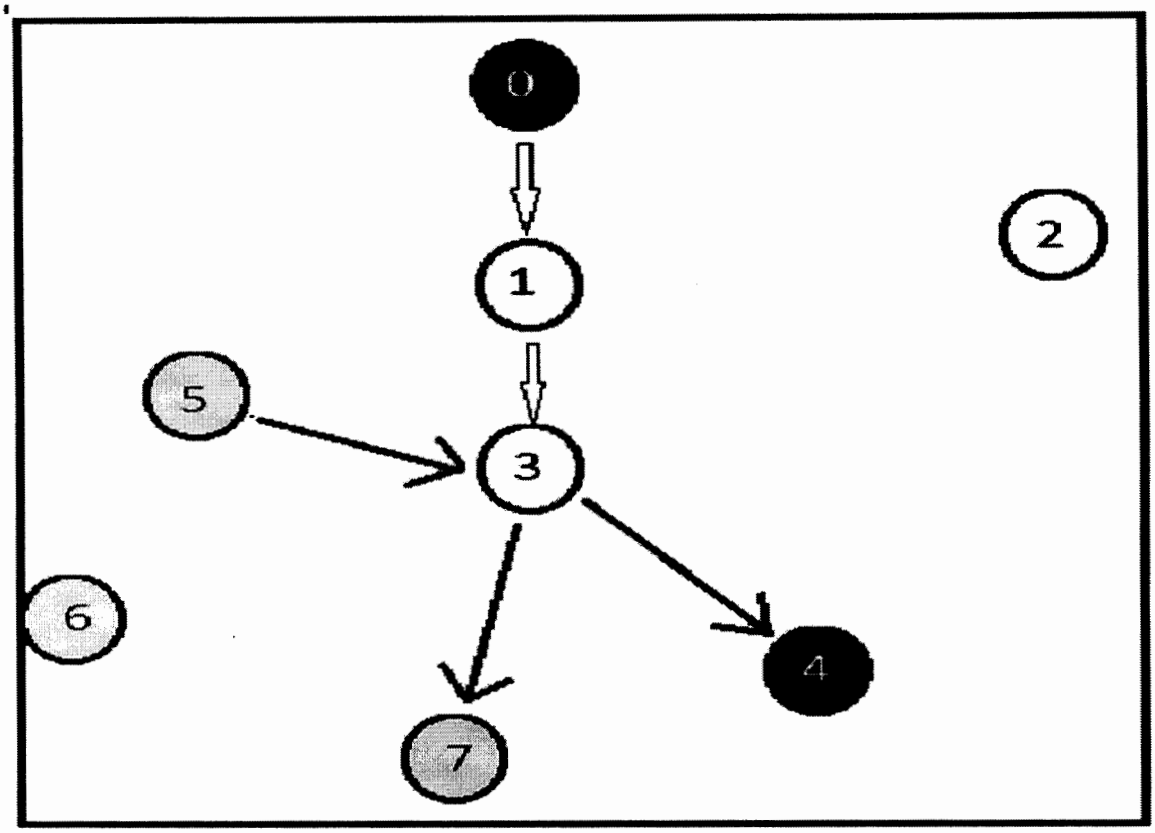


Figure 3: Congestion at node 3

In the fiure-2 the node 0 is the source node for the destination node 4 and node 5 is linked with node 7 both have same color as well. The node 0 and node 4 will remain connected unless until a link failure occurs due to the mobility or transmission is completed and node 5 and 7 will also remain connected as well. As we can see the node 3 is busier node then others and can be down earlier then other nodes in the network because it is part of tow active routes. we can avoid the node 3 from die earlier by shifting route to node 6 which is free this will improve the packet delivery ratio as well because at node 3 the packets can be dropped due to the congestion .

- Proposed protocol ensures shortest routing path, so the source send packets quickly to destination then original AODV.
- Sharing of load decreases the network congestion which directly leads to the decrease of overflowing of queuing buffer and packets loss. Hence packet delivery ratio increases.

- Sharing of load makes the nodes to live long as there burden is shared by other nodes that can take part in the route.

3.2.2 Problem

In the mobile adhoc networks route discovery algorithm of existing AODV [10] protocol is not suitable for the transmissions that require a link for longer period of time. Existing route discovery algorithm selects congested and longer routes, resulting into more packet drops and lesser overall throughput.
We proposed modifications in the route discovery algorithm and route selection criteria of existing AODV routing protocol. The performance of the protocol can be increased by modifying the existing route discovery algorithm of AODV routing protocol.

3.3 Route discovery Algorithm

Here is the brief description of basic AODV route discovery algorithm that has to be modified for the implementation of our solution.

- Fill in the essential fields of RREQ message
- Broad cost of RREQ Message
- Active route time is set to constant
- Reception of RREQ
- If it is my own message discard
- Check whether the reverse route entry is there

If no create the entry

- If yes, Compare

If ((seq no. of RREQ > seq no in the table) OR
((Seq no. of RREQ = seq no in the table) AND
(hop count of RREQ < hop count in table)))

- Update the routing entry otherwise leave it.

- If I am the destination Send RREP message to the updated reverse rout in the table.
- Route expire time is set to current time + active rout timeout.
- Transmission of data packets starts
- Route expire time is updated upon the reception of every data packet.

Chapter 4
Solution design and methodology

Our solution to the problem defined in the first chapter is to modify existing AODV route discovery mechanism in such a way that it will give greater performance then existing AODV protocol. We are not only enhancing the existing AODV protocol route discovery process but also expiring the route after fixed interval to take the mobility of the node into account. This will optimize the performance of the protocol.
Combine, it will not only increase the end-to-end throughput but also give the nodes a more life. Our proposed mechanism is explained in the section below.

4.1 Purposed Mechanism

Our purposed mechanism taking two factors in to accounts in the discovery and selection of route.

- Load on the nodes.
- Mobility of nodes.

First for the load on nodes, route request message and routing table is modified then for the mobility of nodes, changes are made in route expiration method of AODV protocol.

4.1.1 Route request message (RREQ) format

For the calculation of load on each node during the process of route discovery, actually we modified the basic AODV route discovery mechanism and routing table as well. In the route discovery massage (RREQ) we used reserved bytes for the value of load as shown in figure below.

Whenever a node sends a RREQ message to its neighbors the reserved field will be initialized with zero. When a neighbor will receive RREQ message it will calculate the number of packets in the queue and divide it with the size of the queue and add the value in the reserved field of the RREQ message just received for its neighbors. This process will be done at each node in the route to the destination. At the destination the reserved field is divided with the number of hop count. This will give us the average Load ratio on each node. The destination will decide on the basis of average value that to which route it has to send reply. Here the algorithm for the route selection will select the route with smaller average load

value and reply will be unicast to the route which have less congestion on it. In this way we are avoiding the busier nodes in the route .A node which have traffic load on it cannot be made more congested by selecting it in the new route as well. In this way a node which normally must be selected as in case of basic AODV will be avoided so load is shared with other nodes. This will increase network life.

4.1.2 Routing table

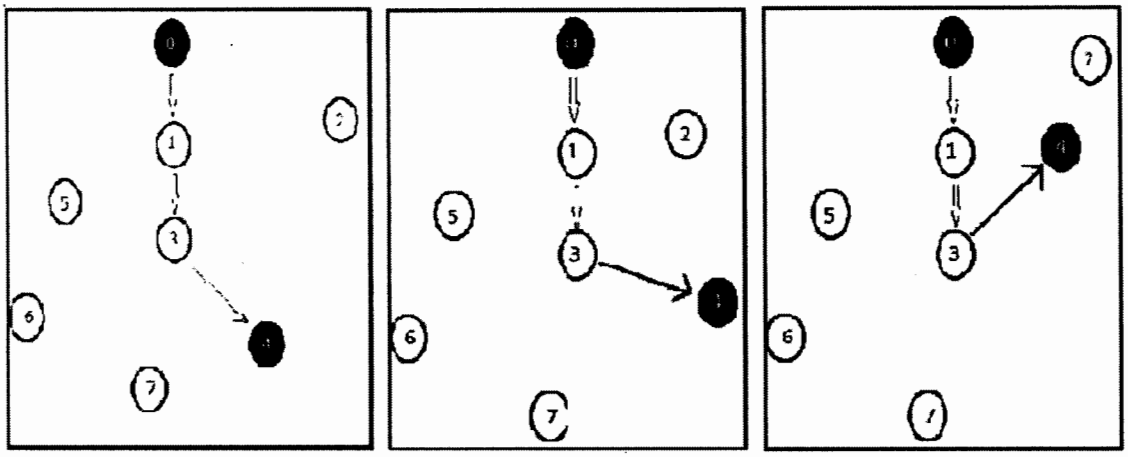
In the routing table we are adding another field with the name of Load which saves the loud status of the route in which the node is participating. This value will be updated whenever the node receives a RREQ message from another neighboring node for the same destination and if the sequence number is greater or hop count is less or load value is greater than average load value of the request message. Load field in the routing table provides us another additional criteria for the route selection .there are lot of question to be answered like; what happened if an intermediate node have fresh route to the destination, what happened if a route request have lesser number of hops but more load value etc. To deal with all such questions mentioned above what we did is;

⇒ At the destination the value in the reserved field will be divided with number of hop count. This will give us average load on each node in the route and it will help us to decide which route is best and have less traffic on it.

⇒ We are considering here that only destination will answer the RREQ message and there is no gratitude reply to the source will be produced. Actually, we are considering there is no intermediate route reply.

4.1.3 Route expiration process

Second part is the route expiration after fixed interval, in basic AODV the route that is selected for transmission not expires until unless the transmission is completed or route is no more required and link failure occurs. Here for our mechanism we have has to expire routes after fixed intervals because for the mobility factor we have to rediscover route and if there is any new efficient route available it will be selected for transmission.



After 38sec after 76sec after 116sec

Figure 4: MANET topology after fixed interval of time

Basically due to the mobility of nodes in MANET it is more likely that more nodes come closer to the active route from source to the destination that can provide another efficient route to the destination so instead of transmitting entirely on the previously selected route new efficient route will be adopted after fixed interval of time which can provide better through put as shown in the figure 5 above.

Figure 5 is showing the routing path of moving nodes. At first, the node 2 moved near node 1. Next, the node 2 moved back to its original location and finally node 4 moved near node 1at 90second.the routing table for the topology will look like as table 1.As shown below.

Time	Source	DES	Next hop	Hops
38	0	4	1	3
76		0	0	1
		4	1	3
116		4	1	3

Table 1: Routing table for fig.1showing entries

Above figure specifies that after an interval of time the topology of MANET is changing and more nodes come closer to the active route but the transmission is still through previously selected route in spite of the fact that an efficient route in terms of less number of hops is available which is

0->2->4 after 60 seconds and another more efficient direct from node 0 to node 4.

This is inefficient for the transmissions that require a link for longer duration if we have large amount of data to send through the link. So, we will expire the route after fixed time to utilize mobility factor of MANET in our proposed solution. Route expiration method will be presented in the later section.

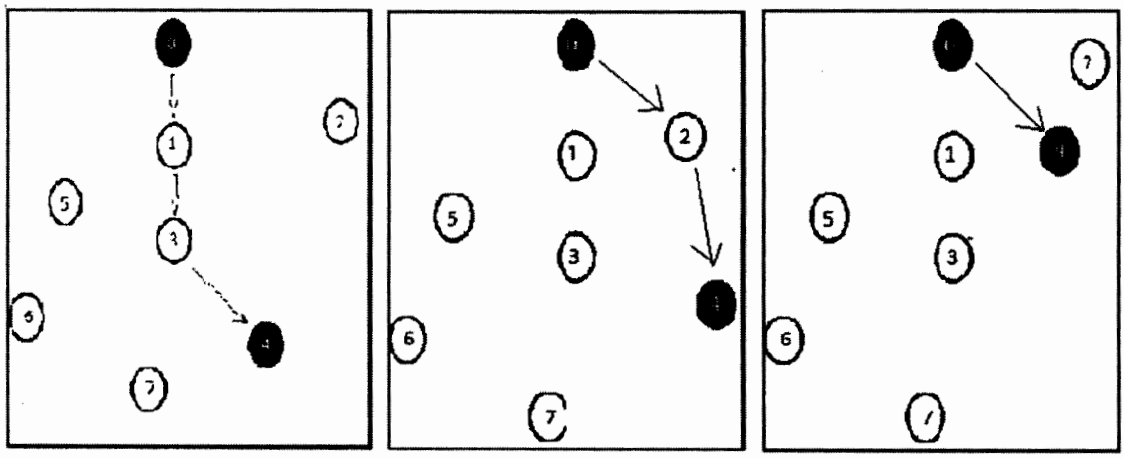
Here if we expire the route after fixed 38 seconds, then at 76 seconds the route rediscovery will taken place and the new efficient route that is **0->2->4** will be selected which is having less number of hops. And after 116 seconds another more efficient path will be selected. The routing table will be look like the table 2 below.

Time	Source	DES	Next hop	Hops
38	0	4	1	3
76		0	0	1
		4	1	2
116		4	1	1

Table 2: Routing table when route expires after fixed time

The table showing the routing entries when route expires after fixed interval, for all rediscoveries the destination is same as shown in column 3 of the table, column 5 showing number of hops after each rediscovery and route efficiency is clearly shown in terms of number of hops. After route discovery at fixed intervals the routing path will look like as in figure below.

If we take node mobility then it might possible that more node com closer to the active route that can provide better path to the same destination. If you look at the figures below it is quite clear that there is another more efficient route to the destination is available and instead of transmitting the entire data through one preselected route (0-->1-->3-->4) we can rediscover another at fixed interval of time as shown in the figure below.



At 38 second **at 76 second** **at 116 second**

Figure 5: Topology after implementation of new protocol

Hence, if route expires after fixed intervals it ensures shortest routing path for transmission due to the mobility of the nodes. So, the source sends packets more quickly to the destination than original AODV and end to end through put will increases. We are using this technique with load status based route selection technique to achieve more through put and less packet loss then basic AODV.

4.2 Mechanism of route selection on the basis of load status

When a source node wants to communicate with another node for which it has no routing information in its routing table then route discovery process starts. Source node broadcasts RREQ message to its neighbors. The RREQ message contains the reserved field in it. This field carries the load on the nodes. When a node receives the

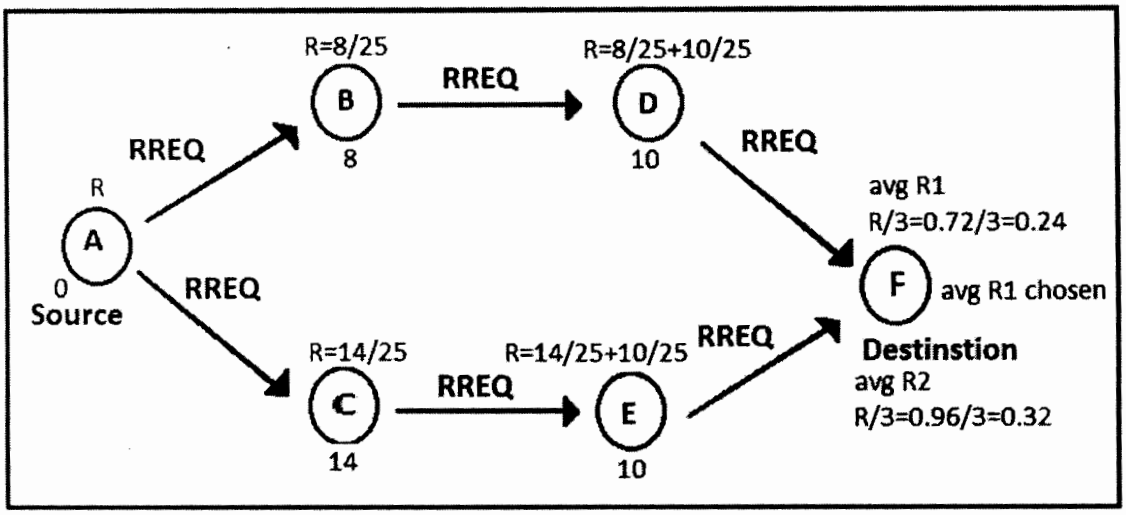


Figure 6: Route request process

RREQ message, the node will add value to the value present in RREQ message's reserved field .at each node the number of packet in the queue of that node is calculated and then ratio of number of packets to the size of queue is calculated .finally this ratio is added to the reserved field of the RREQ message as shown in figure 3 above.

In the figure above the value below the node is indicating the number of packets in the queue of that node and queue size is constant in this case to 25.the value of R above the node is showing the addition of ratio to the reserved field.

This process is repeated at each node through which the RREQ message is passed and reached up to the destination. The value of R is finally reached at the destination and the destination will divide R with hop count value of the REEQ message it has received and produces the average R. The destination will compare the average R value with all requests it received with same sequence number and

broad cast ID and unicast RREP message back to the neighbor whose average R value is less. Figure above is explaining in detail the whole process.

In this way the destination will receive two RREQ messages with the same number of hop count =2 but with two different average R value. Now the destination will select the optimal path on basis of average R on the nodes. Hence the path A->B->D->F will be selected as the route from source to the destination. As shown in the Figure 4 below.

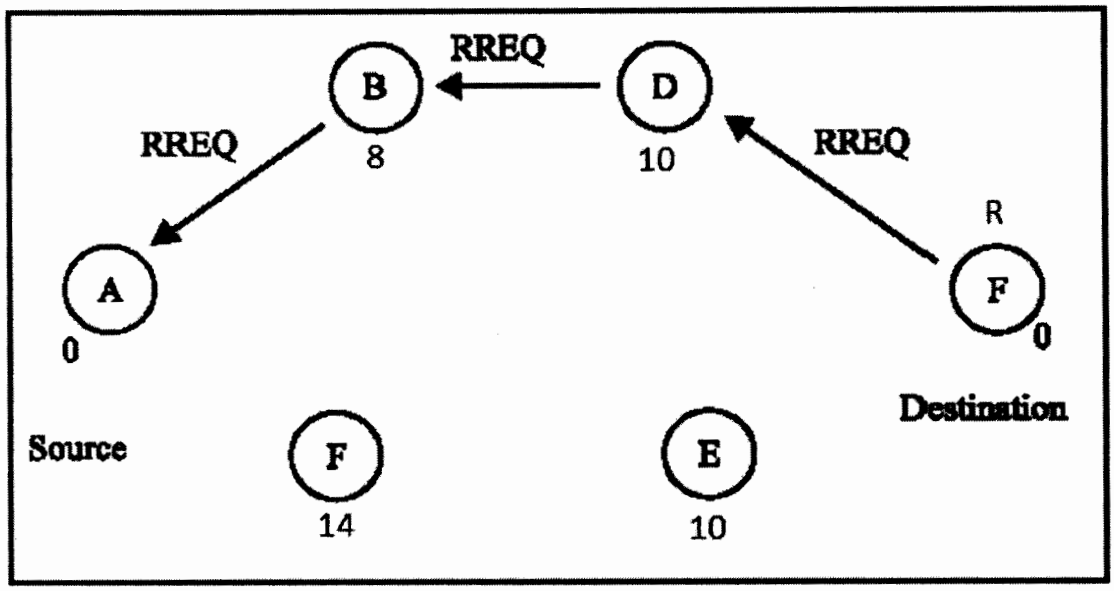


Figure 7: Route reply through selected path

Actually, the destination will selects one of the RREQ messages and ignore others for route reply on the basis of reserved value and it selects node D as in above figure 4. The destination node F will uncast the RREP message to the source in the selected optimal path. By selecting routes in this way we are able to overcome the limitation that the nodes with more load will come in the active route. This will prevent the node with more loads, not to take part in the route if another route is available. We are selecting the path with nodes with fewer loads, the congestion and the load in the network will be decreased.

We are using this load status based route selection mechanism for our MANET scenario to control the congestion and packet loss in the transmission from source to the destination. This will increase not only the end to end delivery but also the network over all life. This method will be used with fixed time route expiration for

the transmission of data that require a link for longer period of duration. So, in this way overall proposed results will be achieved.

By selecting rout on the basis of traffic load on the nodes we can avoid the packet drop and can save the node energy as well.
The figure-3 below is explaining the solution to the problem that was presented in the figure -2 above in chapter 4.

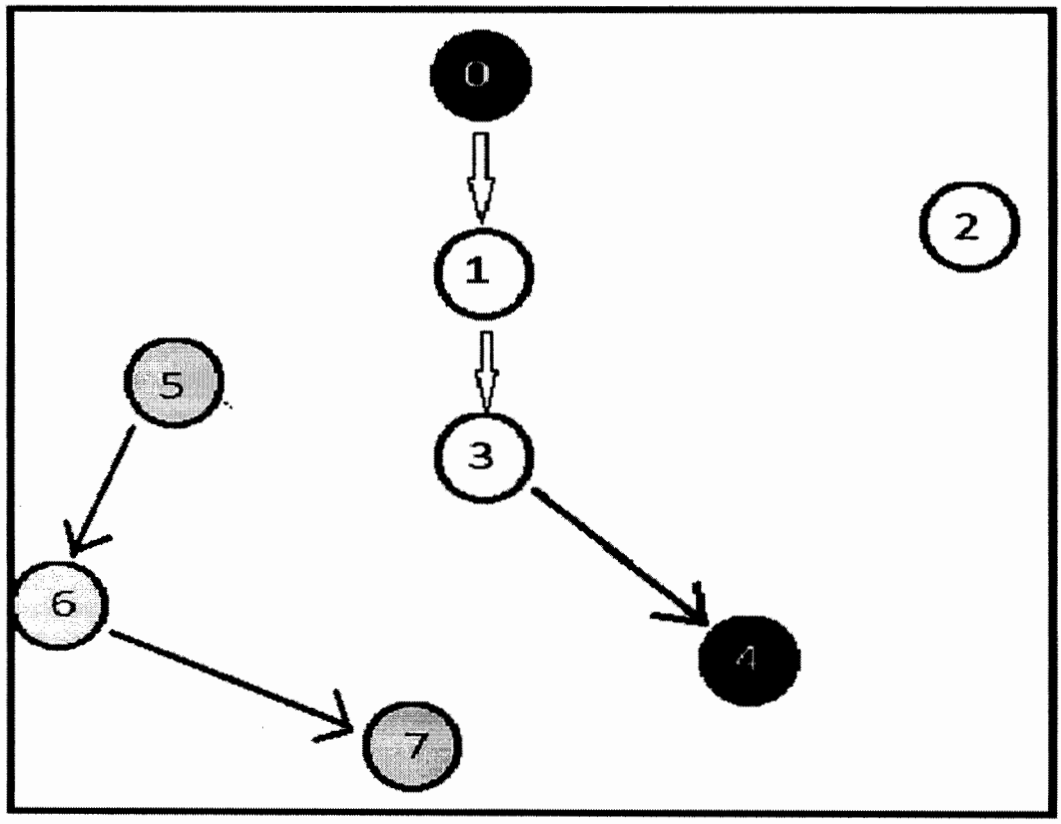


Figure 8: Shifted route due to congestion

In this way load is shifted from busier nodes to the other nodes, which provides load balancing in the network.

4.3 Mechanism for fixed route expiration

Now I will present the mechanism for fixed time route expiration. The basic AODV does not expire route until unless the route is no more required or a link failure occurs. In basic AODV the route expiration time updates (increases) with the transmission of every data packet at each node. The routing table of the node contains a life time field for each routing entry [10]. In the routing table entry the Lifetime field is either set from the control packet, or it is initialized to ACTIVE_ROUTE_TIMEOUT. Now the route may now be used to transmit data packets.

Upon the reception of every data packet at each node the route expiration time is updated with current time.

For our typical scenario of MANET we have to expire the route to check at some specific time that is there any more efficient route is available because due to mobility it is more likely that more nodes come closer to the active route that can provide better route. For this purpose my approach is that by not updating the route expiration time of route at the reception of each data packet .in this way route can be expire at preset time .As if we set ACTIVE_ROUTE _TIMEOUT=38 the route will expire exactly after 38 seconds. In our simulation it is done exactly the same as we claiming above paragraphs.
Route expiration in this way help us in finding of more efficient route for the transmission of data .A route with less number of hops help the data packets to reach the destination more quickly than through previously selected route which improve the end-to-end through put. This phenomenon is shown in the next implantation chapter.

4.4 Enhanced route discovery Algorithm

Following is the brief description of my enhanced route discovery algorithm that provides solution to the problem stated earlier in the problem statement.

- Fill in the essential RREQ fields and
- Initialize reserved field with zero.
- Broad cost of RREQ Message
- Active route time is set to a constant
- Reception of RREQ

- If it is my own message discard
- If I am the destination divide the reserved field with hop count
- Check whether the reverse route entry is there
  - if no create the entry
  - If yes

If ((seq no. of RREQ > seq no in the table) OR
((Seq no. of RREQ = seq no in the table) AND
(hop count of RREQ < hop count in table)) OR
((hop count of RREQ = hop count in table) AND
(traffic load of RREQ < load in routing table)))

- Updates the routing entry in the routing table otherwise leave it.
- If I am not the destination calculate traffic load
- Run Traffic load routine and add the value to the Reserved field of RREQ and forward the request message
- If I am the destination Send RREP message to the updated reverse rout in the table.
- Set the Route expire time to active rout timeout
- Transmission of data packets starts

A reserved field is used for traffic load calculation and another routine is added for the calculation ratio of number of packet to size of the queue.
A modification in the route reply criteria another check of traffic load value whether the RREQ reserved field value is greater than load value in the routing table load value. For the detail of the changes that has been made in each of the files and routines etc AODV protocol see Appendix at the end of the thesis.

Chapter 5
Implementation

5.1 Propagation Model

Two radio propagation models [21] that are considered for the protocol are: the Two-Ray Ground Ref lection model and free space propagation model. Friis free space attenuation model holds true, where the received signal strength is inversely proportional to the square of the distance (d) between the transmitting antenna and the observing point (d^2) while at far distances the received signal strength varies in accordance with the Two-Ray Ground propagation model (inversely proportional to d^4).

In our simulation model, for the parameters used, the transmission ranges of all antennas are assumed to be identical (400 meters), it is decided to use the Two-Ray propagation model for our algorithm at longer distances. The Two-Ray Ground Reflection model equation is as follows:

$$P_r = \frac{P_t * G_t * G_r * (h_t * h_r)^2}{d^4}$$

This equation means the signal power at receiver node has relation (1/d) with the signal power at the transmitter node.

For shorter distances free space propagation model is used and after the cross over distance (point) the two ray ground model is used. Other parameters used for propagation model in NS2 are as follows;

```
Pt = 0.28183815;          // transmit power
Gt = 1.0;                 // transmit antenna gain
Gr = 1.0;                 // receive antenna
freq = 914.0*10^6;        // frequency
sysLoss = 1.0;            // system loss

// for two-ray model
ht = 1.5;          // transmit antenna height
hr = 1.5;          // receive antenna height

// for shadowing model
pathlossExp_ = 2.0;       // path loss exponent
std_db_ = 4.0;            // shadowing deviation
```

```
dist0_ = 1.0;        // reference distance
prob = 0.95;         // correct reception rate
```

$$\text{Crossover distance} = \frac{4*(3.141582)*h_t*h_r}{\lambda}$$

For our algorithm in the simulation the crossover distance is 78.14 meter. After crossover point the two ray ground is used with assumption that some obstacles like building etc are in the line of sight.

5.2 Mobility model

We are using Random way point as a mobility model in our implementation of protocol because it's mostly used mobility model in the simulation of research. Network simulator 2 supporting this mobility model and most of the papers for MANET are using this, so, it is fair to compare the performance with the same mobility model.
Random way point is the entity mobility model in which a node travels in a direction with a speed and after some time it can change its speed and direction randomly.

5.3 Protocol Simulation and performance

Among NS-2, OPNET and Glomosim, the NS-2[19] simulator is chosen as the tool for our adhoc wireless simulations.NS-2 is over the other simulators.

The Network Simulator (NS-2) [20] is a discrete event simulator developed by the University of California at Berkeley. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. There is split programming interface for C++ and TCL language. TCL controls the definition and configuration of simulation. This split-programming approach has proven benefits over conventional programming methods. Also, NS-2 produce a detailed trace file and an animation file for each ad hoc network simulation that is very easier for analyzing the routing behavior of protocol.

5.5 Performance metrics

We will compare our enhanced load and mobility based (AODVLM) protocol with basic AODV .we evaluate mainly the performance according to following metric.

5.5.1 Average throughput

Throughput is the total number of packets received successfully in the given time.

5.5.2 Packet delivery ratio

It is the ratio of the number of packets received successfully to the total number of packets transmitted.

5.5.3 Packet loss

It is the packet loss rate of the transmission.

5.6 Simulation results

These are the simulation results that are produced after the implementation of basic AODV routing protocol and my enhanced (AODVLM) load and mobility based protocol.
First of all basic AODV is implemented on given MANET scenario in Ns2 and different performance parameters are calculated and results are presented in graphical form then AODVLM is checked on that given MANET scenario in NS2 and results are calculated.
We simulated the results in NS2 by taking parameters (pause time) on x-axis and performance metric (delivery ratio, packet loss rate and throughput on Y-axis.
The pause time is the time interval during which different parameters are calculated .In the figure below at equal intervals the throughput is calculated as shown in figure.

Comparison of performance with basic AODV

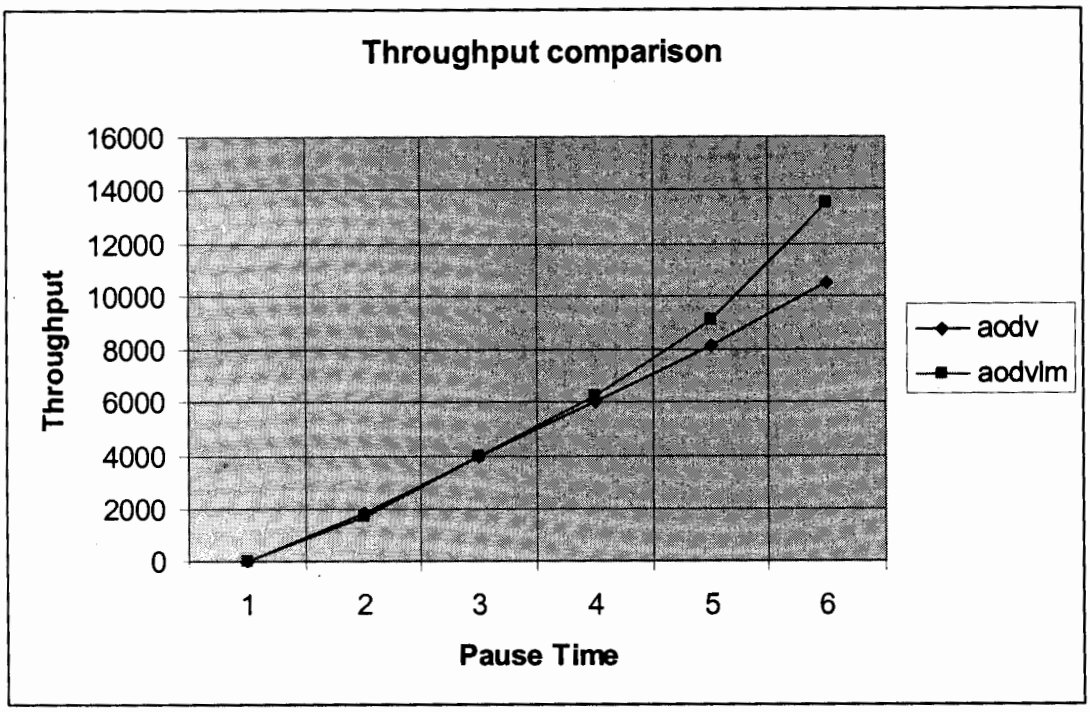


Figure 9: throughput vs. pause time

5.6.1 Throughput comparison

In above figure the throughput comparison between AODV and AODVLM is given and it is quite clear that the throughput of AODVLM is greater than basic aodv as indicated by green line in the graph. With the passage of time it is increasing and at time 150 it is at highest level. This shows that the longer the transmission, the higher is throughput .it proves the improvement in the performance of protocol. The throughput of AODVLM is greater than the basic AODV and it is suitable for the transmissions that require a link for longer period of time.

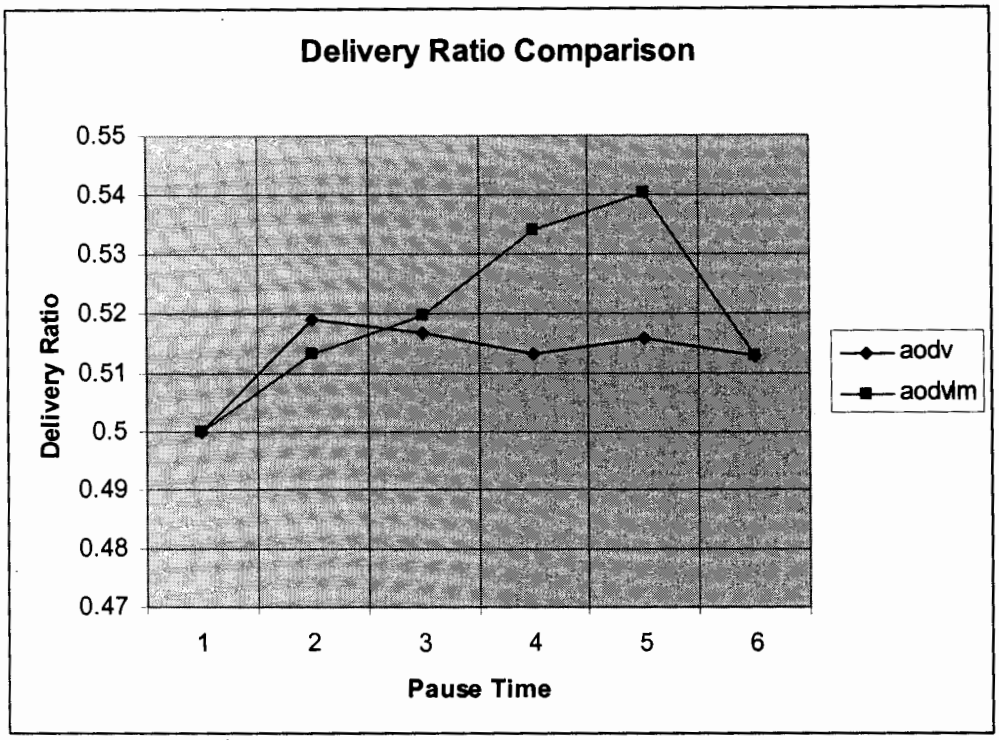


Figure 10: Delivery Ratio

5.6.2 End-to- end Delivery Rate comparison

Graph above showing the delivery ratio of both protocols. Green line is indicating the delivery ratio of AODVLM and it is clear that during the first half of the transmission the delivery ratio of AODVLM is slightly lower then but afterwards the overall delivery ratio of AODVLM is greater as shown by the green line in the above graph. So, we can say that our protocol will show lesser delivery ratio for shorter period transmission but for longer period transmissions it will produce greater delivery ratio. Hence delivery ratio is increased.

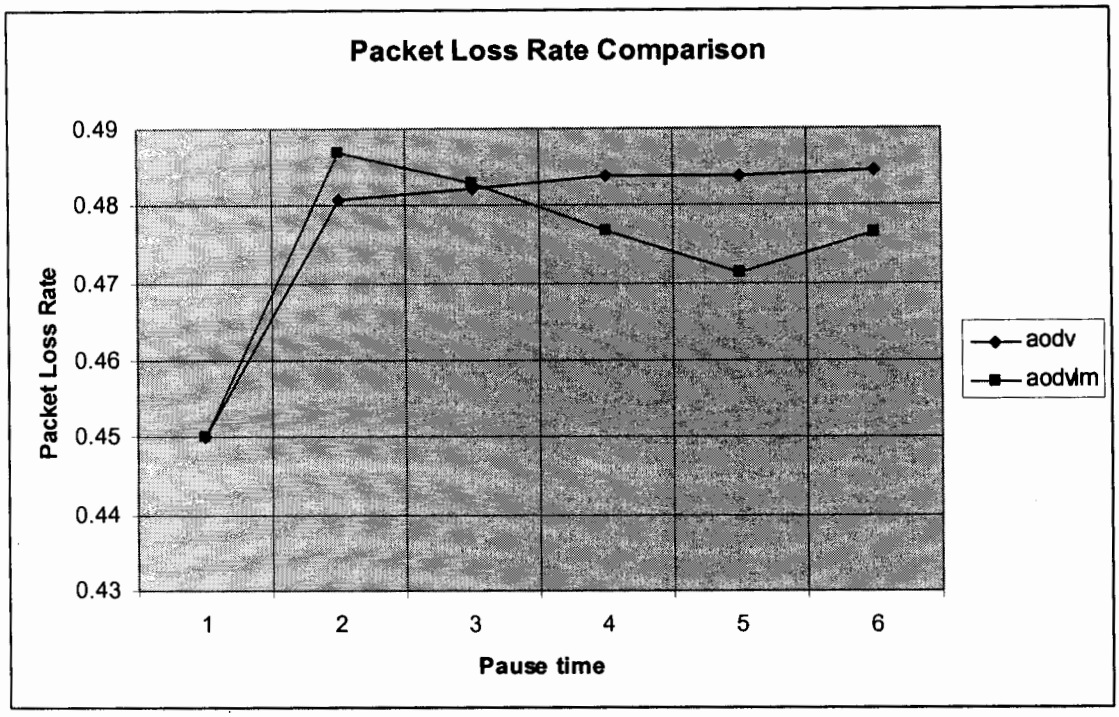


Figure 11: Packet loss rate (packet loss vs. pause time)

5.6.3 Packet Loss comparison

The packet loss rate of basic AODV and AODVLM is compared in the above graph. The packet loss rate of both protocols is calculated at fixed interval of time. First we calculated the number of packets during fixed intervals and packet loss then packet loss rate is calculated .The packet loss rate for the first half of the transmission for AODVLM is slightly greater but after that the packet loss of AODVLM is lesser and it remains lesser then basic AODV. So, packet loss is decreased 1% up to 2.5% which in turn increase delivery ratio.

5.7 Analysis of performance

From the above performance comparison we can easily analyze that the AODVLM is producing greater performance then basic AODV. There is 28% increment in the throughput of the basic AODV routing protocol, which is a greater improvement in the performance with slight increase in the packet loss at the first stage of the transmission. As for as the throughput is concerned the AODVLM shows greater improvement in later stage of the transmission and slight decrease in delivery ratio in the first part of transmission but overall the delivery ratio is higher than the basic AODV.

After the deep analysis of performance we can say that the AODVLM is more suitable mobile adhoc routing protocol then AODV for the transmissions that require a link for longer duration of time .Also, the improvement shown in the delivery ratio, throughput and packet loss rate during the transmission made it possible that the protocol AODVLM is also suitable for the shorter period transmissions with bit decrease in delivery ratio. (As shown in the delivery ratio graph above)

Hence, it is proven through simulation that the AODVLM routing protocol is better than basic AODV protocol for mobile ad hoc networks. So, it is better to use AODVLM routing protocol for the transmissions that require a connection for long duration then AODV.

Chapter 6
Conclusion and Future work

6.1 Conclusion

We developed a protocol with enhanced discovery mechanism that minimizes the congestion on the route. Instead of transmitting entire data through one rout, new efficient paths can be discovered from time to time during transmission by taking node mobility and traffic load into account. Our technique focuses to maintain efficient and less congested path for longer transmissions. With these enhancements our protocol increases the overall throughput of network and packet delivery ratio as compare to the basic AODV routing protocol and hence performance is optimized. The simulation result proves the improvement in the performance of the protocol.

6.2 Future work

An enhancement made in the route discovery of basic AODV improves its throughput and delivery ratio. The network life can also be improved and I am also working on this and I am in the implementation stage of my work.

6.3 References

[1] Yaser Khamayseh, Ghadeer Obeidat and Asmahan Abu Alhassan, "Enhanced VON-AODV Based on Delayed Rebroadcast Scheme", ACM, October 26, 2009.

[2] S. Santtosh baboo and B. Narasimhan, "A Hop-by-Hop Congestion-Aware Routing Protocol for Heterogeneous Mobile Ad-hoc Networks", International Journal of Computer Science and Information Security, Vol. 3,No. 1,2009 .

[3] Hao Jutao, Zhao Jingjing and Li Minglu, "Energy Level and Link State Aware AODV Route Request Forwarding Mechanism Research", WSEAS Transactions on Communications, Issue 2,Volume 8,February 2009.

[4] LI Ting, TANG Rui-bu and JI Hong, "Status adaptive routing with delayed rebroadcast scheme in AODV-based MANETs" .Science Direct, September 2008 Elsevier.

[5] Yu-Doe Kim, II-Young Moon and Sung-Joan Cho, "Enhanced AODV Routing Protocol through fixed Expire-time in MANET", International Conference on Network Application , protocol and Services 2008 21-22 November 2008.

[6] Duc A. Tran and Harish Raghvendra, "Congestion Adaptive Routing in Mobile Ad Hoc Networks", IEEE Transactions on Parallel and Distributed Systems,Vol.17 no.11,November 2006.

[7] Sanjeev Jain, Vankateswarla Pitti and Bhupendra Verma, "Real Time On-demand Distance Vector in Mobile Ad hoc Networks", Asian Journal of Information Technology 5(4): 454-459, 2005.

[8] Ian D. Chakeres and Luke Klien-Berndt, "AODVjr, AODV Simplified", Mobile Computing and Communications Review, Volume 6, Number 3, 2003.

[9] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance issues and Evaluation Considerations", Network Working Group, RFC2501, January 1999.

[10] C. Perkins, E. Belding Royer and S. Das, "Ad hoc On-demand Distance Vector (AODV) Routing", IETF RFC3561,July 2003.

[11] S. –J. Lee and M. Gerla, "Dynamic Load-Aware Routing in Ad Hoc Networks," in Proc. IEEE ICC'01, Helsinki, Finland, June 2001.

[12] H. Hassanein and A. Zhou, "Routing with Load Balancing in Wireless Ad Hoc Networks," in Proc. ACM MSWiM, Rome, Itary, July 2001.

[13] K. Wu and J. Harms, "Load-Sensitive Routing for Mobile Ad Hoc Networks," in Proc. IEEE ICCCN'01, Scottsdale, AZ, Oct. 2001.

[14] S-T. Sheu and J. Chen, "A Novel Delay-Oriented Shortest Path Routing Protocol for Mobile Ad Hoc Networks," in Proc. IEEE ICC'01, Helsinki, Finland, June 2001.

[15] Elizabeth M. Royer, Santa Barbara and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communication, April 1999.

[16] Charles E. Perkins, Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers." Proceedings of the Conference on Communications Architectures, Protocols and Applications, pages 234-244, London, England, August 1994.

[17] Johnson, D. and Maltz, D. (1996). "Dynamic source routing in ad hoc wireless networks," in Mobile Comp uting (ed.T. Imielinski and H. Korth), Kluwer Academic Publishers, Dordrecht, The Netherlands.

[18] Qin, L., "Pro-active Route Maintenance in DSR", M. Sc. Thesis, School of Computer Science, Carleton University, August 2001.

[19] The Network Simulator - ns-2 http://www.isi.edu/nsnam/ns/

[20] Kevin Fall and Kannan Varahan, editors. " NS Notes and Documentation". The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997.

[21] Mesut Gunes, and Martin Wenig ,"Models for Realistic Mobility and Radio Wave Propagation for Ad Hoc Network Simulations" Springer-Verlag London Limited,2009.(A chapter of book "Guide to Wireless Ad Hoc Networks").

Appendix

Appendix

1. In this appendix, the codes added to the appropriate files in the ns-2 simulator software are listed (along with the file names).

1-AODV.h

Some global constents that is to be set at start

```
class AODV;

#define MY_ROUTE_TIMEOUT       38                  // 38 seconds
#define ACTIVE_ROUTE_TIMEOUT   38                  // 38 seconds
#define REV_ROUTE_LIFE         6                    // 5  seconds
#define BCAST_ID_SAVE          6                   // 3 seconds


// No. of times to do network-wide search before timing out for
// MAX_RREQ_TIMEOUT sec.
#define RREQ_RETRIES           3
// timeout after doing network-wide search RREQ_RETRIES times
#define MAX_RREQ_TIMEOUT       10.0 //sec

/* Various constants used for the expanding ring search */
#define TTL_START     5
#define TTL_THRESHOLD 7
#define TTL_INCREMENT 2

// This should be somewhat related to arp timeout
#define NODE_TRAVERSAL_TIME     0.03           // 30 ms
#define LOCAL_REPAIR_WAIT_TIME  0.15 //sec

// Should be set by the user using best guess (conservative)
#define NETWORK_DIAMETER       30           // 30 hops
```

```
// Must be larger than the time difference between a node propagates a route
// request and gets the route reply back.

//#define RREP_WAIT_TIME    (3 * NODE_TRAVERSAL_TIME *
NETWORK_DIAMETER) // ms
//#define RREP_WAIT_TIME    (2 * REV_ROUTE_LIFE)  // seconds
#define RREP_WAIT_TIME       1.0  // sec

#define ID_NOT_FOUND   0x00
#define ID_FOUND       0x01
//#define INFINITY       0xff

// The followings are used for the forward() function. Controls pacing.
#define DELAY 1.0          // random delay
#define NO_DELAY -1.0      // no delay

// think it should be 30 ms
#define ARP_DELAY 0.01     // fixed delay to keep arp happy


#define HELLO_INTERVAL         1            // 1000 ms
#define ALLOWED_HELLO_LOSS     3            // packets
#define BAD_LINK_LIFETIME      3            // 3000 ms
#define MaxHelloInterval       (1.25 * HELLO_INTERVAL)
#define MinHelloInterval       (0.75 * HELLO_INTERVAL)
```

2-AODV.cc

Route request Message (RREQ) routine

```
void
AODV::sendRequest(nsaddr_t dst) {
// Allocate a RREQ packet
Packet *p = Packet::alloc();
```

```
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
aodv_rt_entry *rt = rtable.rt_lookup(dst);

 assert(rt);

 /*
  *  Rate limit sending of Route Requests. We are very conservative
  *  about sending out route requests.
  */

 if (rt->rt_flags == RTF_UP) {
   assert(rt->rt_hops != INFINITY2);
   Packet::free((Packet *)p);
   return;
 }

 if (rt->rt_req_timeout > CURRENT_TIME) {
   Packet::free((Packet *)p);
   return;
 }

 // rt_req_cnt is the no. of times we did network-wide broadcast
 // RREQ_RETRIES is the maximum number we will allow before
 // going to a long timeout.

 if (rt->rt_req_cnt > RREQ_RETRIES) {
   rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
   rt->rt_req_cnt = 0;
 Packet *buf_pkt;
   while ((buf_pkt = rqueue.deque(rt->rt_dst))) {
      drop(buf_pkt, DROP_RTR_NO_ROUTE);
   }
   Packet::free((Packet *)p);
```

```
  return;
 }

#ifdef DEBUG
  fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d\n",
          ++route_request, index, rt->rt_dst);
#endif // DEBUG

 // Determine the TTL to be used this time.
 // Dynamic TTL evaluation - SRD

 rt->rt_req_last_ttl = max(rt->rt_req_last_ttl,rt->rt_last_hop_count);

 if (0 == rt->rt_req_last_ttl) {
 // first time query broadcast
  ih->ttl_ = TTL_START;
 }
 else {
 // Expanding ring search.
  if (rt->rt_req_last_ttl < TTL_THRESHOLD)
    ih->ttl_ = rt->rt_req_last_ttl + TTL_INCREMENT;
  else {
  // network-wide broadcast
    ih->ttl_ = NETWORK_DIAMETER;
    rt->rt_req_cnt += 1;
  }
 }

 // remember the TTL used  for the next time
 rt->rt_req_last_ttl = ih->ttl_;

 // PerHopTime is the roundtrip time per hop for route requests.
 // The factor 2.0 is just to be safe .. SRD 5/22/99
 // Also note that we are making timeouts to be larger if we have
 // done network wide broadcast before.
```

```
rt->rt_req_timeout = 2.0 * (double) ih->ttl_ * PerHopTime(rt);
if (rt->rt_req_cnt > 0)
  rt->rt_req_timeout *= rt->rt_req_cnt;
rt->rt_req_timeout += CURRENT_TIME;

// Don't let the timeout to be too large, however .. SRD 6/8/99
if (rt->rt_req_timeout > CURRENT_TIME + MAX_RREQ_TIMEOUT)
  rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
rt->rt_expire = 0;

#ifdef DEBUG
 fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d, tout %f ms\n",
                 ++route_request,
                   index, rt->rt_dst,
                   rt->rt_req_timeout - CURRENT_TIME);
#endif// DEBUG


 // Fill out the RREQ packet
 // ch->uid() = 0;
 ch->ptype() = PT_AODV;
 ch->size() = IP_HDR_LEN + rq->size();
 ch->iface() = -2;
 ch->error() = 0;
 ch->addr_type() = NS_AF_NONE;
 ch->prev_hop_ = index;          // AODV hack

 ih->saddr() = index;
 ih->daddr() = IP_BROADCAST;
 ih->sport() = RT_PORT;
 ih->dport() = RT_PORT;

 // Fill up some more fields.
 rq->rq_type = AODVTYPE_RREQ;
```

```
rq->rq_hop_count = 1;
rq->rq_bcast_id = bid++;
rq->rq_dst = dst;
rq->rq_dst_seqno = (rt ? rt->rt_seqno : 0);
rq->rq_src = index;
seqno += 2;
assert ((seqno%2) == 0);
rq->rq_src_seqno = seqno;
rq->rq_timestamp = CURRENT_TIME;
rq->reserved[0]=0;
Scheduler::instance().schedule(target_, p, 0.);

}
```

RREQ Reception Routine

```
void
AODV::recvRequest(Packet *p) {
struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
aodv_rt_entry *rt;
u_int8_t aa;
 /*
  * Drop if:
  *      - I'm the source
  *      - I recently heard this request.
  */

 if(rq->rq_src == index) {
#ifdef DEBUG
   fprintf(stderr, "%s: got my own REQUEST\n", __FUNCTION__);
#endif // DEBUG
   Packet::free(p);
   return;
 }
```

```
 if (id_lookup(rq->rq_src, rq->rq_bcast_id)) {

#ifdef DEBUG
   fprintf(stderr, "%s: discarding request\n", __FUNCTION__);
#endif // DEBUG

   Packet::free(p);
   return;
 }

 /*
  * Cache the broadcast ID
  */
 id_insert(rq->rq_src, rq->rq_bcast_id);



 /*
  * We are either going to forward the REQUEST or generate a
  * REPLY. Before we do anything, we make sure that the REVERSE
  * route is in the route table.
  */
 aodv_rt_entry *rt0; // rt0 is the reverse route

   rt0 = rtable.rt_lookup(rq->rq_src);
   if(rt0 == 0) { /* if not in the route table */
   // create an entry for the reverse route.
     rt0 = rtable.rt_add(rq->rq_src);
   }

   rt0->rt_expire = max(rt0->rt_expire, (CURRENT_TIME +
REV_ROUTE_LIFE));

   if ( (rq->rq_src_seqno > rt0->rt_seqno ) ||
```

```
        ((rq->rq_src_seqno == rt0->rt_seqno) &&
         (rq->rq_hop_count < rt0->rt_hops))||
        ((rq->rq_hop_count == rt0->rt_hops) &&
        (rq->reserved [0] < rt0->rt_load)) )
{
  // If we have a fresher seq no. or lesser #hops for the
  // same seq no., update the rt entry. Else don't bother.
rt_update(rt0, rq->rq_src_seqno, rq->rq_hop_count, ih->saddr(),
             max(rt0->rt_expire, (CURRENT_TIME + REV_ROUTE_LIFE))
);
    if (rt0->rt_req_timeout > 0.0) {
    // Reset the soft state and
    // Set expiry time to CURRENT_TIME + ACTIVE_ROUTE_TIMEOUT
    // This is because route is used in the forward direction,
    // but only sources get benefited by this change
     rt0->rt_req_cnt = 0;
     rt0->rt_req_timeout = 0.0;
     rt0->rt_req_last_ttl = rq->rq_hop_count;
     rt0->rt_expire =  ACTIVE_ROUTE_TIMEOUT;
    }

    /* Find out whether any buffered packet can benefit from the
     * reverse route.
     * May need some change in the following code - Mahesh 09/11/99
     */
    assert (rt0->rt_flags == RTF_UP);
    Packet *buffered_pkt;
    while ((buffered_pkt = rqueue.deque(rt0->rt_dst))) {
     if (rt0 && (rt0->rt_flags == RTF_UP)) {
       assert(rt0->rt_hops != INFINITY2);
       forward(rt0, buffered_pkt, NO_DELAY);
     }
    }
  }
  // End for putting reverse route in rt table
```

```
/*
 * We have taken care of the reverse route stuff.
 * Now see whether we can send a route reply.
 */

rt = rtable.rt_lookup(rq->rq_dst);

// First check if I am the destination ..

if(rq->rq_dst == index) {

#ifdef DEBUG
  fprintf(stderr, "%d - %s: destination sending reply\n",
              index, __FUNCTION__);
#endif // DEBUG


  // Just to be safe, I use the max. Somebody may have
  // incremented the dst seqno.
  seqno = max(seqno, rq->rq_dst_seqno)+1;
  if (seqno%2) seqno++;

  sendReply(rq->rq_src,         // IP Destination
        1,                  // Hop Count
        index,              // Dest IP Address
        seqno,              // Dest Sequence Num
        MY_ROUTE_TIMEOUT,   // Lifetime
        rq->rq_timestamp);  // timestamp

  Packet::free(p);
 }

// I am not the destination, but I may have a fresh enough route.
```

```
else if (rt && (rt->rt_hops != INFINITY2) &&
            (rt->rt_seqno >= rq->rq_dst_seqno) ) {

  //assert (rt->rt_flags == RTF_UP);
  assert(rq->rq_dst == rt->rt_dst);
  //assert ((rt->rt_seqno%2) == 0);      // is the seqno even?
  sendReply(rq->rq_src,
        rt->rt_hops + 1,
        rq->rq_dst,
        rt->rt_seqno,
          (u_int32_t) (rt->rt_expire - CURRENT_TIME),
          //          rt->rt_expire - CURRENT_TIME,
        rq->rq_timestamp);
  // Insert nexthops to RREQ source and RREQ destination in the
  // precursor lists of destination and source respectively
  rt->pc_insert(rt0->rt_nexthop); // nexthop to RREQ source
  rt0->pc_insert(rt->rt_nexthop); // nexthop to RREQ destination

#ifdef RREQ_GRAT_RREP

  sendReply(rq->rq_dst,
        rq->rq_hop_count,
        rq->rq_src,
        rq->rq_src_seqno,
          (u_int32_t) (rt->rt_expire - CURRENT_TIME),
          //          rt->rt_expire - CURRENT_TIME,
        rq->rq_timestamp);
#endif

// TODO: send grat RREP to dst if G flag set in RREQ using rq-
>rq_src_seqno, rq->rq_hop_counT
```

// DONE: Included gratuitous replies to be sent as per IETF aodv draft specification. As of now, G flag has not been dynamically used and is always set or reset in aodv-packet.h --- Anant Utgikar, 09/16/02.

```
        Packet::free(p);
}
/*
 * Can't reply. So forward the  Route Request
 */
else {
  aa = rqueue.count();

  rq->reserved[0]+=aa;
   //rt->rt_load +=2;
  ih->saddr() = index;
  ih->daddr() = IP_BROADCAST;
  rq->rq_hop_count += 1;
  //rq->reserved[0] = 2;
  // Maximum sequence number seen en route
  if (rt) rq->rq_dst_seqno = max(rt->rt_seqno, rq->rq_dst_seqno);
  forward((aodv_rt_entry*) 0, p, DELAY);
}

}
```

3- QUEUE.cc

Traffic load calculation routine

```
int
aodv_rqueue::count() {
Packet *p, *prev = 0;
int cnt = 5;

 for(p = head_; p; p = p->next_) {
```

```
 cnt++;
 prev = p;
}
return cnt/size of queue;

}
```

4-ROUTING TABLE.h

Routing table entry with additional load field
load field declaration

```
/*
 Route Table Entry
*/

class aodv_rt_entry {
      friend class aodv_rtable;
      friend class AODV;
        friend class LocalRepairTimer;
 public:
      aodv_rt_entry();
      ~aodv_rt_entry();

      void          nb_insert(nsaddr_t id);
      AODV_Neighbor*  nb_lookup(nsaddr_t id);

      void          pc_insert(nsaddr_t id);
      AODV_Precursor* pc_lookup(nsaddr_t id);
      void            pc_delete(nsaddr_t id);
      void            pc_delete(void);
      bool            pc_empty(void);

      double        rt_req_timeout;  // when I can send another req
      u_int8_t      rt_req_cnt;      // number of route requests
```

```
        u_int8_t      rt_load;
protected:
    LIST_ENTRY(aodv_rt_entry) rt_link;

    nsaddr_t      rt_dst;
    u_int32_t     rt_seqno;
     /* u_int8_t   rt_interface; */
    u_int16_t     rt_hops;              // hop count
     int          rt_last_hop_count; // last valid hop count
    nsaddr_t      rt_nexthop;                 // next hop IP address
     /* list of precursors */
    aodv_precursors rt_pclist;
    double        rt_expire;            // when entry expires
    u_int8_t      rt_flags;

     /*
          >>important note::::copied from aodv.h*/

     //void         enque(aodv_rt_entry *rt, Packet *p);
    //Packet*       deque(aodv_rt_entry *rt);
     //aodv_rqueue       rqueue;


     // >>up to here

#define RTF_DOWN 0
#define RTF_UP 1
#define RTF_IN_REPAIR 2
    /*
     * Must receive 4 errors within 3 seconds in order to mark
     * the route down.
    u_int8_t      rt_errors;     // error count
    double        rt_error_time;
#define MAX_RT_ERROR          4      // errors
#define MAX_RT_ERROR_TIME     3      // seconds
```

```
        */
#define MAX_HISTORY        3
        double              rt_disc_latency[MAX_HISTORY];
        char            hist_indx;
       int              rt_req_last_ttl;       // last ttl value used
        // last few route discovery latencies
        // double              rt_length [MAX_HISTORY];
        // last few route lengths

       /*
        * a list of neighbors that are using this route.
        */
       aodv_ncache         rt_nblist;
};
```

5-ROUTING TABLE.cc

Load field initialization

```
/*
  The Routing Table
*/

aodv_rt_entry::aodv_rt_entry()
{
int i;

 rt_req_timeout = 0.0;
 rt_req_cnt = 0;
 rt_load=2;
 rt_dst = 0;
 rt_seqno = 0;
 rt_hops = rt_last_hop_count = INFINITY2;
 rt_nexthop = 0;
 LIST_INIT(&rt_pclist);
 rt_expire = 0.0;
 rt_flags = RTF_DOWN;
```

these are the files that has been modified for enhancement of AODV protocol in NS2.