ABSTRACT

HOP-LIMITED ROUTING FOR MULTIHOP CELLULAR NETWORKS

by
Cem Dincer

In recent years, conventional cellular systems have experienced evolution in fields of data oriented services. During this period, the requirement for high data-rate stimulated new research proposals, which resulted in a new architecture: Multihop Cellular Networks (MCN), where multihop structure enables mobile stations to forward packets from other mobile stations to the base station on the uplink, and in turn, forward packets to other mobile stations from the base station on the downlink. In this thesis, a new routing algorithm is introduced for MCNs in order to limit the number of hops between the base station and the mobile stations with given delay constraints.

The capacity of MCNs is restricted due to intensive traffic in the network since all nodes has the capability of sending packets simultaneously. The analysis of average end-to-end delay in high bit-rate data transmission reveals that minimizing end-to-end delay with a proper scheduling scheme guarantees the aim of limiting number of hops in MCNs. The proposed algorithm showed that the intensive traffic can be absorbed by the base station by limiting the number of hops between the base station and the mobile stations.

HOP-LIMITED ROUTING
FOR MULTIHOP CELLULAR NETWORKS

by
Cem Dincer

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Telecommunications

Department of Electrical and Computer Engineering

May 2003

APPROVAL PAGE

HOP-LIMITED ROUTING FOR MULTIHOP CELLULAR NETWORKS

Cem Dincer

Dr. Sirin Tekinay — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Ali N. Akansu — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Ali Abdi — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Cem Dincer

Degree: Master of Science

Date: May 2003

Undergraduate and Graduate Education:

- Master of Science in Telecommunications,
  New Jersey Institute of Technology, Newark, NJ, 2003

- Bachelor of Science in Computer Engineering,
  Istanbul University, Istanbul, Turkey, 1999

Major: Telecommunications

To my parents

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. Sirin Tekinay, who served as my research advisor and gave me support, valuable suggestions and reassurance during my studies. Special thanks are given to Dr. Ali Akansu and Dr. Ali Abdi for actively participating to my committee.

I would also like to thank to my all lab friends at New Jersey Center for Multimedia Research (NJCMR) for their help to my work.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Chapter **Page**

LIST OF FIGURES

Chapter Page

CHAPTER 1

INTRODUCTION

Ad hoc network nodes are randomly distributed in a given geographic area. The nodes send packets to others or forward packets to reach the destination nodes in multiple hops. There are many advantages of these networks such as low cost and easy and fast deployment. However, such networks pose many challenges due to lack of infrastructure and the need for decentralized control and dynamic topology.

Cellular networks have rapidly evolved in the recent years. However, the increase in demand for high data rate transmissions in cellular networks requires a new architecture, which must support unpredictable mobility of users without an additional infrastructure. Multihop Cellular Networks (MCN) have appeared as a new architecture by utilizing a mixture of cellular and ad-hoc networks. The multihop structure of cellular networks enables mobile stations to forward packets from other stations to base station in uplink and forward packets to other mobile stations in downlink.

Furthermore, MCNs have several additional benefits beyond offering high data-rates. In recent studies, it has been shown that MCNs have many advantages when compared with single hop cellular networks. (i) By utilizing ad-hoc flexibility the network robustness and scalability of the system increases. (ii) Due to routing alternatives, load balancing between cells and extension of cell coverage area results in capacity enhancement. (iii) By sending packets over relaying nodes, the transmit power is reduced by breaking down the distance between source node and base station into smaller segments due to power law of path loss.

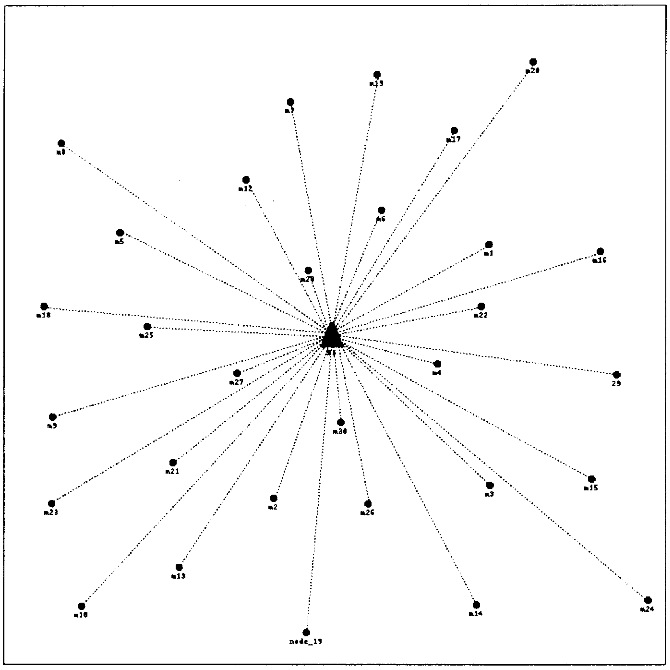

Figure 1.1 Single-hop cellular networks.

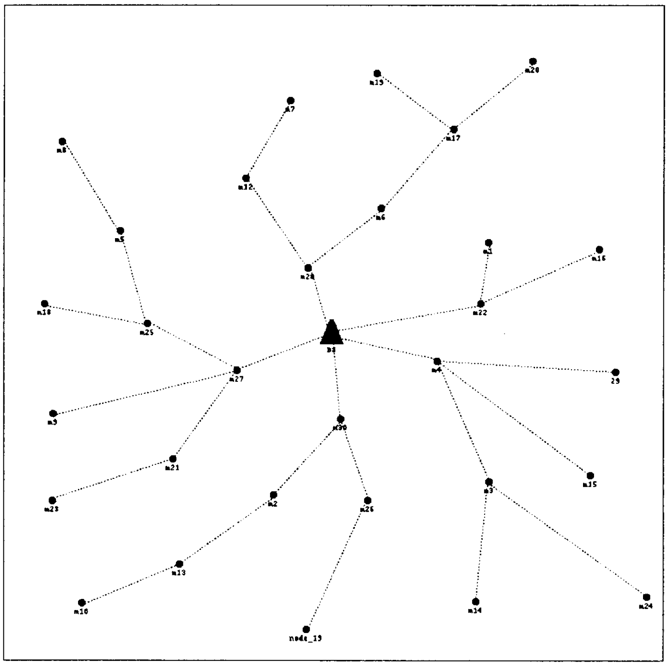

Figure 1.2 Multi-hop cellular networks.

However, in interference-limited systems such as Code Division Multiple Access (CMDA) wireless networks, this path loss reduction may not necessarily increase the capacity since power scale of all nodes does not affect the signal to interference ratio [13]. It is also expected that since all the mobile stations are sending data to base station, there will be a potential congestion in the area that close to base station.

In this chapter, the potential advantages of MCN over single hop cellular networks and previously proposed architectures are presented. It is shown that MCN results in higher performance improvement in terms of system capacity, coverage and power consumption.

1.1 Coverage Area

In future broadband mobile communication systems, faster data transmission services are expected. The distance from base station, at which sufficient signal to interference and noise ratio (SINR) is retained, becomes shorter for high bit-rate data transmissions if the transmit power is kept constant [6]. As a result, the cell size should be reduced to support high data-rate transmission and this fact will impact the economical deployment of cellular systems. In this point, the multihop connection scheme emerges as promising approach to enhancing the coverage area without a significant additional infrastructure cost. In this scheme, one or more mobile terminals relay transmission signals using the same access scheme between a mobile user and its destination.

In wireless networks, data transmission requires transmit power almost in proportion to bit rate. Transmit power; however, is limited especially in a mobile terminal by its battery capacity. As the transmission bit rate increases, the area coverage decreases, assuming that the transmit power remains at constant level. Wide coverage area for high bit-rate data transmission in cellular networks, therefore, leads to high base-station density and this will significantly increase infrastructure cost. The current solution to this problem is that the coverage area for high bit-rate data transmission is limited to the vicinity of the base station or the place where line of sight propagation is available, while low bit-rate data transmission service can cover the entire cell.

The multihop connection scheme is a promising approach to solve the coverage problem at a reasonable infrastructure cost. In this scheme, one or more mobile terminals relay transmission signals by using the same access scheme between a mobile user and its destination base station. The multihop scheme enables the establishment of connections even the signal from an end user mobile station can not directly reach the base station even with its available maximum transmit power.

In [6], it has shown that a significant area coverage enhancement can be obtained by using multihop connection. The multihop scheme achieved 90 % of area coverage, while it was approximately 50% in the single hop case.

1.2 Capacity Enhancement

On the other hand, the capacity of CDMA cellular systems is restricted by the interference of other users in the network. Reducing the transmit power at each mobile terminal along with using multi-user detection techniques is an effective approach to increase the signal to interference ratio (SIR) in the entire network. The transmit power reduction, therefore, contributes to the system capacity enhancement. By applying mutlihop scheme to the CDMA system, inter-cell interference can be reduced because of the average transmit power reduction and thus the capacity is expected to increase. However, transmission on short hops reduces interference; the network traffic is artificially increased due to many retransmissions of the same data [3]. In this thesis, a new transmission strategy Forward-to-Base Station (FBS) is introduced, which will forward the incoming packets to the base station with given delay constraints. As a result, the network traffic will be held at a stable level and number of hops between a mobile user and base station at a maximum level.

1.3 Load Balancing

With the advent of the Internet, especially wireless access to the Internet, wireless data traffic is expected to exacerbate the demand for bandwidth [7]. At the same time, various efforts in providing different access services such as wireless LANs, ad hoc networks, Bluetooth and home RF networks are further stimulating the growth of wireless traffic. Moreover, continued proliferation of these services will call for interoperability between heterogeneous networks. Consequently, such interoperability will create heavier traffic from LANs to ad hoc networks, Bluetooth devices and cellular infrastructure.

Therefore, the traffic in future cellular systems will be burstier and unevenly distributed then conventional voice traffic. It is anticipated that congestion will occur in peak usage hours. Presence of unbalanced traffic exacerbates the problem of limited capacity in existing cellular network systems. Some cells may be heavily congested, while other cells may still have enough available data channels. In other words, even though the traffic load does not reach the maximum capacity of the entire system, a significant number of calls may be blocked and dropped due to localized congestion. A previously proposed model, Integrated Cellular and Ad Hoc Relay (iCAR) [7], has proved that by using load balancing among different cells the congestion problem can be overcome. The basic idea of the proposed system (iCAR) is to place a number of ad hoc relay stations (ARSs) at strategic locations, which can be used to relay signal between mobile terminals and base stations. By using ARSs, it is possible to divert traffic from a congested cell to another non-congested cell. This helps to circumvent congestion, and makes it possible to maintain or hand-off calls of mobile terminals that are moving into a congested cell, or to accept new calls of mobile terminals in a congested cell. Moreover, even in the ideal case where every mobile terminal in an area that is not covered by any base station can find a relaying route through other mobile terminals. The multihop approach will neither increase the system capacity nor decrease the call blocking/dropping probability unless a large percentage of the calls are intra-cell calls, which usually is not the case in practice. Finally, it has been shown that iCAR model increases channel capacity of a congested cell by approximately 70 %.

1.4 Power Consumption

In a MCN, mobile terminals rely on short lifetime batteries and therefore, energy conversation is a critical design criterion. Transmit power in wireless networks can be reduced by breaking down the distance between two communication points into smaller segments.

By taking multiple hops the transmit power can be reduced due to reduced transmission range. However, CDMA cellular networks capacity is interference limited and power control is essential in the use of CDMA systems along with multi-user detection techniques. Because of the power law path loss, the transmit power may be reduced by breaking down the distance between two communication point into smaller segments. In [8], the proposed model, Cellular Augmented Ad Hoc Networks (CAHAN) demonstrates an energy-balancing scheme which balances battery energy of mobile terminals and hence extends the network lifetime that is defined as the time until the first mobile terminal runs out of its battery energy. In [8], it has been shown that total energy consumed by the mobile terminal is lower in the case of CAHAN than in the case of single hop cellular networks. It is also shown that mobility and ad hoc communication range are both factors potentially influencing the network lifetime of CAHAN.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 introduces a new transmission strategy for MCNs in order to limit number of hops. Chapter 3 describes AODV protocol and the proactive technique incorporated into AODV to emulate new transmission strategy. Chapter 4 provides simulation results and proposed framework. Conclusions are given in Chapter 5.

CHAPTER 2

FBS (FORWARD-TO-BASE STATION)

MCN combines the benefit of conventional single hop cellular network where the service infrastructure is constructed by fixed backbone which comprises of base stations and mobile switch centers, and utilizes the flexibility of ad-hoc networks where wireless transmission through mobile stations in multiple hops is allowed. Before developing a transmission strategy, the capacity for MCNs must be defined. In multi-hop wireless networks the capacity is basically defined as the total rate at which information data originated by all sources reaches its final destination. Since the destination is the base station, the capacity for MCNs can be defined as the total rate of packets originated by mobile terminals and delivered to the base station. The capacity of the MCNs is therefore restricted due to intensive traffic in the network since all the nodes has capability of sending packets simultaneously. In this thesis, a new transmission strategy is introduced Forward to Base Station (FBS), which maximizes the capacity of the network by minimizing the number of hops. The strategy is based on a delay constraint: that is the maximum acceptable delay boundary for packets in the network. Since all nodes have the capability of communicating with the base station directly (it is assumed that all nodes are in the coverage area), they must decide to send incoming packets to the either base station or to the next node in the routing table based on the time remaining until each packet must reach to destination.

Routing has a significant impact in the transmission strategy in terms of assigning the transmit power to the next hop for each node and directing the traffic in a way that will limit the total hop count. Transmission strategy FBS utilizes the routing information and satisfies maximum tolerable end-to-end delay in MCNs. This strategy significantly decreases the excess power consumption and traffic due to retransmission of the same data. Quality of Service (QoS) requirements such as maximum acceptable data rate and maximum allowable end-to-end delay constraints will affect the system capacity.

2.1 Performance Metrics

The packet forwarding scheme is based on the time remaining until the packet must reach its destination as well as the number of hops to the base station. During periods of congestion, a flow or packet end-to-end performance properties are strongly influenced by the choice of the scheduling algorithm employed at the network's routers. Employing FBS scheduling mechanism, which is a base station oriented packet forwarding strategy, can limit congestion level; therefore, the targeted delay and throughput are ensured with less number of hops.

Traditional communication applications such as file transfer and electronic mail are examples of non-real-time traffic. The performance metric for such applications is typically average end-to-end delay and throughput. On the other hand, the characteristics of real-time communication applications differ significantly. Most importantly, each real time packet is characterized by a deadline by which it must be received, otherwise, it is considered to be lost. The primary performance metric for this class of traffic is the fraction of packets received within their deadline. On the other hand, efficient routing results in smaller average packet delays, which means that the flow control algorithm can accept more traffic into the network. By combining FBS with a proper routing algorithm, the excessive offered load that would increase packet delays will be rejected. Therefore, decreased end-to-end delay will result with less number of hops which means increased network capacity.

2.2 Analytical Bound

The provision of end-to-end delay guarantees in high speed in mobile networks remains one of the most important and widely studied QoS issues. Many real time audio and video applications rely on the ability of the network to provide small delays. FBS attempts to minimize end-to-end delay forwarding flows to base station. Before introducing the scheme, first the delay bounds for the much studied Weighted Fair Queuing (WFQ) scheduling discipline also known as Packet-by-Packet Generalized Processor Sharing (PGPS) is recalled. In [5], it is shown that WFQ achieves the following session-*i* delay bound for packet switched networks.

$$\frac{\sigma_i + (K_i - 1)L_i}{\rho_i} + \sum_{m=1}^{K_i} \frac{L_{\max}}{r^m} \qquad (1)$$

For session *i*, L_i is the maximum packet size, K_i is the number of servers and r^m is the service rate of m_{th} server. The maximum packet size over all sessions is $L_{\max}$. Session *i* leaky- bucket constrained with burst size σ_i rate ρ_i.

To understand the delay guarantee in (1) better, the delay bound when session *i* has a single hop (K_i=1) is compared with the delay bound when session *i* has multiple hops (K_i>1). When the burst size σ_i is large the multiple-hop-delay bound is much less than K_i times the single-hop delay bound. However, when σ_i is small than the multiple-hop delay can be approximately K_i times the single-hop delay. To prove this, it can be assumed that a uniform packet size for all sessions (L_i=1) and a uniform service rate for all services ($r^m = 1$) the delay bound of (1) becomes

$$\frac{\sigma_i + K_i - 1}{\rho_i} + K_i$$

Hence, for a small burst size, e.g. $\sigma_i = 1$, the multiple-hop delay is

$$\frac{1}{\rho_i} \times K_i$$

and the single hop delay is

$$\frac{1}{\rho_i}$$

By introducing simple coordination among the nodes in the routing algorithm it can be ensured that a packet can arrive at its destination in a given time interval with less number of hops.

CHAPTER 3

ROUTING

Conventional routing protocols developed for wired LANs/WANs can be used for routing in MCN, treating each mobile host as a router. Such algorithms broadly come under the category of pro-active algorithms since routing information is disseminated among all nodes in the network at all times irrespective of the need for any such route. Since channel bandwidth is scarce, many researchers have proposed on-demand routing algorithms. On-demand routing algorithms build or maintain only the routing paths that have changed and are needed to send the data packets currently in the network. Many performance comparisons done until now have shown that on demand algorithms perform better than pro-active routing algorithms and are better suited for mobile environments [15, 16]. On-demand routing algorithms create routes only when desired by source nodes. When a node requires a route to a destination, it initiates a route construction procedure. Route maintenance procedure is triggered whenever a route has been constructed. The procedure is in progress until any node in the route is unreachable or the route is no longer required. The control messages used in on-demand routing networks only record the desirable data on the route such as nodes on the route and other performance metrics and so forth. On-demand routing protocols greatly reduce the size of control message as compared with pro-active routing algorithms and can withstand the increasing number of nodes.

QoS issues in MCN are more complicated than that in conventional cellular networks. The guarantee of QoS in conventional cellular network requires single hop, however, MCN requires hop-to-hop guarantee of each node from source to the destination. Therefore, to guarantee the delay constraints of routing in MCN, each node in the route is required to propagate congestion information within the network and reserve an affordable amount of bandwidth.

Two widely studied on-demand routing algorithms are Ad Hoc On-Demand Distance Vector (AODV) routing protocol and the Dynamic Source Routing Protocol (DSR). Recent studies show that under a wide variety of scenarios, AODV performs much better than DSR in terms of latency, throughput, and even routing overhead (in bits/s). In absolute terms; however, AODV has very high latencies, over 100 ms on the average, especially when the relative speeds of nodes is high and the network load is low. By using FBS, it is shown that packet latencies can be reduced in such conditions in AODV, using a simple pro-active technique that will decrease the number of hops. The technique involves broadcasting receiver-initiated 'beacon' packets periodically which help maintain fresh routes to corresponding receivers at all the nodes along with congestion information.

3.1 AODV

AODV algorithm [9] enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. AODV allows mobile nodes to respond to link breakages and changes in network topology in a timely manner. The operation of AODV is loop-free, and by avoiding the Bellman-Ford “counting to infinity” problem offers quick convergence when the ad hoc network topology changes (typically, when a node moves in the network). When links break, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link.

One distinguishing feature of AODV is its use of a destination sequence number for each route entry. The destination sequence number is created by the destination to be included along with any route information it sends to requesting nodes. Using destination sequence numbers ensures loop freedom and is simple to program. Given the choice between two routes to a destination, a requesting node is required to select the one with the greatest sequence number.

The AODV routing protocol is designed for mobile ad hoc networks with populations of tens to thousands of mobile nodes. AODV can handle low, moderate, and relatively high mobility rates, as well as a variety of data traffic levels. AODV is designed for use in networks where the nodes can all trust each other, either by use of pre-configured keys, or because it is known that there are no malicious intruder nodes. AODV has been designed to reduce the dissemination of control traffic and eliminate overhead on data traffic, in order to improve scalability and performance.

3.1.1 Route Table Entries and Precursor Lists

When a node receives an AODV control packet from a neighbor, or creates or updates a route for a particular destination or subnet, it checks its route table for an entry for the destination. In the event that there is no corresponding entry for that destination, an entry is created. The sequence number is either determined from the information contained in the control packet, or else the valid sequence number field is set to false. The route is only updated if the new sequence number is either (i) higher than the destination sequence number in the route table, or (ii) the sequence numbers are equal, but the hop count (of the new information) plus one, is smaller than the existing hop count in the routing table, or (iii) the sequence number is unknown. The Lifetime field of the routing table entry is either determined from the control packet, or it is initialized to ACTIVE_ROUTE_TIMEOUT. This route may now be used to send any queued data packets and fulfills any outstanding route requests.

Each time a route is used to forward a data packet, its Active Route Lifetime field of the source, destination and the next hop on the path to the destination is updated to be no less than the current time plus ACTIVE_ROUTE_TIMEOUT. Since the route between each originator and destination pair is expected to be symmetric, the Active Route Lifetime for the previous hop, along the reverse path back to the IP source, is also updated to be no less than the current time plus ACTIVE_ROUTE_TIMEOUT. The lifetime for an Active Route is updated each time the route is used regardless of whether the destination is a single node or a subnet. For each valid route maintained by a node as a routing table entry, the node also maintains a list of precursors that may be forwarding packets on this route. These precursors will receive notifications from the node in the event of detection of the loss of the next hop link. The list of precursors in a routing table entry contains those neighboring nodes to which a route reply was generated or forwarded.

3.1.2 Route Discovery

A node disseminates a RREQ when it determines that it needs a route to a destination and does not have one available. This can happen if the destination is previously unknown to the node or if a previously valid route to the destination expires or is marked as invalid. The Destination Sequence Number field in the RREQ message is the last known destination sequence number for this destination and is copied from the Destination Sequence Number field in the routing table. If no sequence number is known, the unknown sequence number flag must be set. The Originator Sequence Number in the RREQ message is the node's own sequence number, which is incremented prior to insertion in a RREQ. The RREQ ID field is incremented by one from the last RREQ ID used by the current node. Each node maintains only one RREQ ID. The Hop Count field is set to zero. Before broadcasting the RREQ, the originating node buffers the RREQ ID and the Originator IP address (its own address) of the RREQ for PATH_DISCOVERY_TIME. In this way, when the node receives the packet again from its neighbors, it will not reprocess and re-forward the packet.

An originating node often expects to have bidirectional communications with a destination node. In such cases, it is not sufficient for the originating node to have a route to the destination node; the destination must also have a route back to the originating node. In order for this to happen as efficiently as possible, any generation of a RREP by an intermediate node for delivery to the originating node should be accompanied by some action that notifies the destination about a route back to the originating node. The originating node selects this mode of operation in the intermediate nodes by setting the ‘G’ flag. A node should not originate more than

RREQ_RATELIMIT RREQ messages per second. After broadcasting a RREQ, a node waits for a RREP (or other control message with current information regarding a route to the appropriate destination). If a route is not received within NET_TRAVERSAL_TIME milliseconds, the node may try again to discover a route by broadcasting another RREQ, up to a maximum of RREQ_RETRIES times at the maximum TTL value. Each new attempt must increment and update the RREQ ID. Data packets waiting for a route (i.e., waiting for a RREP after a RREQ has been sent) should be buffered. The buffering should be "first-in, first-out" (FIFO). If a route discovery has been attempted RREQ_RETRIES times at the maximum TTL without receiving any RREP, all data packets destined for the corresponding destination should be dropped from the buffer and a Destination Unreachable message should be delivered to the application. To reduce congestion in a network, repeated attempts by a source node at route discovery for a single destination must utilize a binary exponential backoff. The first time a source node broadcasts a RREQ; it waits NET_TRAVERSAL_TIME milliseconds for the reception of a RREP. If a RREP is not received within that time, the source node sends a new RREQ. When calculating the time to wait for the RREP after sending the second RREQ, the source node must use a binary exponential backoff. Hence, the waiting time for the RREP corresponding to the second RREQ is 2 * NET_TRAVERSAL_TIME milliseconds. If a RREP is not received within this time period, another RREQ may be sent, up to RREQ_RETRIES additional attempts after the first RREQ. For each additional attempt, the waiting time for the RREP is multiplied by 2, so that the time conforms to a binary exponential backoff.

3.1.3 Controlling Dissemination of Route Request Messages

To prevent unnecessary network-wide dissemination of RREQs, the originating node should use an expanding ring search technique. In an expanding ring search, the originating node initially uses a TTL = TTL_START in the RREQ packet IP header and sets the timeout for receiving a RREP to RING_TRAVERSAL_TIME milliseconds. RING_TRAVERSAL_TIME is calculated as described in section 10. The TTL_VALUE used in calculating RING_TRAVERSAL_TIME is set equal to the value of the TTL field in the IP header. If the RREQ times out without a corresponding RREP, the originator broadcasts the RREQ again with the TTL incremented by TTL_INCREMENT. This continues until the TTL set in the RREQ reaches TTL_THRESHOLD, beyond which a TTL = NET_DIAMETER is used for each attempt. Each time, the timeout for receiving a RREP is RING_TRAVERSAL_TIME. When it is desired to have all retries traverse the entire ad hoc network, this can be achieved by configuring TTL_START and TTL_INCREMENT both to be the same value as NET_DIAMETER. The Hop Count stored in an invalid routing table entry indicates the last known hop count to that destination in the routing table. When a new route to the same destination is required at a later time (e.g., upon route loss), the TTL in the RREQ IP header is initially set to the Hop Count plus TTL_INCREMENT. Thereafter, following each timeout the TTL is incremented by TTL_INCREMENT until TTL = TTL_THRESHOLD is reached. Beyond this TTL = NET_DIAMETER is used. Once TTL = NET_DIAMETER, the timeout for waiting for the RREP is set to NET_TRAVERSAL_TIME. An expired routing table entry should not be expunged before (current_time + DELETE_PERIOD). Otherwise, the soft state corresponding to the route (e.g., last known hop count) will be lost. Furthermore, a longer routing table entry expunge time MAY be configured. Any routing table entry waiting for a RREP should not be expunged before (current_time + 2 * NET_TRAVERSAL_TIME).

3.1.4 Processing and Forwarding Route Requests

When a node receives a RREQ, it first creates or updates a route to the previous hop without a valid sequence number then checks to determine whether it has received a RREQ with the same Originator IP Address and RREQ ID within at least the last PATH_DISCOVERY_TIME. If such a RREQ has been received, the node silently discards the newly received RREQ. The rest of this subsection describes actions taken for RREQs that are not discarded. First, it first increments the hop count value in the RREQ by one, to account for the new hop through the intermediate node. Then the node searches for a reverse route to the Originator IP Address using longest-prefix matching. If need be, the route is created, or updated using the Originator Sequence Number from the RREQ in its routing table. This reverse route will be needed if the node receives a RREP back to the node that originated the RREQ (identified by the Originator IP Address). When the reverse route is created or updated, the following actions on the route are also carried out:

1. The Originator Sequence Number from the RREQ is compared to the corresponding destination sequence number in the route table entry and copied if greater than the existing value there
2. The valid sequence number field is set to true;

3. The next hop in the routing table becomes the node from which the RREQ was received (it is obtained from the source IP address in the IP header and is often not equal to the Originator IP Address field in the RREQ message);
4. The hop count is copied from the Hop Count in the RREQ message;

Whenever a RREQ message is received, the Lifetime of the reverse route entry for the Originator IP address is set to be the maximum of (ExistingLifetime, MinimalLifetime), where MinimalLifetime = (current time + 2*NET_TRAVERSAL_TIME - 2*HopCount*NODE_TRAVERSAL_TIME). The current node can use the reverse route to forward data packets in the same way as for any other route in the routing table. If a node does not generate a RREP and if the incoming IP header has TTL larger than 1, the node updates and broadcasts the RREQ to address 255.255.255.255 on each of its configured interfaces. To update the RREQ, the TTL or hop limit field in the outgoing IP header is decreased by one, and the Hop Count field in the RREQ message is incremented by one, to account for the new hop through the intermediate node. Lastly, the Destination Sequence number for the requested destination is set to the maximum of the corresponding value received in the RREQ message, and the destination sequence value currently maintained by the node for the requested destination. However, the forwarding node must not modify its maintained value for the destination sequence number, even if the value received in the incoming RREQ is larger than the value currently maintained by the forwarding node. Otherwise, if a node does generate a RREP, then the node discards the RREQ. Notice that, if intermediate nodes reply to every transmission of RREQs for a particular destination, it might turn out that the destination does not receive any of the discovery messages. In this situation, the destination does not learn of a route to the originating node from the RREQ messages. This could cause the destination to initiate a route discovery (for example, if the originator is attempting to establish a TCP session). In order that the destination learn of routes to the originating node, the originating node should set the "gratuitous RREP" (G) flag in the RREQ if for any reason the destination is likely to need a route to the originating node. If, in response to a RREQ with the 'G' flag set, an intermediate node returns a RREP, it MUST also unicast a gratuitous RREP to the destination node.

3.1.5 Generating Route Replies

A node generates a RREP if either: (i) it is itself the destination, or (ii) it has an active route to the destination, the destination sequence number in the node's existing route table entry for the destination is valid and greater than or equal to the Destination Sequence Number of the RREQ (comparison using signed 32-bit arithmetic), and the "destination only" (D) flag is NOT set. When generating a RREP message, a node copies the Destination IP Address and the Originator Sequence Number from the RREQ message into the corresponding fields in the RREP message. Processing is slightly different, depending on whether the node is itself the requested destination, or instead if it is an intermediate node with an fresh enough route to the destination Once created, the RREP is unicast to the next hop toward the originator of the RREQ, as indicated by the route table entry for that originator. As the RREP is forwarded back towards the node, which originated the RREQ message, the Hop Count field is incremented by one at each hop. Thus, when the RREP reaches the originator, the Hop Count represents the distance, in hops, of the destination from the originator.

3.1.6 Route Reply Generation by the Destination

If the generating node is the destination itself, it must increment its own sequence number by one if the sequence number in the RREQ packet is equal to that incremented value. Otherwise, the destination does not change its sequence number before generating the RREP message. The destination node places its (perhaps newly incremented) sequence number into the Destination Sequence Number field of the RREP, and enters the value zero in the Hop Count field of the RREP. The destination node copies the value MY_ROUTE_TIMEOUT into the Lifetime field of the RREP. Each node may reconfigure its value for MY_ROUTE_TIMEOUT, within mild constraints.

3.1.7 Route Reply Generation by an Intermediate Node

If the node generating the RREP is not the destination node, but instead is an intermediate hop along the path from the originator to the destination, it copies its known sequence number for the destination into the Destination Sequence Number field in the RREP message. The intermediate node updates the forward route entry by placing the last hop node (from which it received the RREQ, as indicated by the source IP address field in the IP header) into the precursor list for the forward route entry i.e. the entry for the Destination IP Address. The intermediate node also updates its route table entry for the node originating the RREQ by placing the next hop towards the destination in the precursor list for the reverse route entry i.e. the entry for the Originator IP Address field of the RREQ message data. The intermediate node places its distance in hops from the destination (indicated by the hop count in the routing table) Count field in the RREP. The

Lifetime field of the RREP is calculated by subtracting the current time from the expiration time in its route table entry.

3.1.8 Receiving and Forwarding Route Replies

When a node receives a RREP message, it searches (using longest-prefix matching) for a route to the previous hop. If needed, a route is created for the previous hop, but without a valid sequence number. Next, the node then increments the hop counts value in the RREP by one, to account for the new hop through the intermediate node. Call this incremented value the "New Hop Count". Then the forward route for this destination is created if it does not already exist. Otherwise, the node compares the Destination Sequence Number in the message with its own stored destination sequence number for the Destination IP Address in the RREP message. Upon comparison, the existing entry is updated only in the following circumstances: (i) the sequence number in the routing table is marked as invalid in route table entry (ii) The Destination Sequence Number in the RREP is greater than the node's copy of the destination sequence number and the known value is valid, or (iii) the sequence numbers are the same, but the route is is marked as inactive, or (iv) the sequence numbers are the same, and the New Hop Count is smaller than the hop count in route table entry. If the route table entry to the destination is created or updated, then the following actions occur:

1. The route is marked as active,

2. The destination sequence number is marked as valid,

3. The next hop in the route entry is assigned to be the node from which the RREP is received, which is indicated by the source IP address field in the IP header,

4. The hop count is set to the value of the New Hop Count,

5. The expiry time is set to the current time plus the value of the Lifetime in the RREP message,

6. And the destination sequence number is the Destination Sequence Number in the RREP message.

The current node can subsequently use this route to forward data packets to the destination. If the current node is not the node indicated by the Originator IP Address in the RREP message AND a forward route has been created or updated as described above, the node consults its route table entry for the originating node to determine the next hop for the RREP packet, and then forwards the RREP towards the originator using the information in that route table entry. If a node forwards a RREP over a link that is likely to have errors or be unidirectional, the node should set the 'A' flag to require that the recipient of the RREP acknowledge receipt of the RREP by sending a RREP-ACK message back. When any node transmits a RREP, the precursor list for the corresponding destination node is updated by adding to it the next hop node to which the RREP is forwarded. Also, at each node the (reverse) route used to forward a RREP has its lifetime changed to be the maximum of (existing-lifetime, (current time + ACTIVE_ROUTE_TIMEOUT)). Finally, the precursor list for the next hop towards the destination is updated to contain the next hop towards the source.

3.1.9 Operation over Unidirectional Links

It is possible that a RREP transmission may fail, especially if the RREQ transmission triggering the RREP occurs over a unidirectional link. If no other RREP generated from the same route discovery attempt reaches the node, which originated the RREQ message, the originator will reattempt route discovery after a timeout. However, the same scenario might well be repeated without any improvement, and no route would be discovered even after repeated retries. Unless corrective action is taken, this can happen even when bidirectional routes between originator and destination do exist. Link layers using broadcast transmissions for the RREQ will not be able to detect the presence of such unidirectional links.

In AODV, any node acts on only the first RREQ with the same RREQ ID and ignores any subsequent RREQs. Suppose, for example, that the first RREQ arrives along a path that has one or more unidirectional link(s). A subsequent RREQ may arrive via a bidirectional path (assuming such paths exist), but it will be ignored. To prevent this problem, when a node detects that its transmission of a RREP message has failed, it remembers the next-hop of the failed RREP in a "blacklist" set. Such failures can be detected via the absence of a link-layer or network-layer acknowledgment (e.g., RREP-ACK). A node ignores all RREQs received from any node in its blacklist set. Nodes are removed from the blacklist set after a BLACKLIST_TIMEOUT period. Note that the RREP-ACK packet does not contain any information about which RREP it is acknowledging. The time at which the RREP-ACK is received will likely come just after the time when the RREP was sent with the 'A' bit. This information is expected to be sufficient to provide assurance to the sender of the RREP that the link is currently bidirectional, without any real dependence on the particular RREP message being acknowledged. However, that assurance typically cannot be expected to remain in force permanently.

3.1.10 Hello Messages

A node may offer connectivity information by broadcasting local Hello messages. A node should only use hello messages if it is part of an active route. Every HELLO_INTERVAL milliseconds, the node checks whether it has sent a broadcast (e.g., a RREQ or an appropriate layer 2 message) within the last HELLO_INTERVAL. If it has not, it may broadcast a RREP with TTL = 1, called a Hello message, with the RREP message fields set as follows: (Destination IP Address) The node's IP address. (Destination Sequence Number) The node's latest sequence number. (Hop Count) 0. (Lifetime) ALLOWED_HELLO_LOSS * HELLO_INTERVAL.

A node may determine connectivity by listening for packets from its set of neighbors. If, within the past DELETE_PERIOD, it has received a Hello message from a neighbor, and then for that neighbor does not receive any packets (Hello messages or otherwise) for more than ALLOWED_HELLO_LOSS * HELLO_INTERVAL milliseconds, the node should assume that the link to this neighbor is currently lost. Whenever a node receives a Hello message from a neighbor, the node should make sure that it has an active route to the neighbor, and create one if necessary. If a route already exists, then the Lifetime for the route should be increased, if necessary, to be at least ALLOWED_HELLO_LOSS * HELLO_INTERVAL. The route to the neighbor, if it exists, MUST subsequently contain the latest Destination Sequence Number from the

Hello message. The current node can now begin using this route to forward data packets. Routes that are created by hello messages and not used by any other active routes will have empty precursor lists and would not trigger a RERR message if the neighbor moves away and a neighbor timeout occurs.

3.1.11 Maintaining Local Connectivity

Each forwarding node should keep track of its continued connectivity to its active next hops (i.e. which next hops or precursors have forwarded packets to or from the forwarding node during the last ACTIVE_ROUTE_TIMEOUT), as well as neighbors that have transmitted Hello messages during the last (ALLOWED_HELLO_LOSS * HELLO_INTERVAL). A node can maintain accurate information about its continued connectivity to these active next hops, using one or more of the available link or network layer mechanisms, as described below.

1. Any suitable link layer notification, such as those provided by IEEE 802.11, can be used to determine connectivity, each time a packet is transmitted to an active next hop. For example, absence of a link layer ACK or failure to get a CTS after sending RTS, even after the maximum number of retransmission attempts, indicates loss of the link to this active next hop.

2. If layer-2 notification is not available, passive acknowledgment should be used when the next hop is expected to forward the packet, by listening to the channel for a transmission attempt made by the next hop. If transmission is not detected within NEXT_HOP_WAIT milliseconds or the next hop is the destination (and

thus is not supposed to forward the packet) one of the following methods should be used to determine connectivity:

* Receiving any packet (including a Hello message) from the next hop.

* A RREQ unicast to the next hop, asking for a route to the next hop.

* An ICMP Echo Request message unicast to the next hop.

If a link to the next hop cannot be detected by any of these methods, the forwarding node should assume that the link is lost, and take corrective action by following the methods specified in next section.

3.1.12 Route Error (RERR) Messages, Route Expiry and Route Deletion

Generally, route error and link breakage processing requires the following steps:

1. Invalidating existing routes
2. Listing affected destinations
3. Determining which, if any, neighbors may be affected
4. Delivering an appropriate RERR to such neighbors

A Route Error (RERR) message may be broadcast (if there are many precursors), unicast (if there is only 1 precursor), or iteratively unicast to all precursors (if broadcast is inappropriate). Even when the RERR message is iteratively unicast to several precursors, it is considered to be a single control message for the purposes of the description in the text that follows. With that understanding, a node should not generate more than RERR_RATELIMIT RERR messages per second. A node initiates processing for a RERR message in three situations: (i) if it detects a link break for the next hop of an active route in its routing table while transmitting data (and route repair, if attempted, was unsuccessful), or (ii) if it gets a data packet destined to a node for which it does not have an active route and is not repairing (if using local repair), or (iii) if it receives a RERR from a neighbor for one or more active routes.

For case (i), the node first makes a list of unreachable destinations consisting of the unreachable neighbor and any additional destinations in the local routing table that use the unreachable neighbor as the next hop. In this case, if a subnet route is found to be newly unreachable, an IP destination address for the subnet is constructed by appending zeroes to the subnet prefix as shown in the route table entry. This is unambiguous, since the precursor is known to have route table information with a compatible prefix length for that subnet. For case (ii), there is only one unreachable destination, which is the destination of the data packet that cannot be delivered. For case (iii), the list should consist of those destinations in the RERR for which there exists a corresponding entry in the local routing table that has the transmitter of the received RERR as the next hop. Some of the unreachable destinations in the list could be used by neighboring nodes, and it may therefore be necessary to send a (new) RERR. The RERR should contain those destinations that are part of the created list of unreachable destinations and have a non-empty precursor list. The neighboring node(s) that should receive the RERR are all those that belong to a precursor list of at least one of the unreachable destination(s) in the newly created RERR. In case there is only one unique neighbor that needs to receive the RERR, the RERR should be unicast toward that neighbor. Otherwise the RERR is typically sent to the local broadcast address (Destination IP == 255.255.255.255, TTL == 1) with the unreachable destinations, and their corresponding destination sequence numbers, included in the packet. The DestCount field of the RERR packet indicates the number of unreachable destinations included in the packet. Just before transmitting the RERR, certain updates are made on the routing table that may affect the destination sequence numbers for the unreachable destinations. For each one of these destinations, the corresponding routing table entry is updated as follows:

1. The destination sequence number of this routing entry, if it exists and is valid, is incremented for cases (i) and (ii) above, and copied from the incoming RERR in case (iii) above.
2. The entry is invalidated by marking the route entry as invalid
3. The Lifetime field is updated to current time plus DELETE_PERIOD. Before this time, the entry should not be deleted.

3.1.13 Local Repair

When a link break in an active route occurs, the node upstream of that break may choose to repair the link locally if the destination was no farther than MAX_REPAIR_TTL hops away. To repair the link break, the node increments the sequence number for the destination and then broadcasts a RREQ for that destination. The TTL of the RREQ should initially be set to the following value:

max(MIN_REPAIR_TTL, 0.5 * #hops) + LOCAL_ADD_TTL

Where #hops is the number of hops to the sender (originator) of the currently undeliverable packet. Thus, local repair attempts will often be invisible to the originating node, and will always have TTL >= MIN_REPAIR_TTL + LOCAL_ADD_TTL. The node initiating the repair then waits the discovery period to receive RREPs in response to the RREQ. During local repair data packets should be buffered. If, at the end of the discovery period, the repairing node has not received a RREP (or other control message creating or updating the route) for that destination, it proceeds as described in Section 6.11 by transmitting a RERR message for that destination. On the other hand, if the node receives one or more RREPs (or other control message creating or updating the oute to the desired destination) during the discovery period, it first compares the hop count of the new route with the value in the hop count field of the invalid route table entry for that destination. If the hop count of the newly determined route to the destination is greater than the hop count of the previously known route the node should issue a RERR message for the destination, with the 'N' bit set. Then it proceeds as described in Section 6.7, updating its route table entry for that destination. A node that receives a RERR message with the 'N' flag set must not delete the route to that destination. The only action taken should be the retransmission of the message, if the RERR arrived from the next hop along that route, and if there are one or more precursor nodes for that route to the destination. When the originating node receives a RERR message with the 'N' flag set, if this message came from its next hop along its route to the destination then the originating node may choose to reinitiate route discovery.

Local repair of link breaks in routes sometimes results in increased path lengths to those destinations. Repairing the link locally is likely to increase the number of data packets that are able to be delivered to the destinations, since data packets will not be dropped as the RERR travels to the originating node. Sending a RERR to the originating node after locally repairing the link break may allow the originator to find a fresh route to the destination that is better, based on current node positions. However, it does not require the originating node to rebuild the route, as the originator may be done, or nearly done, with the data session. When a link breaks along an active route, there are often multiple destinations that become unreachable. The node that is upstream of the lost link tries an immediate local repair for only the one destination towards which the data packet was traveling. Other routes using the same link must be marked as invalid, but the node handling the local repair may flag each such newly lost route as locally repairable; this local repair flag in the route table must be reset when the route times out (e.g., after the route has been not been active for ACTIVE_ROUTE_TIMEOUT). Before the timeout occurs, these other routes will be repaired as needed when packets arrive for the other destinations. Hence, these routes are repaired as needed; if a data packet does not arrive for the route, then that route will not be repaired. Alternatively, depending upon local congestion, the node may begin the process of establishing local repairs for the other routes, without waiting for new packets to arrive. By proactively repairing the routes that have broken due to the loss of the link, incoming data packets for those routes will not be subject to the delay of repairing the route and can be immediately forwarded. However, repairing the route before a data packet is received for it runs the risk of repairing routes that are no longer in use. Therefore, depending upon the local traffic in the network and whether congestion is being experienced, the node may elect to proactively repair the routes before a data packet is received; otherwise, it can wait until a data is received, and then commence the repair of the route.

3.2 Modifying AODV for FBS

In order to emulate FBS transmission strategy, the nodes periodically, i.e. 1 second or less intervals, send beacons that are broadcasted through out the network. A beacon entry has the receiver's sequence number, IP address, broadcast ID, the hop count, and the "B" flag, which indicates the buffer information whether it is empty or full. The hop count is incremented at each of the forwarding nodes. If the received sequence number or better hop count with the same sequence number is higher than the one in the routing table then the routing table is updated. Additionally, AODV routing table entries will have an extra field for "B" flag (B=0: buffer is empty, B=1: buffer is full). If a beacon entry has a different "B" flag, then the node that receives the beacon will change its entry. In this case (Figure 3.2), if the received beacon has "B" flag 1, which means that next node's buffer is not empty, then the node will set its next hop field to base station's IP address (It is assumed that all nodes are in coverage are of the base station). Also, if the beacon entry does not result in adding or updating the corresponding entry in nodes routing table, then the node will not propagate it further. Thus beacons will disseminate buffer information for each node and refresh the routing entries for receivers in other node's routing tables, even before they are expired.

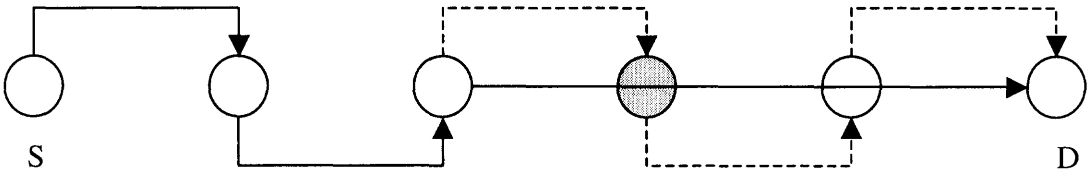

Figure 3.2 Congestion oriented AODV.

Consequently, the data packets will have fresher routes to its destinations available readily without the need for a route discovery process. Furthermore, congestion controlled routing scheme will provide the packets with less latencies and less number of hops. On the other hand, there is a potential disadvantage of this technique is the increase in the number of routing packets due to beacon packets. In order to mitigate this effect, the broadcast type packets route request (RREQ) and route reply (RREP) can be utilized. Beacons can be allowed to be piggybacked on RREQ and RREP, since they are both broadcasted through out the network.

CHAPTER 4

SIMULATION MODEL AND RESULTS

The performance of FBS based on routing protocol is studied with simulation. The new QoS routing protocol has been implemented with OPNET 7. The implementation is based on the AODV module contributed by National Institute of Technology (NIST) and additional functions presented at Chapter 3.2 are added.

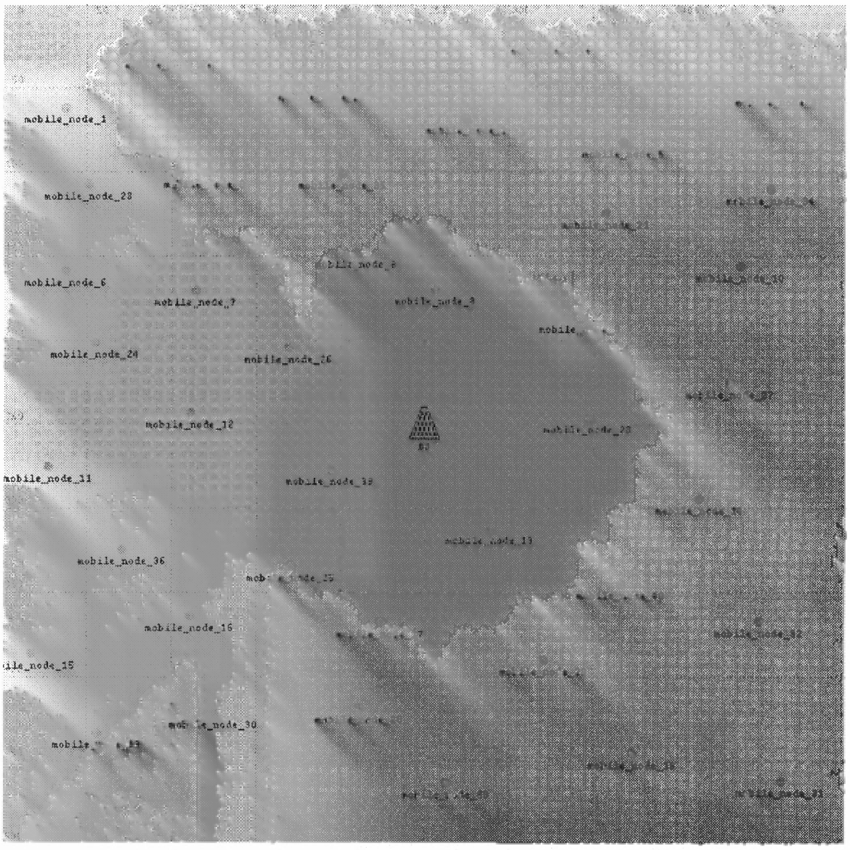

Figure 4.1 Simulation network model.

The simulated network (Figure 4.1) has 40 mobile users randomly placed, initially on a 500X500 meters field. The wireless channel has 2 Mbps bandwidth and radio transmission range is 275 meters. The mobility model is random waypoint scheme, where nodes move randomly around the specified area. Each node moves with a speed limit between 0 and 20 m/s in small step intervals and updates its position at every step time period. The traffic is simulated with Poisson model where 10 connections are allowed at the same time. The packet interarrival time is exponentially distributed with mean 4 seconds and packet sizes are distributed with mean 1024 bits. The beacon packets interarrival time is 1 second and simulation time is 1 hour.

4.1 Performance Analysis

The average end-to-end delay (the time it takes for a data to reach its destination from the time it is generated at the source) is used, which includes all the queuing and protocol processing delays in addition to propagation and transmission delays. Also the network throughput is given (total number of data bits delivered) in Kb/s and the packet delivery rate which gives the ratio of number of packets delivered at the destinations to the number of packets originated by the sources. In order to present the difference between hop-limited routing, which emulates FBS strategy, and hop-unlimited routing enhanced AODV is denoted by 'HL-AODV'.

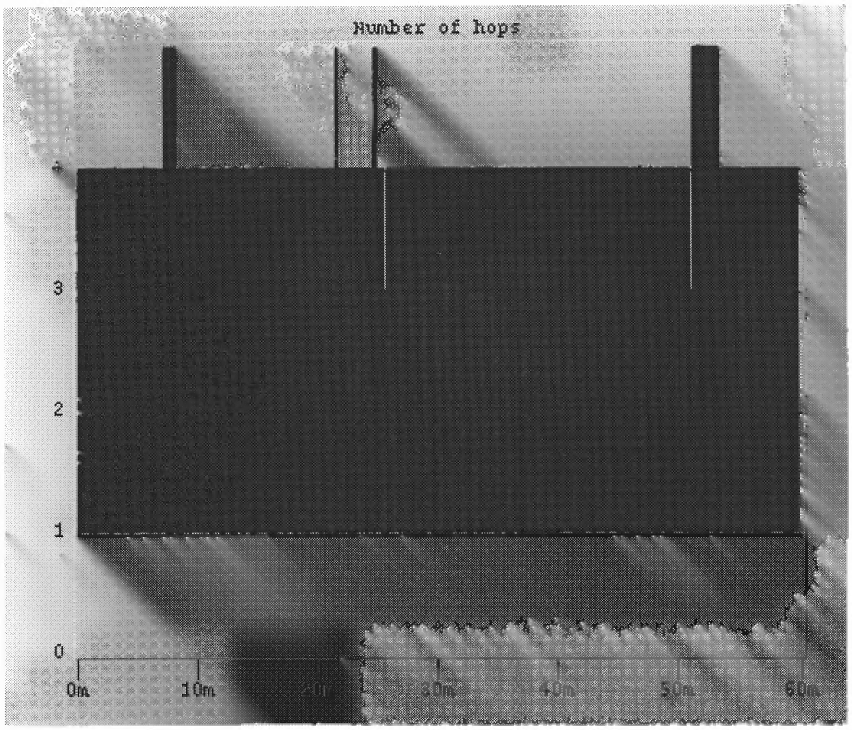

Figure 4.2 Number of hops for AODV.

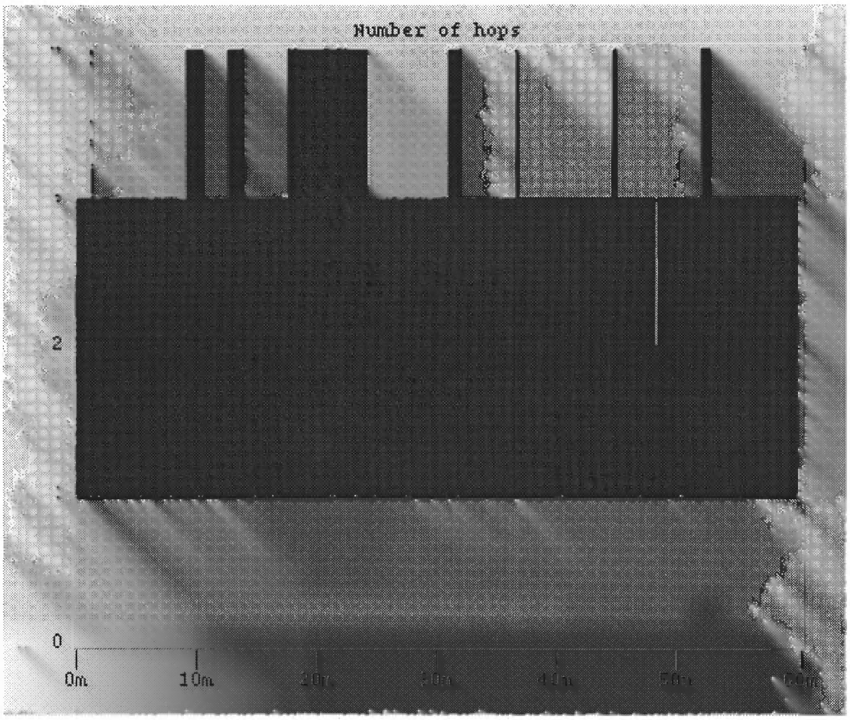

Figure 4.3 Number of hops for HL-AODV.

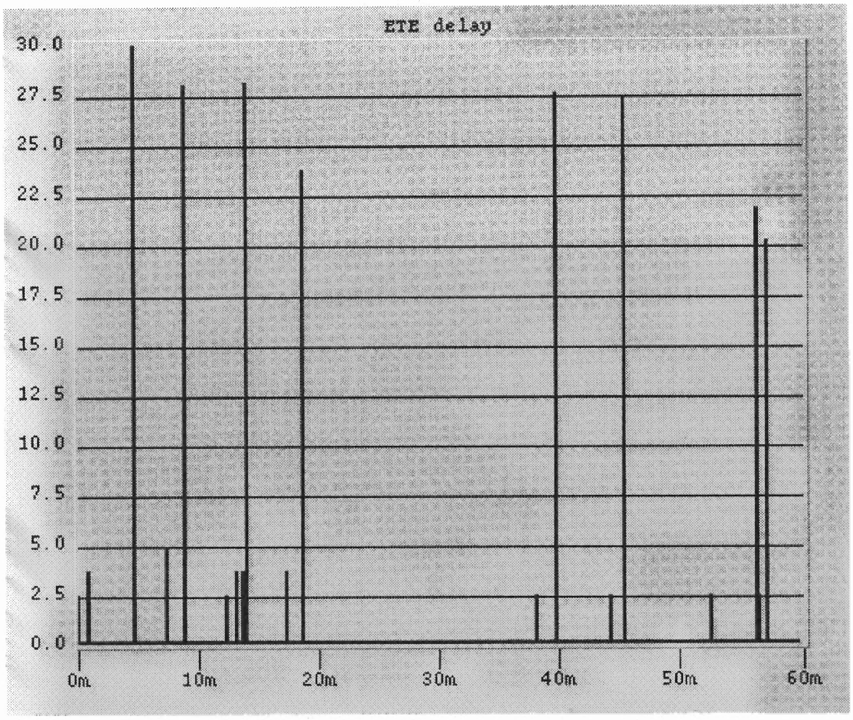

Figure 4.3 End-to-end delay for AODV.

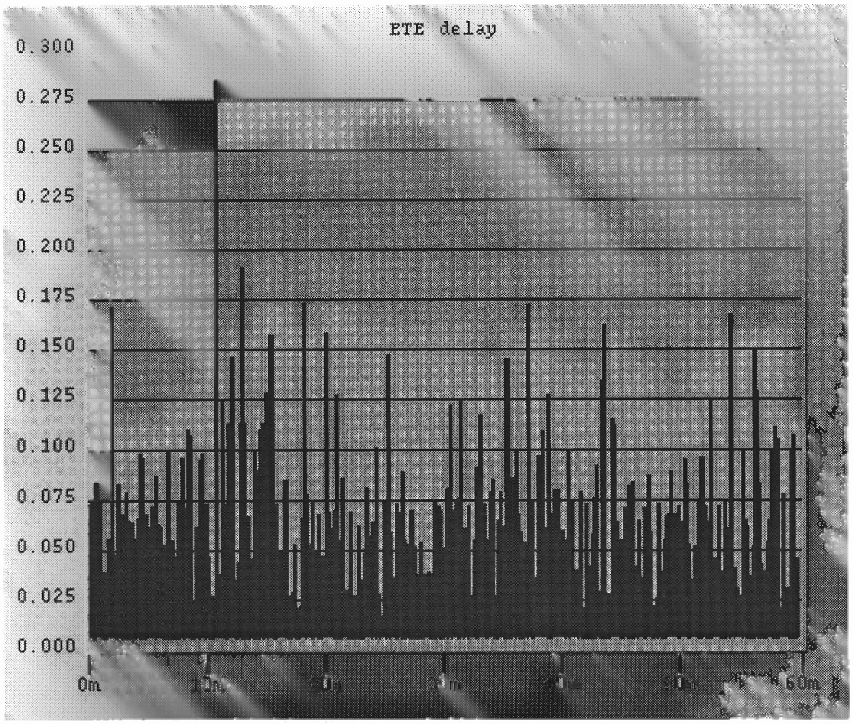

Figure 4.4 End-to-end delay for HL-AODV.

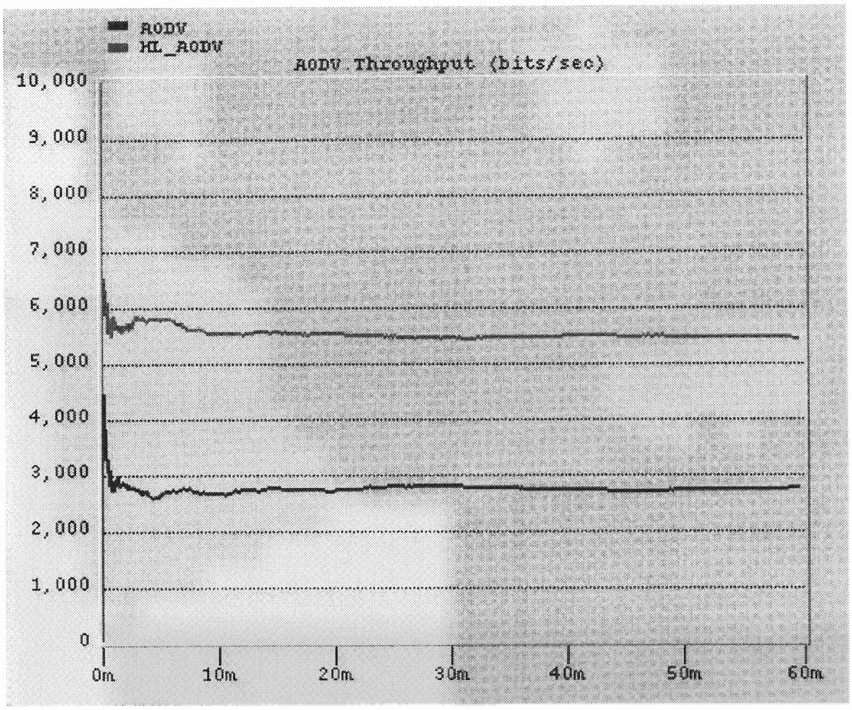

Figure 4.5 Throughput.

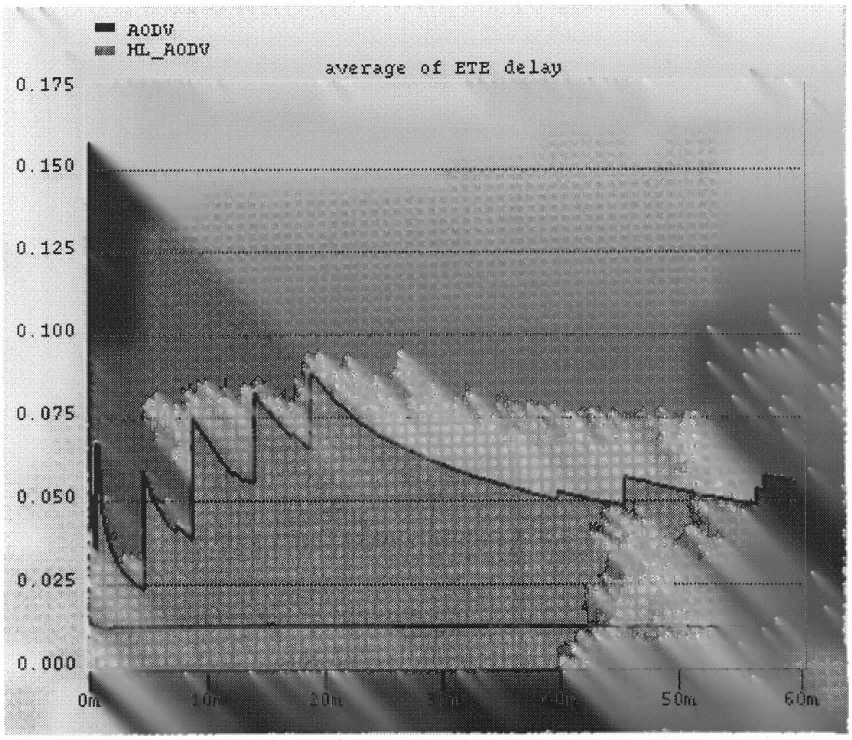

Figure 4.6 Average end-to-end delay.

In AODV, number of hops is ranged between 1 and 5 (Figure 4.2), where in HL-AODV it is ranged between 1 and 4 (Figure 4.3). The distribution of hop numbers is given in Table 4.1.

Table 4.1 Number of Hops

# of hops	n=1	n=2	n=3	n=4	n=5
AODV	3318	8118	3402	1953	92
HL-AODV	4216	6813	3101	2703	0

From Figure 4.4 it is observed that by applying FBS strategy the throughput can be improved by as much as 90%. This is because of the limited number of hops where the excess traffic due to simultaneous transmission and hence probability of error is decreased. It is also observed that although HL-AODV gives higher delivery rates (thereby higher throughputs), the network is saturated in both algorithms.

Figure 4.5 shows that average end-to-end delay decreases by HL-AODV, which forwards the incoming packets to the base station instead of next node, whose buffer is full. This because of congestion controlled transmission. Figure 4.2 and Figure 4.3 show the effect of hop limited routing: in AODV packet delays may reach 30 seconds where it is under 0.30 seconds in HL-AODV. A packet created by the source to be delivered by base station suffers from a higher amount of queuing delay. Under HL-AODV, packets are delivered with less number of hops and therefore, they are less likely to queue in the buffers. Average end-to-end delay is at a stable level of 1 ms during the simulation in HL-AODV case where in AODV case it reaches a stable level after 40 minutes.

CHAPTER 5

CONCLUSION

Previously proposed models have proved that MCN performs better than conventional cellular networks. It is addressed that there are many issues to be studied in order to standardize architecture of MCNs. In this thesis, a transmission strategy is presented to improve the capacity in terms of end-to-end delay and throughput. Simulation results have shown that by applying a proactive technique on AODV, a better throughput and lower delay can be achieved. It is observed that excess traffic due to retransmission of same data can be absorbed by base station, which results with a less network load. It is also expected that QoS issues for MCN will be more complicated than conventional cellular networks; therefore, efficient routing algorithms are required.

REFERENCES

[1] L. Ying-Dar and H. Yu-Ching, "Multihop Cellular: a new architecture for wireless communications," IEEE INFOCOM 2000.

[2] M. Souryal, B. Vojcic, R. Pickholtz, "Ad Hoc, Mutihop CDMA Networks with Route Diversity in a Rayleigh Fading Channel," IEEE MILCOM 2001.

[3] A. N. Zadeh and B. Jabbari, "A High Capacity Multihop Packet CDMA Wireless Network," ACM 2001.

[4] A. K. Parekh, R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case," IEEE/ACM TRANSACTIONS 1993.

[5] A. K. Parekh, Robert G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case," IEEE/ACM TRANSACTIONS 1994.

[6] A. Fujiwara, S. Takeda, H. Yoshino, T. Otsu, "Area Coverage and Capacity Enhancement by Multihop Connection of CDMA Cellular Network," Vehicular Technology Conference, 2002, Volume 4: pp 2371-2374 .

[7] S. De, O. Tonguz, H. Wu, C. Qiao, "Integrated Cellular and Ad Hoc Relay (iCAR) Systems: pushing the performance limits of conventional wireless networks," System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference, 2002 IEEE.

[8] C. Chen, S. Tekinay, C. Saraydar, "Cellular Ad Hoc Augmented Networks," submitted to Kluwer Academic, Mobile Networks and Applications, 2003.

[9] C. E. Perkins, E. M. Royer, S. R. Das, http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt.

[10] C. Zhu and M. Scott Corson, "QoS routing for mobile ad hoc networks," IEEE INFOCOM 2002.

[11] J. Tsai T. Chen and M. Gerla, "QoS Routing Performance in Multihop Multimedia, Wireless Networks" In Proc. of IEEE ICUPC, 1997.

[12] C. R. Lin, "On demand QoS routing in multihop mobile networks," In Proc. of Infocom, 2001.

[13] A. N. Zadeh, B. Jabbari, R. Pickholtz, B. Vojcic, "Self-organizing packet radio ad hoc networks with overlay (SOPRANO)," IEEE Communications Magazine, Jun 2002, pp. 149-157.

[14] C. Qiao, H. Wu, and O. Tonguz, " Load balancing via relay in next generation wireless systems," in Proc. of IEEE Mobile Ad Hoc Networking & Computing, pp. 149-150, 2000.

[15] J. Broch, "Performance comprasion of multihop wireless ad hoc network routing protocols," ACM Mobicom, Oct. 1998.

[16] S. Lee, M. Gerla, and C. K. Toh, "A simulation study of table driven and on-demand routing protocols for mobile ad hoc networks," IEEE Network Magazine, Aug. 1999.