MZR: A Multicast Protocol for Mobile Ad Hoc Networks

Vijay Devarapalli
Nokia Research Center
313 Fairchild Drive,
Mountain View, CA 94043
vijayd@iprg.nokia.com

Deepinder Sidhu
Maryland Center for Telecom Research
University of Maryland Baltimore County
1000 Hilltop Circle,
Baltimore, MD 21250
sidhu@umbc.edu

***Abstract* - This paper proposes a new multicast protocol for Mobile Ad Hoc networks, called the Multicast routing protocol based on Zone Routing (MZR). MZR is a source-initiated on-demand protocol, in which a multicast delivery tree is created using a concept called the zone routing mechanism. It is a source tree based protocol and does not depend on any underlying unicast protocol. The protocol's reaction to topological changes can be restricted to a node's neighborhood instead of propagating it throughout the network. A detailed simulation and performance analysis of MZR is presented in this paper.**

I. INTRODUCTION

Mobile Ad Hoc networks are self-organizing network architectures in which a collection of mobile nodes with wireless network interfaces may form a temporary network without the aid of any established infrastructure or centralized administration. The characteristics that distinguish these networks from wired networks include a distributed peer-to-peer mode of operation, multi-hop routing over wireless links, and relatively frequent changes in topology. Communication between mobile nodes in an ad hoc network requires routing over multiple-hop wireless paths. The main difficulty in designing a routing protocol for these networks is the dynamically changing topology, due to the random movement of mobile nodes.

Much work has been done in developing unicast protocols for ad hoc networks. However, multicast communication is more applicable for ad hoc networks than unicast communication, because in a typical ad hoc environment, it is mostly the case that the mobile nodes work as a group and are involved in collaborative computing. We propose a new multicast routing protocol for ad hoc networks in this paper.

II. RELATED WORK

Other multicast routing protocols have been proposed for ad hoc networks. On-Demand Multicast Routing Protocol (ODMRP) [5] is a mesh based multicast protocol in which a mesh of nodes for forwarding packets is created between the senders and receivers. The mesh is created using the forwarding group concept. The main disadvantage with ODMRP is the excessive overhead incurred in keeping the forwarding group current and in the global flooding of the JOIN-REQUEST packets. The Adhoc Multicast Routing Protocol (AMRoute) [6] is a shared tree based protocol, in which a bi-directional shared user-multicast tree is created involving only the group members. The tree links are created as unicast tunnels between the tree members. The problem with AMRoute is that it depends heavily on an underlying unicast protocol for creating these unicast tunnels. AMRoute is shared tree based protocol. The shared tree approach has a few drawbacks. First, paths are non-optimal and traffic is concentrated on the shared tree, rather than being evenly distributed across the network. Secondly, all shared tree based protocols need a group leader (or a core or a rendezvous point) to maintain group information and to create multicast trees. Group leader can become central point of failure. Also network mobility tends to increase the overhead for group leader selection. A multicast extension of AODV has also been proposed [7].

III. MZR

The multicast routing protocol described here is a source initiated, on-demand routing protocol. It is source-initiated because a multicast delivery tree is created when the source needs to send multicast data to its group members. It is on-demand because the delivery tree is created only when there is data to be sent to the group members. The protocol belongs to the family of Source Tree Based multicast protocols. For each multicast session, identified by a <source_id, group_id> pair, a multicast delivery tree rooted at the source is created. It does not depend on any underlying unicast routing protocol for its operation. It also belongs to a class of routing protocols called the Zone Routing Protocols. The original Zone Routing Protocol [8] was proposed at the Wireless Networks Lab, Cornell University. MZR uses the zone routing mechanism in ZRP for creating and maintaining multicast trees. So we call it the Multicast Routing Protocol based on Zone Routing (MZR). MZR assumes bi-directional links between the mobile nodes.

A. Zone Routing

Existing routing protocols can be classified either as proactive or reactive routing protocols. Pure proactive protocols [1] are not applicable for MANETs, as they continuously use large portion of the network bandwidth, trying to keep the routing information current. This overhead becomes quite significant because of the highly dynamic topology in MANETs. The changes in topology could be more frequent than routing requests, in which case most of the routing information is stale. Pure reactive protocols [2], [3] are not entirely suited for MANETs either. Since route information may not be available at the time a routing request is received, the delay to determine a route can be quite significant. Moreover the on-demand route discovery process is typically implemented as a global search process, in which route request packets are flooded throughout the network in search of a particular destination. This global flood-search procedure requires significant control traffic in a small interval of time, which can saturate the network. Another related issue is a routing protocol's reaction to topology changes. A topology change at one end of the network is an important local event, but not significant at the other end of the network. Proactive protocols tend to distribute such topological changes widely in the network, incurring large costs.

The Zone Routing Protocol (ZRP) [8] is a hybrid (proactive and reactive) routing protocol suitable for mobile ad hoc networks. It limits the scope of the proactive procedure only to the node's local neighborhood. On the other hand, the reactive global search is done efficiently by querying only selected nodes in the network, as opposed to querying all the nodes [9]. Each node in the network defines a zone around itself with the zone radius measured in terms of network hops. The beauty of the original Zone Routing Protocol is in the way the global flood search algorithm is implemented. Instead of flooding the route request packet throughout the network, a node selectively queries its border nodes for a route to a particular destination. These border nodes first look in their zones for the destination. If not found, they query their border nodes in turn. The destination node eventually gets the route discovery packet and replies to the source. As the zone radius is significantly smaller than the network radius, the cost of learning the zones' topologies is a very small fraction of the cost required by a global proactive mechanism. Zone routing is also much cheaper and faster than a global reactive route discovery mechanism, as the number of nodes queried in the process in on the order of $[r_{zone} / r_{network}]^2$ of the number of nodes queried by a global flooding process (r_{zone} is the zone radius and $r_{network}$ is the network radius). MZR makes use of this querying mechanism for building source based multicast delivery trees.

B. Protocol Description

MZR has two parts to it. A proactive protocol runs inside each zone, maintaining an up-to-date zone routing table at each table. A reactive multicast tree creation is initiated when a source needs to send multicast data to its group members. The following sections describe the multicast protocol in brief.

C. Zone Construction and Maintenance

Each mobile node participating in an ad hoc network constructs a zone around itself with a pre-configured zone radius. A simplified distance vector protocol is implemented for creating zones and for maintaining a Zone Routing Table at each node. Every node in the mobile ad hoc network periodically broadcasts an ADVERTISEMENT packet, identifying itself. The propagation of the advertisement packets is restricted to a zone by setting the time-to-live (TTL) value of these packets to the zone radius. The nodes that are within the transmission range of a node *A* pick up the advertisement packet sent by *A*. Each node that receives an ADVERTISEMENT packet rebroadcasts it, if the TTL of the packet is still valid. When a node *B* receives the advertisement packet from *A*, a route entry for *A* is created and stored in *B*'s zone routing table. The distance to *A* from *B* is set to the hop count in the advertisement packet and the next hop in the route entry is set to the node from which *B* received *A*'s advertisement packet. A soft state approach is followed to remove stale routes from the zone routing table. Route entries expire if advertisement packets from the corresponding destination nodes do not periodically refresh them.

A zone routing table (containing unicast routes to each zone node) is kept up-to-date through this proactive protocol built on periodic advertisements. Routes to destinations that moved away are removed when they expire. Routes to new destinations are added to a node's zone routing table, when it receives advertisement packets from these destinations. Also, the protocol's reaction to the changes in topology is localized to a zone. Only the nodes within a zone are affected and only they need to update their zone routing tables. By looking at the zone routing table and the number of hops to each destination, a node can identify the interior zone nodes and the border nodes. Each node also maintains a Neighbor Table, which contains all those nodes from which a node received ADVERTISEMENT packets with a hop count of one.

D. Multicast Tree Creation

For each multicast session in the ad hoc network, a tree rooted at the source and identified by a <source, group> pair is created. Multicast group information is distributed separately and is not within the scope of this protocol. Every node maintains a multicast routing table, which contains route entries corresponding to each active multicast session in the ad hoc network. A multicast route entry is identified by the multicast session id, a <source, group> pair, and contains the IP address of the upstream node and a list of downstream nodes on the corresponding multicast tree.

A multicast source initiates the creation of a multicast data delivery tree. The tree creation is done in a two-stage process. The source initially forms the tree inside its zone and then tries to extend the tree to the entire network. The source sends a TREE-CREATE to each zone node, through unicast routes obtained from the zone routing table. A TREE-CREATE packet is uniquely identified by the corresponding session id. As the TREE-CREATE packet is propagated to a zone node, reverse route entries are created at each intermediate node. The reverse routes are basically multicast route entries with the list of downstream nodes empty and the upstream node set to the node from which the TREE-CREATE packet was received. These multicast route entries at the intermediate nodes are set to inactive when they are created. When a zone node, interested in the multicast group session, receives the TREE-CREATE packet, it creates a multicast route entry and replies to the source with a TREE-CREATE-ACK packet. The TREE-CREATE-ACK packet is sent back to the source through the reverse route created by the TREE-CREATE packet. As the TREE-CREATE-ACK travels back to the source, the corresponding multicast route entry at each intermediate node is completed, activated and the node from which the TREE-CREATE-ACK was received is add to the list of the downstream nodes. When the multicast route entry is activated at a node, it signals the creation of a new tree branch on which multicast data can be forwarded. Even nodes, which are not interested in the multicast session, become members of the multicast tree if they provide connectivity to downstream member nodes. Figure 1 illustrates multicast tree creation inside a zone. Source *S* initiates tree creation by sending TREE-CREATE messages to all its zone nodes. Nodes *A*, *B* and *D* are interested group members and they respond to the source with TREE-CREATE-ACK messages. A multicast route entry is created at the intermediate node *C* for the tree branch connecting group members *S* and *D*, even though *C* is not interested in the multicast group.

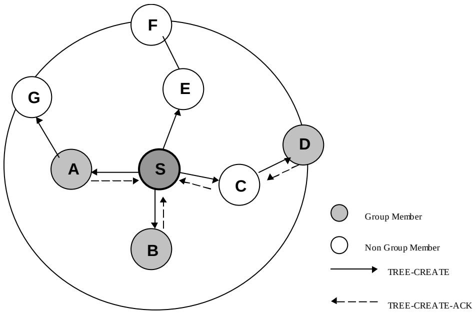

Fig. 1. Multicast Tree creation inside a zone

Once the source is done with its zone, it tries to extend the multicast tree to the entire network. The source identifies all the border nodes in its zone and sends a TREE-PROPAGATE message to each one of them. The TTL of the TREE-PROPAGATE packet is set to the zone radius and the packet is unicasted to the border nodes. Each TREE-PROPAGATE packet is uniquely identified by the corresponding multicast session id. A TREE-PROPAGATE packet basically tells a node to extend the multicast tree inside its zone. When a border node receives a TREE-PROPAGATE packet, it creates a multicast route entry for the session, and then sends a TREE-CREATE packet to all its zone nodes. This is done, even if the border node is not interested in the multicast session. If a node in the border node's zone is interested in the session, it replies to the border node with a TREE-CREATE-ACK. The same procedure as described in the previous section is followed to create multicast route entries at each intermediate node. The border node in turn sends a TREE-CREATE-ACK packet, unicasted to the source. This basically extends the multicast tree into the border node's zone with a unicast link between the source and the border node and multiple tree branches within the border node's zone. Once the border node is done with its zone nodes, it sends a TREE-PROPAGATE message to all its border nodes. These border nodes in turn try to extend the multicast tree inside their zones. This continues till every mobile node in the ad hoc network gets a TREE-CREATE packet corresponding to the multicast tree being created. Figure 2 illustrates the multicast tree extension through the entire network. Source *S* initiates the multicast tree creation. Border nodes *E*, *P* and *N* in turn extend the multicast tree inside their zones and to the rest of the network. Nodes *E*, *I*, *L*, *N* and *P*, though not interested in the multicast group become multicast tree members because they provide connectivity to other member nodes.

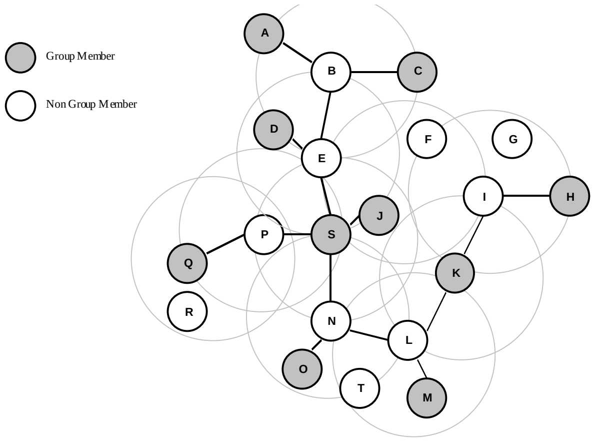

Fig. 2. Multicast Tree Extension through the entire network

A node can be a member of multiple routing zones at the same time because the routing zones heavily overlap. It is possible that a node receives multiple TREE-CREATE and TREE-PROPAGATE messages, effectively flooding the network. This is prevented by early detection and termination of redundant TREE-PROPAGATE threads.

E. Routing Mechanism

The source starts transmitting data packets to the group members once the multicast delivery tree is created. When a node on the multicast tree receives a data packet from its upstream node, it replicates the data packet and sends a copy to each node in the downstream list. A node stops transmitting data packets to a downstream node, if the downstream node migrates and moves out of its transmission range.

F. Multicast Tree Maintenance – Tree Refresh

To ensure continuous multicast data delivery in the presence of node mobility, the multicast tree has to be maintained and reconfigured quickly. The following mechanism maintains up-to-date multicast routing information at each tree member. Each route entry in a node's multicast routing table has a timer associated with it. When the timer expires, the corresponding multicast route entry is removed. This is necessary to remove stale multicast information. To ensure that the multicast route entries do not expire for the duration of the multicast session, the source sends a TREE-REFRESH packet every REFRESH_INTERVAL down the tree. The TREE-REFRESH packet is identified by a unique session id, a <source, group> pair. When a tree member receives a refresh packet, it updates the timer for the corresponding multicast route entry, and sends the refresh packet to all the downstream nodes on the tree. The source stops sending refresh packets once it finishes sending all the data for the corresponding session. This mechanism ensures that a data delivery tree is maintained as long as the session is active.

G. Multicast Tree Maintenance – Reaction to Link Breaks

The downstream nodes are responsible for detecting link breaks and reconfiguring the tree. When a link to an upstream node breaks, the downstream node can easily detect this by looking at its neighbor table. A node (for e.g. *A*) initiates branch reconstruction when it loses connection to its upstream node. The node *A* initiates a global search for the multicast tree by using the zone routing mechanism. It first sends JOIN packets to all its zone nodes. Each JOIN-PACKET is uniquely identified by the corresponding multicast session id. The TTL of the JOIN-PACKET is set to the zone radius. If any of the nodes in *A*'s zone is on the multicast tree and has a valid multicast route entry, it replies to *A* with a JOIN-ACK packet. The node, before replying with a JOIN-ACK packet adds the node from which it received the JOIN request to the list of downstream nodes in the corresponding multicast route entry. Multicast route entries are also set up at the intermediate nodes. Figure 3 illustrates tree repair by node *A*, which lost connection to its upstream node, by sending a JOIN packet to all its zone nodes. Node *G* is connected to the multicast tree and responds with a JOIN-ACK packet. This creates a new tree branch connecting group members *A* and *C* to the multicast tree.

If node *A* does not get a reply from its zone nodes, it tries to propagate its join request through the entire network. It sends a JOIN-PROPAGATE packet to all its border nodes. These border nodes in turn send join requests to their zone nodes. If they get a response from any of their zone nodes, they reply to *A* with a JOIN-ACK. If not, they send a JOIN-PROPAGATE packet to their border nodes. If *A* does not get a reply at all, it assumes that the network has been partitioned and it cannot connect to the existing multicast tree. It repeatedly tries connecting after exponentially increasing intervals. The advantage with this two-stage rejoin process is that the branch reconstruction is localized if *A* manages to find a node within its zone.

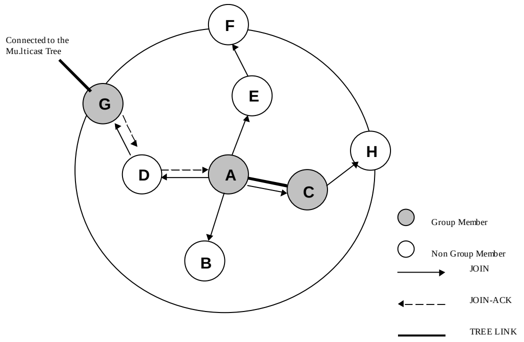

Fig. 3. Rejoin inside a zone

I. Tree Prunes

When a tree member is no longer interested in the group, it sends an explicit TREE-PRUNE message to its upstream node, which in turn removes it from its list of downstream nodes. If a node's downstream list becomes empty as a result if this action and it is not interested in the group, it sends an explicit prune message to its upstream node.

A Message cache is used for detecting duplicate data and control packets and it is maintained at each mobile node. The reliable transmission of the TREE-CREATE and TREE-CREATE-ACK packets is very essential for the multicast tree creation. MZR operates on top of a reliable MAC protocol, the IEEE 802.11, which performs reliable transmission by retransmitting packets if acknowledgments are not received.

IV. Description of Simulation Model

MZR was simulated on the NIST Network Simulator from the National Institute of Standards and Technology (NIST) [10]. The original simulator was developed at the Laboratory of Computer Science at MIT [11]. Our main goal in the simulation was to understand the protocol's multicast routing behavior and detect any major flaws in it. The input network scenario consisted of 50 mobile nodes moving according to the random waypoint model, on a 500m x 500m two dimensional grid. Each simulation executes for 300 seconds of simulation time. The wireless link capacity is assumed to be 2 Mbps. The transmission range in our model is set to 100 meters. Multiple simulation runs were conducted for each experiment with the average taken as the result.

Nodes in the simulation move according to a model called the “random waypoint” model [12]. We use a free space propagation model where the power of a signal decreases inversely with the distance. This model attenuates the power of a signal as $1 / r^2$, where r is the distance between the antennas. A wireless application was created on each mobile host. If the corresponding mobile host is the source for a group, the wireless application generates multicast data for its group members at a constant data rate of 64 Kbps. A constant data packet size of 500 bytes was assumed. In most of the simulation runs we had three multicast sessions running in the ad hoc network.

We chose the zone radius to be two because the optimal zone radius for zone routing protocol was found to be two[13]. A zone radius of one or less would result in a large number of TREE-PROPAGATE messages propagating throughout the network and involve almost every mobile node in the tree creation process. It would also result in longer delays for tree creation. A zone radius of more than two would imply a dominant proactive protocol running in the ad hoc network. This creates excessive routing control traffic and other disadvantages that proactive protocols suffer from.

V. Performance Analysis

A. *Packet Delivery Ratio and Node Mobility*

Packet delivery ratio is the ratio of number of data packets actually delivered to the multicast group members versus the total number of data packets that were supposed to be delivered. This measure represents the effectiveness of the protocol. The dynamic nature of the network causes frequent disruptions and link breaks in the multicast delivery trees. Therefore the speed at which the nodes move is a significant factor affecting the maintenance of a multicast delivery tree once it is created. Figure 4 shows the packet delivery ratio for three simultaneous multicast sessions as the speed of the nodes is varied from 0 m/s to 20 m/s. We notice that the packet delivery ratio is high when the nodes have low mobility and goes down as the speed of the nodes increases. At low speeds, the multicast tree structure is mostly static and therefore the packet delivery ratio is high. At high speeds, the tree links break down quite often, leading to constant branch reconstructions and therefore larger packet losses.

B. *Routing Overhead and Node Mobility*

Routing overhead is calculated as the ratio of control packets sent versus all the packets sent. It is a measure of efficiency of the protocol in terms of channel access and is very important in ad hoc networks since link layer protocols are typically contention-based. Highly mobile networks cause more tree links to break and therefore more branch reconstructions. Since tree reconfiguration involves control traffic, the node mobility is an important factor influencing the routing overhead. Figure 5 illustrates the increase in routing overhead as the speed of the mobile nodes increases from 0 m/s to 20 m/s. At low speeds, the multicast delivery tree is highly stable and the routing overhead is low. The overhead increases, as the ad hoc nodes become more mobile. It reaches almost 40% when the mobile nodes move at a high speed of 20 m/s (72 km/hr).

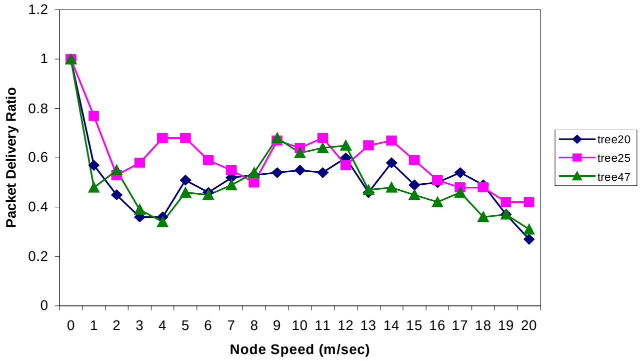

Fig. 4. Packet Delivery Ratio for three active sessions as the mobility increases

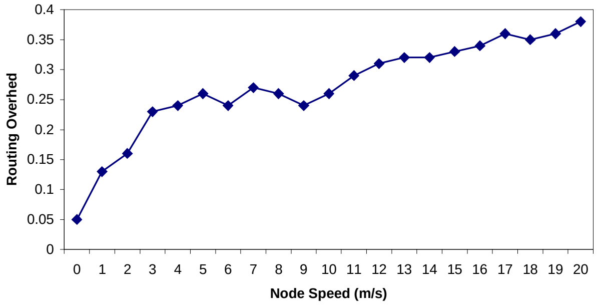

Fig. 5. Total routing overhead as the node mobility increases; pause time of 10 secs

C. *Multicast Group Size*

The multicast group size determines how many nodes are interested in a particular group and are actually involved in the multicast tree setup process. We varied the number of group members to investigate the scalability of the MZR protocol. The following experiments were performed with the nodes moving at a uniform speed of 5 m/s with a 10 second pause time. In a branch reconstruction process, the overhead can be kept to a small value if the node initiating the rejoin process is able to find a tree member inside its own zone itself or close to its neighborhood. If the group size is small, it takes longer for a node, which has lost connection to its upstream node to reconnect to the existing multicast tree. It also involves more control traffic. Figure 6 illustrates how the routing overhead is affected as the size of the multicast group is varied. When the size of the group is small, the overhead is quite high. It is almost 50% when the group size is 5. It decreases as the size of the group increases and is less than 10% when the multicast group includes every mobile node in the ad hoc network. Figure 7 shows how the routing overhead stabilizes as the simulation proceeds for different group sizes. The behavior is almost the same irrespective of the size of the group. From this, we can conclude that the MZR protocol performs well for any multicast group size in the ad hoc network, in terms of a stable and constant routing overhead. The packet delivery ratio is also affected by the multicast group size. It takes a node longer to reconnect to the multicast tree if there are very few group members in the ad hoc network. Figure 8 depicts the performance of the protocol in terms of packet delivery ratio when the size of the group is varied from 5 to 50. As expected the ratio is directly proportional to the group size and increases to more than 90% when the multicast group includes every node in the ad hoc network.

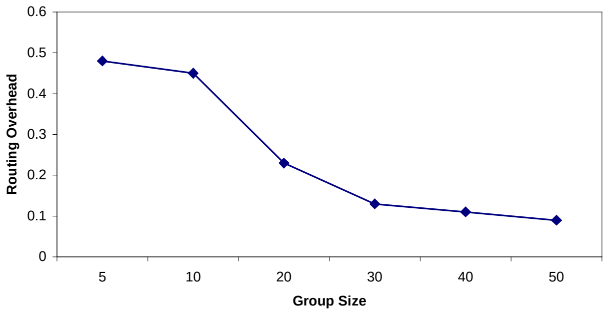

Fig. 6. Total routing overhead as the group size changes; pause time of 10 secs and a speed of 5 m/sec

VI. Conclusions

A new multicast routing protocol (MZR) for ad hoc networks has been proposed in this paper. The proposed protocol is a source initiated, on-demand multicast routing protocol. It builds a multicast delivery tree, rooted at the source and connecting all the group members. MZR guarantees that, within a finite time, every multicast receiver in the ad hoc network joins the multicast sessions it is interested in. MZR implements a very simple proactive protocol built on a periodic beacon mechanism inside each zone. For building the multicast delivery tree, the protocol uses a reactive zone routing mechanism. MZR does not depend on any underlying unicast routing protocol. A performance analysis of the MZR protocol was performed. From the simulation results, we can conclude that MZR scales well for different group sizes and different mobility speeds. The routing overhead does not explode and reaches a stable value for a wide variety of traffic and mobility patterns. In terms of throughput, MZR does well, delivering near 100 percent of the data traffic for low mobility speeds. Even for highly mobile networks, MZR does reasonably well delivering more than 50 percent of the data traffic.

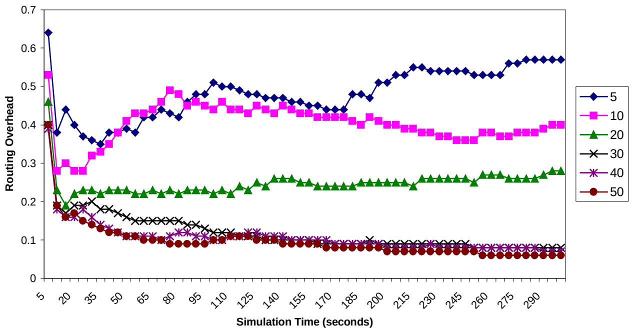

Fig. 7. Total routing overhead as the simulation proceeds for different group sizes; pause time of 10 secs and a speed of 5 m/sec

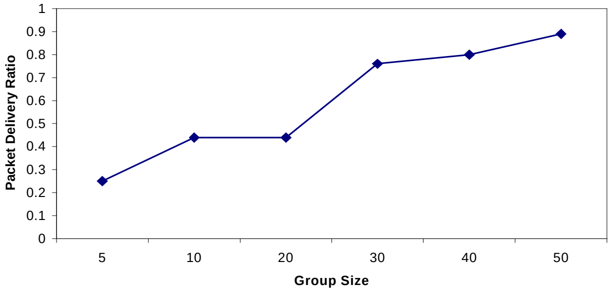

Fig. 8. Total Packet delivery ratio as the group size changes; pause time of 10 secs and a speed of 5 m/sec

References

[1] Charles Perkins and Pravin Bhagwat, " Highly dynamic Destination-Sequenced Distance-Vector routing for mobile computers," *Proceedings of the SIGCOMM* 1994, August 1994

[2] Josh Broch, D. B. Johnson, and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet-Draft, draft-ietf-manet-dsr-03.txt, Oct 1999.

[3] Samir Das, Elizabeth Royer and Charlie Perkins, "Ad Hoc On-Demand Distance Vector (AODV) Routing" Internet-Draft, draft-ietf-manet-aodv-06.txt, July 2000

[4] S.J. Lee and et. al., "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols", *Proceedings of IEEE INFOCOM 2000*, March 2000.

[5] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol", *Proceedings of IEEE WCNC'99*, September 1999

[6] Bommaiah, Ethendranath, and et.al., "AMRoute: Adhoc Multicast Routing Protocol", Intenet-Draft, draft-talpade-manet-amroute-00.txt, August 1998

[7] Elizabeth M. Royer and Charles Perkins, "Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol", *Proceedings of MobiCom'99*, August 1999

[8] Z.J. Haas and M.R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," Internet Draft, draft-ietf-manet-zone-zrp-02.txt, June 1999

[9] Z.J. Haas and Mark Pearlman, "Determining the Optimal configuration for the Zone Routing Protocol", *IEEE JSAC, special issue on Ad Hoc Networks*, vol. 17, no. 8, August 1999

[10] The NIST ATM/HFC Network Simulator, Operation and Programming Guide, December 1998

[11] A. Heybey, "The Network Simulator," *Laboratory of Computer Science*, Massachusetts Institute of Technology, October 1989.

[12] J. Broch, et.al., "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", *Proceedings of IEEE/ACM Mobicom 99*, August 1999.