A Performance Comparison of Zone-Based Multicast Protocols for Mobile Ad Hoc Networks

***Abstract*—*With the current trend toward ubiquitous computing comes wireless devices capable of forming the nodes of mobile ad hoc networks. Such networks typically rely on routing protocols in order to communicate messages from a source node to a destination node through a set of intermediary nodes. In MANETs, it is more effective to use multicast communication, because in a typical ad hoc environment, mobile nodes mostly work as a group and are involved in collaborative computing. This paper presents the comparison of the performance of two zone-based multicast routing protocols for mobile ad hoc networks. Shared-tree MZR is a shared tree variant of the Multicast Routing Protocol based on Zone Routing (MZR). The concept of zone-based multicast routing for mobile ad hoc networks was originally proposed in MZR. We compare the two variants and analyze their performance under various network conditions. The test results show that Shared-tree MZR protocol performs well and has significantly low overhead in scenarios with multiple sources.*

Index Terms—**Ad hoc networks, routing protocols, multicast routing, zone routing.**

I. Introduction

In the area of mobile ad hoc networks (MANETs) [1], there has been an increased interest in the development of ad hoc routing protocols. The development of these protocols is motivated in part by a need to enhance the communication capabilities of current wireless technologies (e.g., Bluetooth) by allowing a node to communicate with another node that is outside of its transmission range.

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. In an ad hoc network, routing is a non-trivial task as nodes can move unpredictably. This behavior causes topological changes that require robust and flexible methods to discover and maintain routes. In MANETs, it is more effective to use multicast communication, because in a typical ad hoc environment, mobile nodes mostly work as a group and are involved in collaborative computing. In response to the severe constraints imposed by ad hoc networks, several multicast protocols have been proposed. Excluding the basic flooding protocol, they can be grouped under two approaches according to their packet distribution algorithms, namely tree-based and mesh-based protocols.

Tree based protocols are further categorized as source tree based and shared tree based protocols. In source based protocols, every node that wants to send data to the multicast group creates a tree and maintains it as long as it stays active as a sender. Source-based routing enables the usage of the optimal tree for every source and better distribution of the network traffic. Despite its advantages, source-based routing can cause significant overhead when there are many sources in a group. If the nodes in the network are highly dynamic, overhead of maintaining a separate tree for every source can be prohibitively expensive. Shared tree protocols create and maintain a single tree for all multicast sources.

In this paper, we compare the source-tree and shared-tree variants of MZR protocol. Section II gives a brief background of tree based multicast protocols. In section III, we describe the two protocols, namely MZR and SH-MZR. In section IV, we conduct simulation experiments to compare the performance of the two protocols under varying network conditions. Finally, we draw conclusions and describe future work in section V.

II. Background

Tree based delivery structure is a well established concept in multicast communication. The properties of a good multicast tree are:

- *Low Cost*: The cost of the multicast tree is the sum of costs of all individual tree links. The cost of the multicast tree should be minimized.
- *Low Delay*: The end-to-end delay from a source node to a group member is the sum of the delay along the tree links. The multicast protocol should try to minimize the delay for each source-destination pair.
- *Scalability*: It should be possible to create a multicast tree for a large number of nodes with reasonable amounts of time and resources. Each node should also be able to support a large number of trees.
- *Survivability*: The multicast tree should be able to survive multiple link and node failures. The importance of this property is increased in ad hoc networks due to the frequent failures of nodes and links due to mobility and loss of battery power in the wireless nodes.

Other properties for a multicast tree include loop freedom and the ability to support dynamic group membership.

Multicast trees can be classified into two categories: source-tree and shared-tree. A key difference, between the two, is that the source tree is optimized for source specific multicast communication, while a shared tree is optimized for communication among the whole group.

In a source tree based multicast routing protocol, a multicast tree is created for every source node participating in a multicast group. Whenever a node in a multicast group wishes to send data, it will create a multicast tree rooted at itself. The tree links are based on the shortest paths from the source node to each node in the multicast group. The source tree distributes the traffic evenly in the network (assuming that sources and receivers are evenly distributed in the network).When the source node finishes transmitting data, the multicast tree is dismantled. No multicast tree exists if none of the sources are transmitting. Source tree multicast algorithms assume that the receiver population is dense and therefore the accompanying control overhead is justified. If the receiver population is not dense, the overhead for maintaining different source trees will be prohibitively high. Source tree protocols do not scale well. If each group member is an active source, a distinct tree has to be created for each node. If a node, which exists in multiple tree, fails then multiple trees have to be repaired. i.e. the control overhead depends on the number of broken trees.

Shared tree protocols are designed to address the scalability issue. In shared tree protocols, a single tree is created for the whole multicast group. There is one designated node responsible for the maintenance of the tree. Each node that wishes to send data to the group, transmits along this tree. Since the tree links are added and deleted dynamically (based on group membership), the resultant tree structure is not optimal for any particular source. The tree structure exists even if none of the sources are transmitting. This also reduces the initial delay that a source tree based protocol experiences during the tree creation phase. Since the delivery structure already exists, the source can start sending data instantaneously. An advantage of shared tree protocols is the decrease in control overhead to repair a broken link. Since each node is part of just one multicast tree, only one tree needs to be repaired in case of a node failure. Thus shared tree protocols are less sensitive to node mobility. The shared tree approach has some drawbacks. The tree has to be maintained even when there is no data transmission. This increases the control overhead of the protocol. Since there is only one delivery structure, the traffic is concentrated on the shared tree instead of being evenly distributed over the network. This leads to lower throughput efficiency. The resources (energy) at the nodes on the shared tree are also consumed at a faster rate.

AMRoute [2], AMRIS [3] and MAODV [4] are tree based protocols, in which a shared tree is created involving the entire multicast group. CAMP [5] and ODMRP [6] [7] allow multiple paths to cope with link failures, resulting in a mesh structure. In ODMRP, the mesh is created using the *forwarding group* concept and a reactive approach is followed to keep the forwarding group current. On the other hand, CAMP exemplifies a proactive mesh based protocol. However, to control the overhead effectively and provide scalability, a hybrid approach is needed. Multicast Routing Protocol based on Zone Routing (MZR) [8] follows the hybrid approach by creating a multicast source tree based on the zone routing concept [9]. In a zone routing network, every node maintains a proactive unicast route to every other node within a certain range. SH-MZR [10] is a shared tree variant of MZR. MZR and SH-MZR will be discussed in the following section.

III. Protocol Review

Routing protocols for ad hoc networks are differentiated as proactive, reactive and hybrid protocols. In a proactive protocol, each node constantly maintains a route to every other node in the network. A reactive protocol reduces this overhead by allowing a node to query for a route to a destination node only when it has some traffic for the destination. Hybrid protocols have a reactive as well as a proactive component. Both Multicast based on Zone Routing (MZR) [8] and Shared-tree MZR (SH-MZR) [10] are based on the principles of zone routing.

A. Concept of Zone Routing

The concept of zone routing [9] is a hybrid of proactive and reactive routing protocol components. The scope of the proactive procedure is limited to the node's local neighborhood, called the zone. Each node keeps track of nodes in its zone by running a proactive routing protocol. For routes to destinations outside a node's zone, a reactive route discovery process is initiated. The search throughout the network, although global, is done efficiently by querying only selected nodes in the network, as opposed to querying all the network nodes.

Each node in the network defines its zone with a preconfigured *zone radius.* Nodes that are exactly "zone radius" number of hops away from the node are called the *border nodes.* All other nodes in the zone are called *interior nodes.* As the zone radius is significantly smaller than the network radius, the cost of learning the zone's topologies is a very small fraction of the cost required by a global proactive mechanism. Zone routing is also much cheaper (in terms of control traffic and congestion) and faster than a global reactive route discovery mechanism, as the number of nodes queried in the process is very small.

B. Shared Tree MZR

In Shared-tree MZR, a multicast routing tree is created when a node wants to join the multicast group. A node, that wants to join the multicast group, will first look for the existing tree by sending a *request* message to the nodes within its zone. If the tree exists, a node on the tree will reply back. If the sender does not get a *reply* within a prespecified interval, it unicasts a message to the border nodes to extend the search inside their zones. If any node in the border node's zone is in the tree, it will reply to the border node. The border node will then send a *reply* to the original sender. Otherwise, the same procedure would be followed to extend the search through the network. The original sender, on receiving a *reply*, sends an *activate* message to the node from which it received the reply. The *activate* message will activate the tree link between the sender node and the node which sent the *reply* message. If the sender node does not receive any replies, it repeats this process. If it still does not receive any replies, it assumes that there is no tree in the network and claims itself to be the group leader. The group leader is responsible for maintaining the multicast tree. Periodically, the group leader sends a *tree refresh* message to refresh the tree.

The group leader then initiates tree creation by broadcasting a *tree create* message within its zone. A zone node, interested in the multicast group, replies to the source. This mechanism allows the source to create a multicast tree, rooted at the group leader and extending throughout its zone. Once the group leader is done with its zone, it tries to extend the multicast tree to the entire network. The group leader unicasts a *tree propagate* message to the border nodes of its zone. The border node then repeats the above mentioned process within its zone. If a node in the

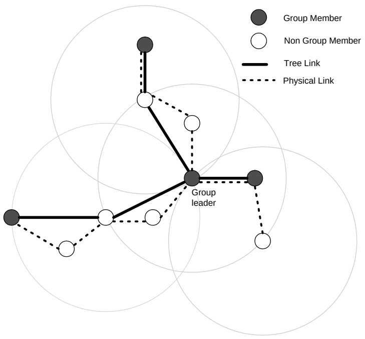


Fig. 1. Tree Extension in SH-MZR

border node's zone is interested in the group, it replies to the border node. The border node in turn sends a reply to the group leader. This basically extends the multicast tree into the border node's zone with a unicast link between the group leader and the border node and multiple tree branches within the border node's zone. Once the border node is done with its zone, it propagates the search to all its border nodes. These border nodes in turn try to extend the multicast tree within their zones. This continues until every node in the network is reached. An example of a tree created by this mechanism can be seen in fig. 1.

The source starts transmitting data packets to the group members once it is connected to the multicast tree. If it becomes the group leader then it starts when the multicast delivery tree is created. When a node on the multicast tree receives a data packet, it replicates the data packet and sends a copy to all links other than the link on which it received the packet.

Node mobility can cause frequent link breakages in the multicast delivery tree. This requires that tree link breakages be detected quickly and the multicast tree reconfigured. The downstream node is responsible for detecting link breaks and reconfiguring the tree. A downstream node "A" initiates a search for the multicast tree by using the zone routing mechanism. It broadcasts a *request* to the nodes in its zone. This message also contains the distance from this node to the group leader in terms of hops. If any node in A's zone is on the multicast tree and its distance to the group leader is less than the distance advertised, it sends a *reply* to A. The node A sends an *activate* message to activate the new tree link.

If node A does not get a reply from its zone nodes, it tries to propagate its search through the entire network by sending a *request propagate* packet to all its border nodes. These border nodes in turn search their zones. If they get a response from any of their zone nodes, they send a *reply* to A. If not, they propagate the search to their border nodes. Using this mechanism, node A's request may be propagated to the entire network. If A does not get a reply at all, it assumes that the network has been partitioned. If there is a network partition, the node, A, claims itself to be the group leader of the part of tree that is on this side of the network partition. It then sends periodic *tree refresh* packets to inform other nodes that there is a new group leader.

The mechanism to merge the two partitioned tree is specified in [10]. The membership of the multicast group can be updated dynamically by adding new nodes and pruning nodes that are no longer interested in the multicast group.

C. Source Tree MZR

The multicast protocol described here is a source initiated, on-demand routing protocol. The multicast delivery tree is created when the source needs to send multicast data to the group members. A multicast source initiates the creation of a multicast data delivery tree rooted at itself and identified by a $< source, group >$ pair. The tree creation is done in a two-stage process. The source initially tries to extend 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. When a zone node, interested in the multicast group, receives this packet, it replies to the source with a *tree create ack*. As the *tree create ack* travels back to the source, the corresponding tree links are activated and the node from which the *tree create ack* was received is add to the multicast tree. Through this mechanism, the source succeeds in creating a multicast tree, rooted at the source and extending throughout its zone.

Once the source is done with its zone, it tries to extend the multicast tree to the entire network. The tree creation is propagated by asking the border nodes to query their zones and their border nodes. Data transmission begins when the multicast tree is created. A node stops transmitting data packets to a downstream node, if the downstream node migrates and moves out of its transmission range.

A soft-state multicast entry is maintained in each tree node's multicast routing table. To ensure that the multicast route entries do not expire for the duration of the multicast transmission, the source sends a periodic *tree refresh* packet down the tree. The source stops sending refresh packets once it finishes sending all the data for the corresponding group. This mechanism ensures that a data delivery tree is maintained as long as the session is active.

In a mobile ad hoc network, tree links are broken frequently because of the change in the topology of the network. To ensure continuous multicast data delivery , the multicast tree has to be reconfigured. The downstream nodes are responsible for detecting link breaks and reconfiguring the tree. The mechanism for branch reconstruction is similar to that described for Shared-tree MZR. The only difference is the reaction of the downstream in case of failure to reconstruct the link. The node assumes that the network has been partitioned and it cannot connect to the existing multicast tree. It repeatedly tries connecting after exponentially increasing intervals.

IV. Simulation and Results

We carried out extensive simulation tests to analyze the performance of the shared-tree and source-tree MZR protocols. In this section we describe the simulation model and summarize the results of the simulations.

A. The Simulation Model

The simulation tests were performed on the NIST Network simulator [11] developed at the National Institute of Standards and Technology. To evaluate the performance of the multicast routing protocols, we setup a packet-level simulation, which allowed us to observe and measure the performance of the protocols under a variety of conditions.

Each mobile node is defined by its position and moves around on a flat two-dimensional grid. The position of a mobile node can be calculated as a function of time and is used by the radio propagation model to calculate the propagation delay from one node to another. It is also used to determine the power level of a received signal at each mobile node. Nodes in the simulation move according to the "random waypoint" model [12]. The movement scenarios are characterized by a *pause time* and *distance* between successive positions of a node. Pause time is the interval that a node remains stationary. The distance between the old and new position is distributed uniformly between a minimum and maximum value. The wireless link is characterized by the link distance and the link bandwidth. The transmission range in our model is set to 100 meters. The wireless link capacity is assumed to be 2 Mbps. The link transmission delay, being dependent on link capacity and packet size, works out to be 2 ms for a packet of size 500 bytes. A separate module generates group information and supplies it to the mobile nodes.

A wireless application created at the source generates multicast data for the group members at a constant data rate of 64 Kbps. The variable simulation parameters include the number of nodes in the system, maximum distance between successive positions of a node, pause time, and the number of sources in a group.

B. Simulation Metrics

We analyze the protocols over two performance measures: packet delivery ratio and protocol overhead.

B.1 Delivery Ratio

Packet delivery ratio is the ratio of the number of data packets actually delivered to the multicast group members to the total number of data packets that were supposed to be received. A measure of this ratio tell us how many packets were lost and not received. A number of factors like node mobility, pause time and erroneous transmissions could be responsible for the packet loss. The delivery ratio of the protocols is naturally limited by the topology of the network (i.e., by its being connected or disconnected).

B.2 Protocol Overhead

The protocol overhead is calculated to include both the control packets and the data packets. Counting the control overhead without the data transmission overhead is not sufficient, because it is always possible to reduce the control overhead by increasing the data transmission.

Control overhead is calculated as the ratio of control packets generated to the total data packets generated by the source(s). Mobile nodes create link breakages in the multicast tree and therefore more branch reconstructions. Since tree reconstruction involves control traffic, node mobility is a major factor that influences control overhead.

Data overhead is important as it determines the efficiency of the multicast delivery structure. It gives a measure of how many non-group-member nodes are present in the data delivery structure. It is calculated as a ratio of data packets received by the nodes in the delivery structure to the product of number of group members and the number of data packets generated by the source(s). Data overhead is indirectly influenced by node mobility.

C. Discussion of the Test Results

The graphs presented here include the results of simulations conducted for 20, 30 and 40 nodes. The results for comparison consist of two sets of simulations. Fig. 2 represents the first set of simulations where the maximum distance between consecutive positions of a node is fixed at 50 meters.

Fig. 2(a)-(c) show the two protocols for various metrics as the pause time is varied from 2 seconds to 50 seconds. For these experiments, a single multicast group with one source was considered. The packet delivery ratio is low for highly mobile nodes. It increases as the pause time increases, i.e. as mobility reduces. The delivery ratio of SH-MZR is slightly lower than MZR. The decrease can be explained by the following scenario. In a SH-MZR, the tree is rooted at the group leader. The source is just another node in the multicast tree (refer fig. 3(a)). If any of the links near the source break, the source is disconnected from most of the nodes on the multicast tree. This affects the delivery ratio adversely. In case of a source tree based protocol, the multicast tree is rooted at the source (refer fig. 3(b)). The multicast tree is an optimal tree for the source. Even if one of the links near the tree fails, the source is not completely disconnected from the multicast tree. Thus the decrease in delivery ratio is less as compared to shared tree protocol.

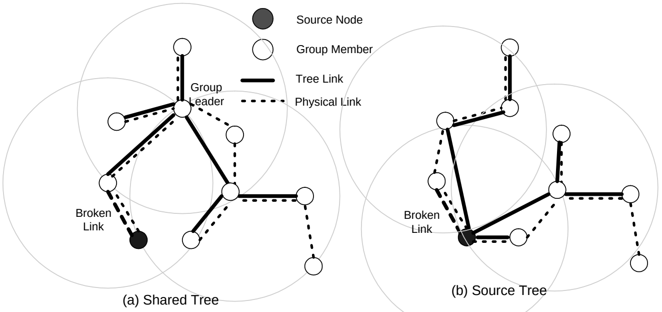
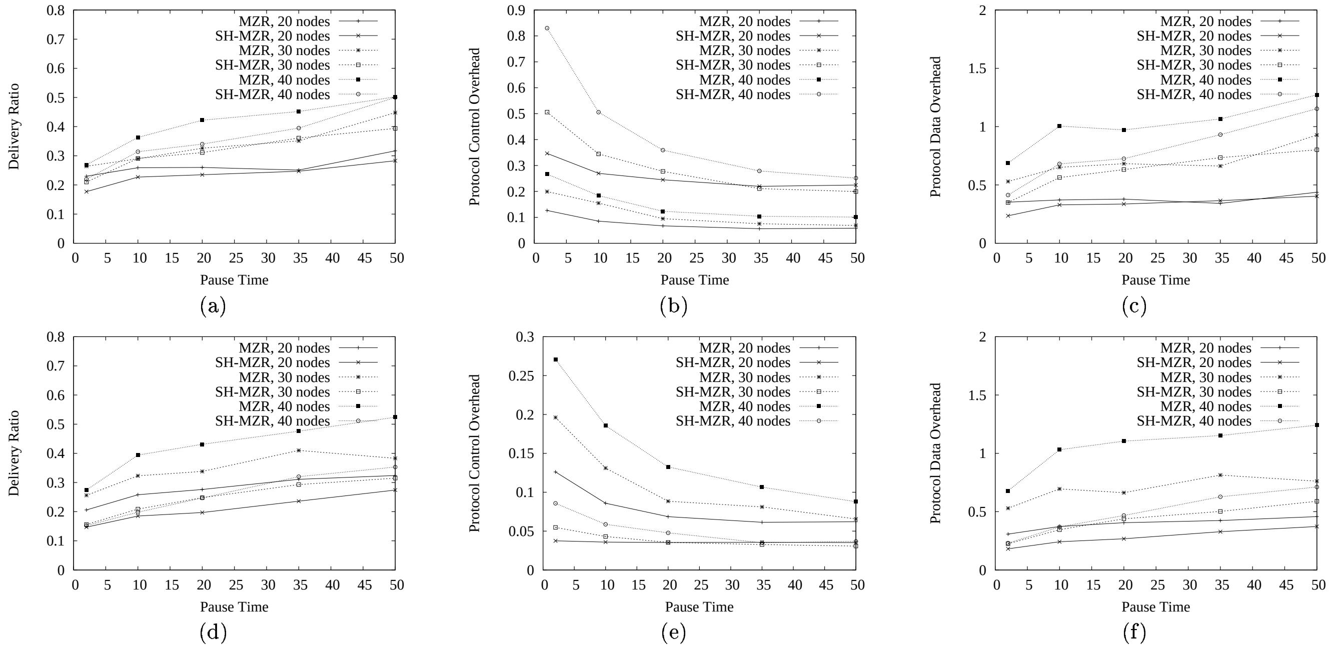


Fig. 3. Mis-alignment of source node in SH-MZR

Another observation from fig. 2(a) is that increase in the number of nodes increases the packet delivery ratio. This is because the dynamic nature of the network topology also depends on where the nodes are located. If the number of nodes is small, then the network is sparsely populated



(a) (b) (c)

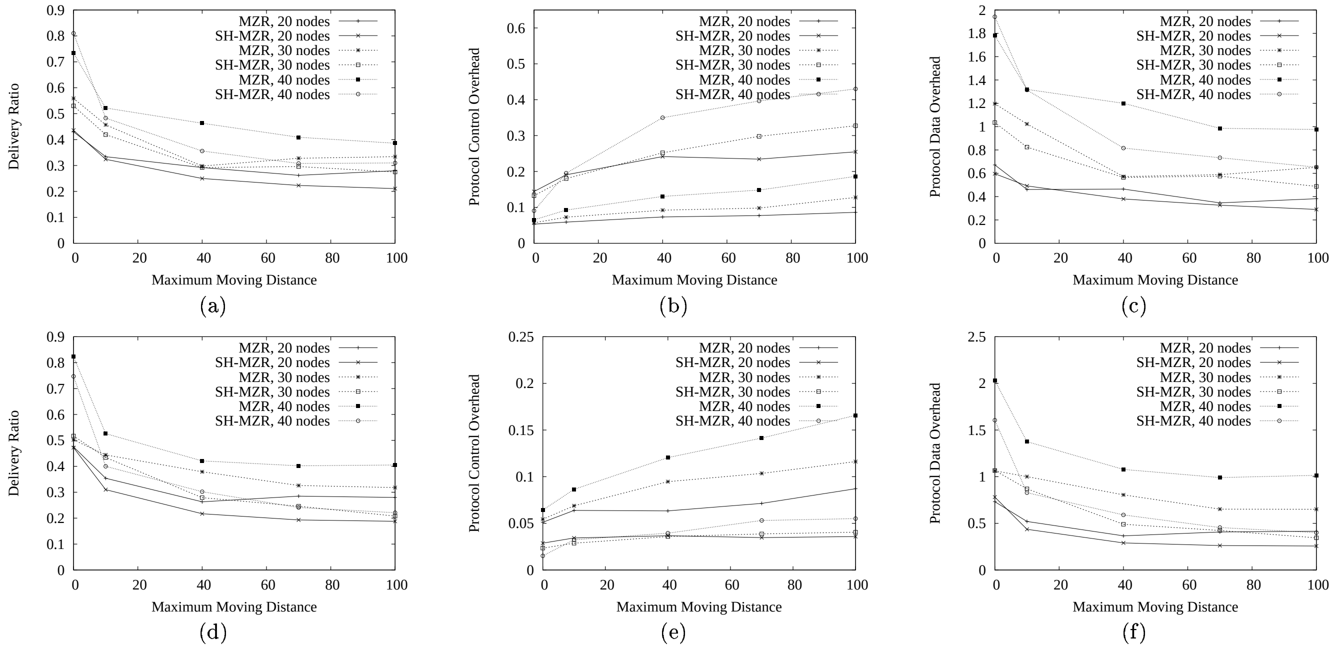
(d) (e) (f)

Fig. 2. Performance Comparison of Source-tree and Shared-tree MZR. Max distance = 50 mts. (a) Delivery ratio, single source. (b) Control Overhead, single source. (c) Data Overhead, single source. (d) Delivery ratio, six sources. (e) Control Overhead, six sources. (f) Data Overhead, six sources

and hence network connectivity is low. There may even be network partitions. As the number of nodes increases, the connectivity increases and hence the packet delivery ratio also increases. Fig. 2(b) shows the variance in routing control overhead as the pause time is varied. As mobility reduces, link breakages are rare and therefore the need to repair broken tree links is less. Thus the routing control overhead decreases. The control overhead for SH-MZR, for simulations with one source, is greater than that of MZR. This is due to the excess processing required to maintain the multicast tree. The tree has to be maintained even if no source is transmitting data. Fig. 2(c) shows that the graph for protocol data overhead is quite similar to the graph for delivery ratio. This is due to the fact that reduction in link breakages increases the number of group members present in the tree and reachable from the source. This has a twin effect of increasing delivery ratio as well as data overhead.

Fig. 2(d)-(f) show the graphs for the same experiments but the number of sources in the multicast group is increased to six. It can be seen from fig. 2(d) that the delivery ratio of MZR does not change significantly as the number of sources is increased. But the delivery ratio of SH-MZR is decreased. This is due to the centralized nature of the delivery structure. All sources in SH-MZR use the same multicast tree. If any link on the tree is broken, the traffic from all the sources gets disrupted. In a source tree protocol, if a tree link is broken, it affects one source only. Thus the link breakages affect SH-MZR more severely than MZR. Fig. 2(e) shows a drastic decrease in the control overhead of SH-MZR. This is due to the fact that the data is delivered by a shared tree. Once a tree is created, the only control overhead is due to the control packets sent for tree maintenance. Since there is one common shared tree, this overhead is divided over the multiple sources. Thus as the number of sources increases, the divided overhead decreases.

Fig. 4 represents the second set of simulations. Here the pause time is kept constant and the maximum distance between two consecutive positions of a node is varied. Maximum distance of 0 meters represents the scenario where the network is completely static. Maximum distance of 100 meters implies a highly mobile network. Fig. 4(a)-(c) show the graphs for a multicast group with a single source. Packet delivery ratio is very high for a static network. But as the nodes move further away, packet delivery ratio drops drastically. Fig. 4(a) shows that variation in delivery ratio is consistent with the discussion in the previous paragraph. for a particular distance and pause time, delivery ratio increases as the number of nodes increase. This happens due to increase in network connectivity. Fig. 4(b) shows that control overhead decreases with reduced mobility. Highly mobile nodes cause more tree links to break and therefore more reconstructions. Since reconstruction involves control traffic, node mobility is an important factor influencing the amount of protocol control overhead. Fig. 4(d)-(f) show the results when the same experiments were conducted for a multicast group with six sources. There is not a significant change in the graphs of delivery ratio and data overhead. But the control overhead is reduced considerably. The control overhead of SH-MZR is high for a single source. But as the number of source is increased, the control overhead is very low as compared to MZR. This is due to the fact that all six sources use the same tree and the tree maintenance cost, i.e. the control overhead, is



(a) (b) (c)

(d) (e) (f)

Fig. 4. Performance Comparison of Source-tree and Shared-tree MZR. Pause time = 20 sec. (a) Delivery ratio, single source. (b) Control Overhead, single source. (c) Data Overhead, single source. (d) Delivery ratio, six sources. (e) Control Overhead, six sources. (f) Data Overhead, six sources

distributed over the different sources.

Remark. It should be noted that the simulations given here are specific to the model described in Section IVA, where there is a relatively high rate (64 Kbps) and continuous data transmission. Also the node mobility in these simulations is high. It is impossible to give the test results for all possible multicast models here due to the space limitations. The results described here should be taken as an analysis for multicast groups with relatively high data rate and continuous transmission.

V. Conclusions

In this paper, we compared source tree and shared tree variants of a protocol for multicasting in mobile ad hoc networks. Both the protocols deploy zone routing to maintain multicast routes. The zone-based routing increases the robustness of the multicast tree in face of moving nodes (i.e. reduces the chance of links being broken in the tree) and enables more rapid recovery when a link is broken.

We performed extensive simulations to compare the performance of SH-MZR and MZR. MZR outperforms SH-MZR when the number of sources is less. But as the number of sources increases, the drastically increasing control overhead of MZR makes it infeasible. Shared tree MZR scales better than source tree MZR. The test results showed that SH-MZR protocols performs quite well except in cases where network is extremely sparse. The improvements were most significant when there were multiple sources in the multicast group. Future extensions to this work will include a comparison of SH-MZR with other multicast protocols.

References

[1] S. Corson and L. Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” *RFC 2501*, January 1999.

[2] M. Liu, R. R. Talpade, A. McAuley, and E. Bommaiah, “AMRoute: Adhoc Multicast Routing Protocol,” *Technical Report*, vol. TR 99-8, The Institute for Systems Research, Univesity of Maryland, 1999.

[3] C.W. Wu and Y.C. Tay, “AMRIS: A Multicast Protocol for Ad hoc Wireless Networks,” in *Proceedings of IEEE MILCOM’99*, November 1999.

[4] Elizabeth M. Royer and Charles Perkins, “Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol,” *Internet-Draft*, vol. draft-ietf-manet-maodv-00.txt, 2000.

[5] J.J Garcia-Luna-Aceves and E.L. Madruga, “The Core-Assisted Mesh Protocol,” *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 8, August 1999.

[6] S.-J. Lee, M. Gerla, and C.-C. Chiang, “On-Demand Multicast Routing Protocol,” in *Proceedings of IEEE WCNC’99*, September 1999.

[7] S. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, “A performance Comparison Study of Ad Hoc Wireless Multicast Protocols,” *IEEE INFOCOM 2000*, March 2000.

[8] V. Devarapalli and D. Sidhu, “MZR : A Multicast Protocol for Mobile Ad Hoc Networks,” *IEEE International Conference on Communications*, vol. Proceeedings of ICC 2001, June 2001.

[9] Z.J. Haas and M.R. Pearlman, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks,” *Intenet-Draft*, vol. draft-ietf-manet-zone-zrp-04.txt, January 2001.

[10] A. Rangnekar, Y. Zhang, A. Selcuk, A. Bicak, V. Devarapalli, and D. Sidhu, “A Zone-Based Shared-Tree Multicast Protocol for Mobile Ad Hoc Networks,” *Technical Report*, vol. TR-CS-03-23, Department of Computer Science and Electrical Engineering, UMBC, March 2003.

[11] N. Golmie, F. Mouveaux, L. Hester, Y. Saintillan, A. Koenig, and D. Su, “The NIST ATM/HFC Network Simulator,” *Operation and Programming Guide*, December 1998.

[12] Josh Broch, D. B. Johnson, and D. A. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” *Internet-Draft*, vol. draft-ietf-manet-dsr-08.txt, Feb 2003.