Robust Tree-based Multicasting in Ad hoc Networks

Ravindra Vaishampayan
Department of Computer Science
University of California Santa Cruz
ravindra@cse.ucsc.edu

J.J. Garcia-Luna-Aceves
Department of Computer Engineering
University of California Santa Cruz
jj@cse.ucsc.edu

***Abstract*— We examine on-demand multicasting in ad hoc networks. We study a wide range of simulation scenarios and identify key limitations of MAODV. Based on our findings, we propose a number of changes in MAODV, and call the resulting protocol *robust multicasting in ad hoc networks using trees (ROMANT)*. We compare ROMANT to MAODV and ODMRP, for a wide range of simulation scenarios. Our results indicate that ROMANT effectively eliminates the limitations of MAODV. Moreover, it provides comparable or better packet delivery ratio than ODMRP at only a fraction of the overhead incurred by ODMRP.**

***Keywords*— Simulations, ad hoc networks, multicasting, meshes, trees, cores.**

I. INTRODUCTION

Mobile ad hoc networks have applications in a wide range of areas including disaster relief and military. Most of these scenarios need one to many or many to many communication. In fact, some networks may need multicast routing only and not need unicast routing at all. This makes multicasting a very important feature in such networks. As a result, it is important to have a multicasting protocol that provides a high packet delivery ratio even in extreme conditions (e.g., high mobility and high traffic load). It is equally important for such protocols to have a low overhead, because bandwidth and battery power are extremely precious in these kinds of networks.

Over the past few years, several multicast routing protocols have been proposed for ad hoc networks [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. For the purposes of our discussion, the approaches taken to date can be classified into tree-based and mesh-based approaches.

A tree-based multicast routing protocol establishes and maintains either a shared multicast routing tree or multiple source-based multicast routing trees (one for each group source) to deliver data packets from sources to receivers of a group. As a tree has only one path from senders to receivers it is extremely important that broken links be fixed quickly and effectively.

In contrast, a mesh-based multicast routing protocol maintains a mesh consisting of a connected component of the network containing all the receivers of a group. Because the multicast mesh may have multiple paths to the receivers, failures of links and nodes are less disruptive for packet delivery than in trees. However, redundancy also results in higher overhead because of data packet duplication. Furthermore, under conditions of high traffic load, mesh based protocols congest the network faster which may lead to a drop in the packet delivery ratio.

Section II describes the related work on multicasting for ad hocnetworks. Section II-A examines MAODV [4] in detail and identifies a number of limitations. Section III describes ROMANT, which is derived from the changes we propose to overcome the limitations of MAODV. Section IV compares ROMANT to MAODV and ODMRP [2]. Our simulation results indicate that ROMANT's packet delivery ratio remains high even in those scenarios where the packet delivery ratio of MAODV drops. The overhead of ROMANT is comparable or less than that of MAODV for all simulation scenarios. Furthermore, although ROMANT is a tree-based protocol, it provides the same or better packet delivery ratio than ODMRP, which is a mesh-based protocol, while incurring only a fraction of the overhead incurred by ODMRP.

II. RELATED WORK

Recent examples of tree-based protocols are MAODV and ADMR [8]. MAODV maintains a shared tree for each multicast group, consisting of only receivers and relays. Sources wishing to send to the group acquire routes to the group on demand in a way similar to the ad hoc on demand distance vector (AODV) [17] protocol. Each multicast tree has a group leader, which is the first node to join the group in the connected component. The group leader in each connected component periodically transmits a group hello packet to become aware of reconnections. Receivers join the shared tree with a special route request. The route replies coming from different multicast tree members specify the number of hops to the nearest tree member. The node wishing to join the tree joins through the node reporting the freshest route with the minimum hop count to the tree. The primary drawback of MAODV is high delay and overhead in fixing broken links in conditions of high mobility and traffic load. It may be argued that its dependence on a unicast routing protocol (AODV) makes it less flexible.

ADMR maintains source-based trees, i.e., a multicast tree for each source of a multicast group. A new receiver performs a network-wide flood of a multicast solicitation packet when it needs to join a multicast tree. Each group source replies to the solicitation, and the receiver sends a receiver join packet to each source answering its solicitation. An individual source-based tree is maintained by periodic keep-alive packets from the source, which allow routers to detect link breaks in the tree by the absence of data or keep-alive packets. A new source of a multicast group also sends a network-wide flood to allow existing group receivers to send receiver joins to the source. MZR [15] like ADMR, maintains source based trees. MZR performs zonal routing, so control packet floods are less expensive. Compared to approaches based on shared trees, the use of source-based trees as in ADMR and MZR creates much more state at those routers participating in many groups, each with multiple sources.

Recent examples of mesh based protocols are CAMP [1] and ODMRP. ODMRP requires control packets originating at each source of a multicast group to be flooded throughout the ad hoc network. The control packet floods help repair the link breaks that occur between floods. The limitations of ODMRP are the need for network-wide packet floods and requiring that the sources of multicast packets for a group be part of the group's multicast mesh, even if such sources are not interested in receiving multicast packets sent to the group. DCMP [14] is an extension to ODMRP in that it designates certain senders as cores and reduces the number of senders performing flooding. NSMP [16] is another extension which aims to restrict the flood of control packets to a subset of the entire network. However both DCMP and NSMP do not entirely eliminate ODMRP's drawback of multiple control packet floods per group.

CAMP avoids the need for network-wide floods from each source by using one or more cores per multicast group. Routers rely on a unicast routing protocol to obtain routing information for the cores. A receiver-initiated approach is used for receivers to join a multicast group by sending unicast join requests towards a core of the desired group. The drawback of CAMP is that it needs a unicast routing protocol to maintain routing information about the cores, which may incur considerable overhead in a large ad hoc network.

A. Limitations of MAODV

Based on simulation results shown in Figures 5(a), 6(a), 6(c) we can see that the packet delivery ratio of MAODV is low in in scenraios with high mobility, large numbers of members, or high traffic loads. We also note that the drop in the packet-delivery ratio is not gradual. When a certain threshold is crossed in terms of mobility, number of members, or traffic load, we see from these figures that the packet-delivery ratio drops drastically. We call this threshold the "stress threshold". This is accompanied by a corresponding increase in packet overhead, as shown in Figures 5(b), 6(b), 6(d).

A careful analysis of the packets sent in all three scenario show that a large number of RREQ (with Join flag set), RREP and MACT packets are sent. These are the packets associated with tree reconstruction. This indicates that the multicast tree is unstable and needs significant reconstruction activity. A multicast tree becomes unstable when the likelihood of links breaking increases. Links are assumed to break if neighbors do not hear each other's hello packets. The multicast tree can become unstable due to different reasons. In the case of high mobility, links actually break when nodes move in and out of each other's range. In the case of large numbers of members, the multicast tree is much larger. Assuming that a certain fraction of links are broken, a larger number of links means that a larger number of links are broken. In the case of higher traffic load, the links are not really broken; however, a larger number of packets are lost due to collisions. Hence, when hello packets are lost due to collisions, nodes infer erroneoulsy that links have been broken. We call this phenomenon an "apparent link break".

Our analysis leads us to believe that MAODV's response to fixing broken links is its greatest limitation. The fact that nodes believe that links are being broken indicates that the network is operating in stress mode, and MAODV responds by injecting three kinds of packets, i.e., RREQ, RREP and MACT packets. As a result, many RREQ packets may be flooded if a RREP packet is not received soon enough. The injection of these packets may in fact lead to more apparent link breaks due to the loss of more hello packets in collisions, which in turn leads to the injection of more RREQ, RREP and MACT packets, in an attempt to fix these new link breaks. As a result of this cyclic nature of congestion, there is sharp decrease in packet delivery ratio and a sharp increase in control overhead as the network crosses a certain "stress threshold".

III. ROMANT DESCRIPTION

Our principal objective in designing ROMANT was to avoid the problem in fixing broken links faced by MAODV. To do so, we needed a new approach to tree building and maintenance. Instead of using a new kind of control packet we used an already existing control packet, the group hello and used it for an entirely new purpose.

A. Group Hello

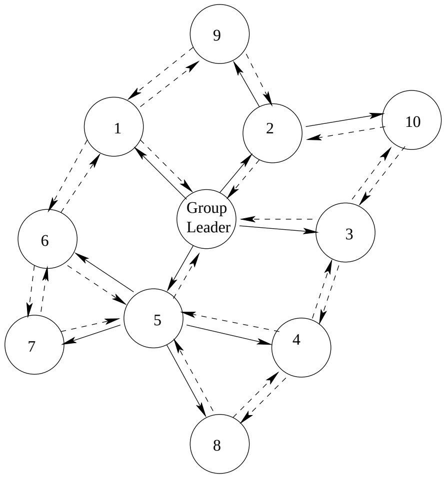
The group hello in ROMANT like the group hello in MAODV is periodically broadcast by the group leader. The group leader in ROMANT, like in MAODV is the first receiver that joins the group.

The process of merging of partitions in ROMANT is much simpler than that of MAODV. MAODV has an elaborate mechanism where a node from a partition with the group leader with a lower address explicitly requests its leader for permission to perform the reconnection. On receiving such permission, it explicitly requests the new leader whether the partitions can merge. In ROMANT, nodes receiving group hello messages from two group leaders simply disregard the messages from the group leader with a lower address. Eventually, the group leader with a lower address receives the group hello from the the group leader with a higher address and stops sending group hello's. We call this kind of merging "passive merging" as opposed to the "active merging" of MAODV. The formation of the multicast tree is intrinsically related to the propagation of the group hello's.

B. *Multicast Tree formation and Maintenance*

The sequence number of each group hello is used by nodes to drop duplicate group hello's. As in MAODV, nodes increment the hop count of a group hello before forwarding it. Although duplicate group hello's are dropped, information about each group hello is stored in a data structure called the *connectivity list*. Each entry in the connectivity list stores the next-hop from which the group hello was received, the sequence number of the group hello, and the hop count to the group leader. Entries in the connectivity list are sorted first by sequence number, then by hop count. clashes are resolved based on the address of the next-hop. Hence, each node in the network has multiple next-hops to the group leader. The connectivity list is refreshed every time the group hello packet is flooded. Figure 1 illustrates this. The solid arrows indicate the paths along which the group hellos are received for the first time. The dashed arrows indicate the paths along which additional group hellos are received. Figure 1 indicates the connectivity list built at node 6, which has potentially three neighbors as possible next hops to the group leader. Hence, when router 6 needs to send a packet (either data or control), it can use these routes in the order they exist in the connectivity list.

In ROMANT, each receiver periodically transmits a packet called a "join announcement". The join announcement contains the address of the best next-hop towards the group leader. The next-hop considers itself a member of the multicast tree after receiving a join announcement, and begins the periodic transmission of join announcements sent towards its own next-hop to the group leader. This sequence of periodic transmissions of join announcements forms a tree rooted at the group leader. Figure 2 shows how receivers R1, R2, R3, and R4 form the tree rooted at the group leader. The arrows indicate the direction in which tree members periodically transmit join announcements. If a node does not hear the join announcements of its next hop it assumes that the next hop is not in range and deletes the next-hop from its connectivity list. It then sends join announcements to its next best next-hop. In contrast to MAODV, where all nodes periodically transmit hello packets, only tree members transmit join announcements in ROMANT.



Connectivity List at Node 6		
Next Hop	Hops To Group Leader	Group Hello Sequence Number
5	2	79
1	2	79
7	3	79

Fig. 1. Dissemination of group hellos in an ad hoc network

The most important reason for the better performance of ROMANT compared to MAODV is the manner in which the multicast tree is created and maintained. At speeds higher than 10 m/s, MAODV suffers a drop in packet delivery ratio as shown in Figure 5(a). For 50 nodes moving at 15 m/s in an area of 1000m × 1000m it can be shown that the average link life is around 10 seconds. In MAODV, individual links are repaired only when broken. As a result, the overall structure of the tree would be far from optimal over a period of time due to the random movement of nodes, which makes the tree susceptible to even more link breakages. On the other hand, the number of broken links in ROMANT is much lower because an optimal[1] tree is constructed every time a group hello is sent (i.e., every 3 seconds) which is significantly less than the average link life. As a result, most

---

[1] *Based on the flooding of group hello's and the sending of join announcements along a shortest path a group hello was received, it is trivial to prove that each receiver is connected via a shortest path to the group leader*

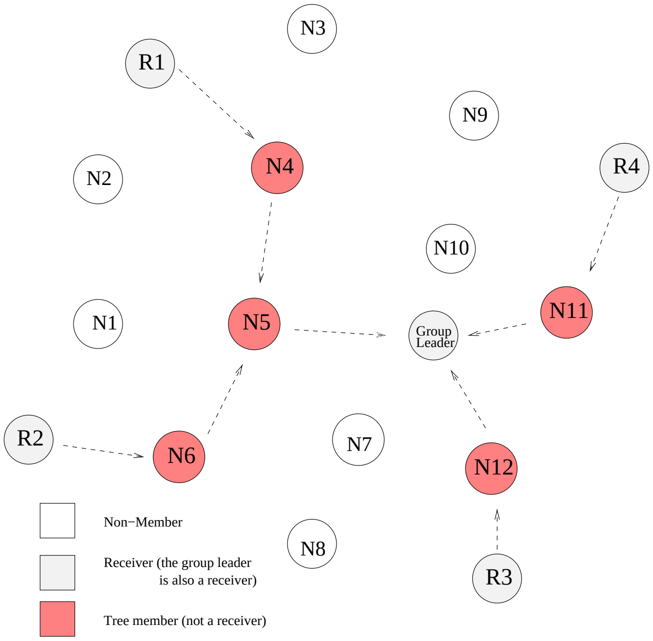


Fig. 2. Tree formation in ROMANT

links are repaired before they are broken.

Furthermore, even when a link breaks, nodes in ROMANT can look up an alternate next-hop in the connectivity list. By contrast, in MAODV, fixing a link takes much longer, because it involves a RREQ-RREP-MACT exchange with a tree member that may be several hops away! The injection of more control packets in order to fix the link breakages leads to packet collisions and more link breakages, which further aggravates the situation, as explained in Section II-A. All scenarios resulting in higher link breakages viz. a very large multicast tree or high traffic load also have the same effect in MAODV.

C. Data Forwarding

In MAODV, when a sender needs to send data packets to a multicast group, it acquires a route to the multicast tree on demand. This involves injecting a RREQ packet in the network, to which all tree members reply with a RREP packet. The sender then chooses the route that offers the lowest cost. By contrast, this kind of route acquisition is not required in ROMANT, because all nodes are already aware of their next-hop towards the group leader based on their connectivity list. Nodes simply forward their data packets to their next hop to teh group leader. Due to the broadcast nature of wireless ad hoc networks, nodes receive an implicit acknowledgment once the data packet is forwarded by the next-hop. If no implicit acknowledgment is received within a certain time, the next-hop is removed from the connectivity list and a the next best next-hop is chosen for future data packet forwarding. This process continues, until the data packet reaches the first multicast tree member. From there, the data packet is flooded within the multicast tree with a packet cache used to drop duplicate packets.

Simulator	Qualnet 3.5
Total Nodes	50
Simulation Time	350 seconds
Simulation Area	1000m X 1000m
Node Placement	Random
Pause Time	0
Mobility Model	Random Waypoint
Radio Range	250m
Channel Capacity	2 Mbps
MAC protocol	IEEE 802.11−1997
Data packet size	512 bytes

Fig. 3. Simulation Environment

IV. PERFORMANCE COMPARISON

We compared the performance of ROMANT against the performance of ODMRP [2] and MAODV [4] which are two state-of-the-art multicast routing protocols for ad hoc networks. Figure 3 gives details about the simulation environment.

The distribution of Qualnet had the ODMRP code, which was used for ODMRP simulations. The MAODV code for Qualnet was obtained from a researcher[2] who followed the MAODV IETF Specification [18]. We employed RTS/CTS exchanges when packets were directed to specific neighbors. All other transmissions used CSMA/CA. Each simulation was run for four different seed values. All timer values for ROMANT and ODMRP (i.e., interval for sending JOIN requests and JOIN tables in ODMRP and the interval for sending group hellos and join announcements in ROMANT) were set to 3 seconds. For the MAODV code, the parameter values were as follows: allowed hello loss = 2, group hello interval = 3 sec, hello interval = 1 sec, hello life = 3 sec, pkt id save = 3 sec, prune timeout = 750 ms, rev route life = 3 sec, rreq retries = 2, route discovery timeout = 1 sec, retransmit timer = 750 ms. We have also implemented and tested ROMANT in Linux 2.4.20-8, Red Hat Release 9.

Several experiments were carried out in order to compare the performance of the three protocols. They were:

1) **Impact of Mobility** : In this experiment, the mobility of the nodes was varied from 0 m/s to 20 m/s. 20 nodes were selected as members and 5 were selected as senders. Each sender sent 2 pkts/sec, to give an overall network load of 10 pkts/sec. Results are indicated in Figures 5(a) and 5(b).

---

[2] *We thank Venkatesh Rajendran for providing the simulation code of MAODV*

Metric	Meaning
Packet Delivery Ratio	$\frac{\text{data packets delivered}}{\text{data packets expected}^{*}}$
Control Overhead	$\frac{\text{total control packets transmitted}}{\text{data packets delivered}}$
Total Overhead	$\frac{\text{total control packets transmitted + total data packets transmitted}}{\text{data packets delivered}}$

*data packets expected = data packets sent X number of receivers

Fig. 4. Metrics used for Performance Evaluation

2) **Impact of Number of Senders** : In this experiment, the number of senders was varied across the values {1, 2, 5, 10, 20}. 20 nodes were selected as members and the mobility was set to 5 m/s. The load of the network was kept constant at 10 pkts/sec. Results are indicated in Figures 5(c) and 5(d).
3) **Impact of Number of Members** : In this experiment, the number of members was varied across the values {5, 10, 20, 30, 40}. Five nodes were selected as senders and the mobility was set to 5 m/s. The load of the network was kept constant at 10 pkts/sec. Results are indicated in Figures 6(a) and 6(b).
4) **Impact of Traffic Load** : In this experiment, the traffic load of the network was varied across the values {1, 2, 5, 10, 25, 50} packets/second. The idea was to study how the protocols behaved under conditions of overload. Five nodes were selected as senders, and the nodes were stationary, so that the packet drops that occurred, were due to collisions and congestion. 20 nodes were selected as members. Results are indicated in Figures 6(c) and 6(d).

The metrics used for our evaluation were **packet delivery ratio** and **total overhead**, which are defined in Figure 4. **Total overhead** is a more important metric than **control overhead** because we are concerned about the number of packets transmitted to get a certain number of data packets to the receivers, regardless of whether those packets were data or control, hence we do not evaluate **control overhead**.

A. Discussion

1) ROMANT vs ODMRP: As we can see from Figures 5(a) and 6(a), the packet delivery ratio of ROMANT is comparable to that of ODMRP for varying mobility and number of multicast members. However, for increasing numbers of senders and increasing traffic load, the packet delivery ratio ROMANT is actually better than that of ODMRP, as shown in Figures 5(c) and 6(c). The packet delivery ratio of ROMANT is significantly higher than ODMRP for more than 10 senders, as shown in Figure 5(c). This is because the per source flooding of ODMRP leads to significant number of packet drops due to congestion as the number of senders is increased beyond 10. In ROMANT on the other hand, the only node that floods the network is the group leader. Similarly, the higher overhead of ODMRP due to per-source flooding and data packet overhead due to its mesh structure results in network saturation much earlier. As a result, when the traffic load is increased beyond 10 packets/second, the packet delivery ratio of ROMANT is higher than that of ODMRP as shown in Figure 6(c). We consider this a significant contribution of ROMANT that it equals or betters the packet delivery ratio of ODMRP, even though it is a tree-based protocol as opposed to ODMRP which is a mesh based protocol.

The packet overhead of ODMRP is twice or more than ROMANT for all simulation scenarios. This is expected and is due to two reasons, namely: (a) ROMANT is a tree based protocol and has a lower data packet overhead than ODMRP, which is a mesh based protocol; and (b) every sender in ODMRP performs flooding, while only the group leader performs flooding in ROMANT.

2) ROMANT vs MAODV: As can be seen from Figures 5(a), 6(a), 6(c), 5(b), 6(b) and 6(d), ROMANT fixes the performance problems faced by MAODV (high control overhead and low packet delivery ratio in situations of high mobility, high traffic load and a large number of members). Moreover, ROMANT does not introduce new problems, i.e., it does not perform worse that MAODV in situations when the performance of MAODV is actually very good.

We believe that ROMANT is a very attractive alternative for the support of multicast routing in ad hoc networks. This is not only because of its impressive performance as discussed earlier, but also because of its ease of implementation, which we tested by implementing ROMANT in Linux 2.4.20-8, Red Hat Release 9. ROMANT has only two kinds of control packets viz. group hello and join announcement. In contrast, MAODV has five viz. group hello, RREQ. RREP, MACT and hello. Additionally, RREQs in MAODV are further subdivided into normal RREQ's, Join RREQ's or Repair RREQ's. Nodes in MAODV need to maintain seven timers, as described in Table 1 of [4]. In ROMANT, on the other hand, a node needs to implement only four timers viz. timers for group hellos, join announcements, implicit data acknowledgments, and for aging old entries from the packet cache. The partition merging mechanism is much simpler in ROMANT when compared to MAODV, as described in Section III-A. Moreover, data packet forwarding is also much simpler in ROMANT, as described in Section III-C. Although ODMRP is also relatively easy to implement, its per-source flooding and high rate of data-packet duplication are major drawbacks.

V. CONCLUSIONS

By studying the reason for the performance problems of MAODV, we derived a new approach to multicast-tree building and maintenance in ad hoc networks called

Robust Multicasting in Ad hoc Networks using Trees (ROMANT). Based on simulations in Qualnet 3.5, we showed that ROMANT eliminates the drawbacks of MAODV, and avoids any dependency on unicast routing protocols (e.g., AODV) without incurring any extra overhead. ROMANT also provides equal or better packet delivery ratio than ODMRP, which is the state of the art mesh-based protocol, at only a fraction of the total overhead incurred by ODMRP.

REFERENCES

[1] J. J. Garcia-Luna-Aceves and E.L. Madruga, "The core assisted mesh protocol," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, vol. 17, no. 8, pp. 1380–1394, August 1999.

[2] S.J. Lee, M. Gerla, and Chian, "On-demand multicast routing protocol," in *Proceedings of WCNC*, September 1999.

[3] S.J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A performance comparison study of ad hoc wireless multicast protocols," in *Proceedings of IEEE INFOCOM, Tel Aviv, Israel*, March 2000.

[4] E. Royer and C. Perkins, "Multicast operation of the ad hoc on-demand distance vector routing protocol," in *Proceedings of Mobicom*, August 1999.

[5] Park and Corson, "Highly adaptive distributed routing algorithm for mobile wireless network," in *Proceedings of IEEE INFOCOM*, March 1997.

[6] J. Xie snd R.R. Talpade, A. McAuley, and M. Liu, "Amroute: Ad hoc multicast routing protocol," *Mobile Networks and Applications (MONET)*, December 2002.

[7] C.W. Wu and Y.C. Tay, "Amris: A multicast protocol for ad hoc wireless networks," *Proceesings of IEEE MILCOM*, October 1999.

[8] J.G. Jetcheva and David B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, October 2001.

[9] A. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (cbt)," in *Proceedings of ACM SIGCOMM*, September 1993.

[10] L. Ji and M. S. Corson, "A lightweight adaptive multicast algorithm," in *Proceedings of IEEE GLOBECOM 1998*, December 1998, pp. 1036–1042.

[11] L. Ji and M.S. Corson, "Differential destination multicast - a manet multicast routing protocol for small groups," in *Proceedings of IEEE INFOCOM*, April 2001.

[12] P. Sinha, R. Sivakumar, and V. Bharghavan, "Mcedar: Multicast core extraction distributed ad-hoc routing," in *Proceedings of the Wireless Communications and Networking Conference, WCNC*, September 1999, pp. 1313–1317.

[13] C.K. Toh, G. Guichala, and S. Bunchua, "Abam: On-demand associativity-based multicast routing for ad hoc mobile networks," in *Proceedings of IEEE Vehicular Technology Conference, VTC 2000*, September 2000, pp. 987–993.

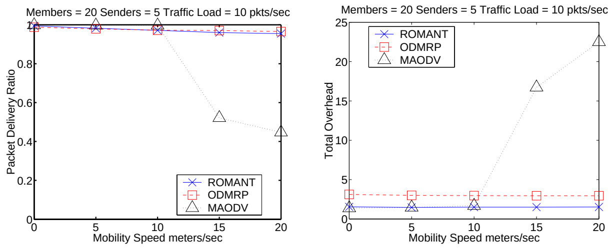
[14] S.K. Das, B.S. Manoj, and C.S. Ram Murthy, "A dynamic core based multicast routing protocol for ad hoc wireless networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002.

[15] V. Devarapalli and D. Sidhu, "Mzr: A multicast protocol for mobile ad hoc networks," in *ICC 2001 Proceedings*, 2001.

[16] S. Lee and C. Kim, "Neighbor supporting ad hoc multicast routing protocol," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, August 2000.

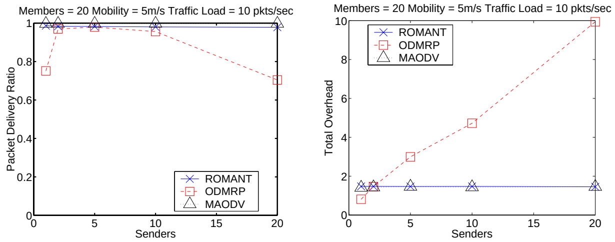
[17] C. Perkins and E. Royer, "Ad hoc on demand distance vector (aodv) routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

[18] E.M. Royer and C.E. Perkins., "Multicast ad hoc on demand distance vector (maodv) routing," *Internet-Draft, draft-ietf-draft-maodv-00.txt.*



(a) Packet Delivery Ratio Vs Mobility

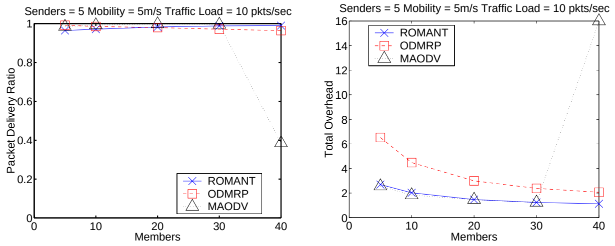
(b) Total Packet Overhead Vs Mobility



(c) Packet Delivery Ratio Vs Senders

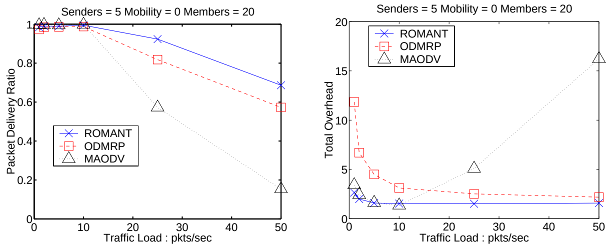
(d) Total packet Overhead Vs Senders

Fig. 5. Protocol comparison results for Mobility and Number of Senders



(a) Packet Delivery Ratio Vs Members

(b) Total Packet Overhead Vs Members



(c) Packet Delivery Ratio Vs Traffic Load

(d) Total packet Overhead Vs Traffic Load

Fig. 6. Protocol comparison results for Number of Members and Traffic Load