Efficient Hybrid Multicast Routing Protocol for Ad-Hoc Wireless Networks

Jayanta Biswas and Mukti Barai and S. K. Nandy
CAD Lab, Indian Institute of Science
Bangalore, 560 012, India
{jayanta@cadl, mbarai@cadl, nandy@serc} .iisc.ernet.in

Abstract

An ad hoc network is composed of mobile nodes without any infrastructure. Mobile nodes self organize themselves to form a network over radio links. The trend of applications of mobile ad hoc networks requires increased group oriented services. Hence multicast support is critical for ad hoc networks. As the number of participating nodes increase, scalability becomes an important issue. On Demand Multicast Routing Protocol (ODMRP) [1] provides high Packet Delivery Ratio in presence of high mobility. But ODMRP suffers from higher control overhead as the network size and the number of source nodes increase.

In this paper, we present an efficient hybrid multicast routing protocol suitable for high mobility applications and it addresses the scalability issue of ODMRP protocol. This protocol separates out data forwarding path from join-query *forwarding path. We incorporate low overhead local clustering technique to classify all nodes into* core *and* normal *categories. When multicast routes to destination nodes are unavailable,* join-query *messages are sent to all nodes in the network and data packets are forwarded by the* core *nodes to the destination nodes using* Differential Destination Multicast *[2]. Through simulations we show that this protocol reduces control overhead and increases packet delivery ratio by 20-50% for different network scenarios.* [1]

1. Introduction

Mobile nodes move arbitrarily in an ad hoc network and this changes the network topology in a frequent and unpredictable way. The recent trend of applications of mobile ad hoc networks requires increased group oriented services. Hence multicast support is critical for ad hoc networks. The design of a high performance multicasting scheme in presence of high mobility for these networks is a complex issue because of continuous change in network topology and limited channel bandwidth.

Many multicast routing schemes are proposed for ad hoc networks. They can broadly be classified into *tree based* or *mesh based* routing protocols. ODMRP [1], FGMP [4], CAMP [5] and DCMP [8] are example of *mesh based* routing protocols. MAODV [7], MCEDAR [6], AMRoute [9] are example of *tree based* routing protocols. Mesh based protocols perform better in high mobility situation as they provide redundant paths from source to destinations while forwarding data packets. However, *mesh based* approach sacrifices multicast efficiency in comparison to *tree based* approach.

Out of all currently available proposal for Multicast Routing in MANET environment, ODMRP outperforms other proposals in presence of high mobility. This is confirmed by Lee *et al.* [3]. However, as the number of nodes increase, scalability becomes an important issue. ODMRP suffers from scalability issues as the network size and the number of source nodes increase.

To overcome the scalability issue of ODMRP we propose Efficient Hybrid Multicast Routing Protocol for Ad-Hoc Wireless Networks. This is a modified version of ODMRP protocol. Here we separate the data packets forwarding path from *join-query* forwarding path when multicast routes to destination nodes are unavailable. Rest of the protocol works in the same way as ODMRP. We reduce the control overhead for large network size at high network load and solve the scalability issues of ODMRP. We incorporate Differential Destination Multicast and low overhead local clustering from MCEDAR [6] in the new proposal and hence the new proposal is termed as hybrid.

The remaining sections are organized as follows. In section 2, we describe Efficient Hybrid Multicast Routing Protocol (EHMRP) in detail. In section 3, we describe simulation model and methodology and present comparative performance analysis of the proposed protocol with ODMRP in section 4. We present the conclusion in section 5.

2. The proposed protocol

This protocol separates out data forwarding path from *join-query* forwarding path. We incorporate low overhead

---

[1] This work was partially supported out of a research grant with ST Microelectronics.

local clustering technique to classify all nodes into *core* and *normal* categories. When multicast routes to destination nodes are unavailable, *join-query* messages are sent to all nodes in the network and data packets are forwarded by the *core* nodes to the destination nodes using DDM. DDM is a stateless multicast approach where multicast tree information is appended with each data packet header. The key components of the proposed protocol are (a) classifying *core* and *normal* nodes, (b) separating out data forwarding path while sending *join query* request and sending data packets using DDM, (c) separate handling of received data packets coming through DDM path, (d) group membership update and (e) normal functionality of ODMRP protocol. The proposed protocol does not require any underlying unicast protocol.

2.1. Node Classification

We have used low overhead constant time core extraction algorithm from MCEDAR [6] protocol. We could have used any local clustering technique available in the literature. Core nodes are ideally the minimum domination set of the the network topology. But we use a constant time approximate algorithm which provides good result in the average case. This algorithm runs periodically at each node.

Each *normal* node selects another *core* node as its dominator. Each *normal* node forwards data packets to its dominator node. Effective degree of a node is the number of neighbors who have chosen the current node as their dominator node. Each node selects the highest degree node with maximum effective degree and lowest node id among one hop neighbors as its dominating node. When a node transits from *normal* to *core* node, it broadcasts a beacon in its third neighborhood containing node id, maximum hop count to travel and path traversed information. Intermediate nodes decrement hop count, update path traversed information and broadcast the message. Intermediate nodes also update their database with the source node and path traversed information.

Each node stores received beacons of its one hop neighbors. It also calculates and stores its degree, effective degree and its dominator node id. If a node dominates one or more nodes, it is termed as *core* node. Each *core* node stores the nodes it dominates, path of its nearby *core* nodes and dominated nodes of each *core* node in the third neighborhood. It also stores certain information for each node that it dominates. It includes all one hop neighbors and the dominator of each neighbor node. Each *core* node maintains sufficient information to reach nearby nodes and setup virtual connection. However, no *core* node is aware of the complete core graph. The complete algorithm is based on local computation and hence it is efficient.

If a node looses connection with its dominating node, it hears the beacon from its neighbors and selects a new dominator node.

2.2. Separation of data path from Join Query send path

When a source node wants to send data to a destination multicast group, it checks whether it has established active connections for the same multicast group. If it does not find any, it sends *join query* request for the destination multicast group. In our proposed protocol we separate out data forwarding path from *join query* sending path. Data packet is duplicated and *join query* is sent with IP packet headers but the data portion of the packet is stripped from the *join query* request. Data portion of the packet is sent to the members of that multicast group using DDM. Source node finds all the members of the destination multicast group from locally maintained database. With DDM, source node encodes all members of the destination multicast group and attaches with the data packet by means of a new IP option field defined by this protocol. This packet is sent to the next hop receiver nodes in the same multicast group using single hop broadcast.

2.3. Receiving data packets from DDM path

EHMRP checks for presence of special IP option field defined by EHMRP in each incoming data packet header. If it finds the special IP option field in the data packet, it implies that the data packet has traveled through DDM path and it has DDM header associated with the packet. Each receiving node checks DDM header in the packet whether it is a next hop node. If the receiving node is not a next hop node, it drops the received packet. Otherwise it retrieves the destination multicast group address from the data packet. If the current node is member of the same multicast group, a copy of the data packet is created and sent to the corresponding application after stripping of IP header. It extracts its corresponding DDM header from the received packet and updates the data packet with the new DDM header and forwards the data packet to the next hop destination nodes using single hop broadcast.

2.4. Group Membership Update

When a node joins or leaves the group, it sends broadcast message with group membership update information and inserts the source node and sequence number in the message cache. The required information of a message transmitted by the node is stored in the message cache.

When a node receives a group membership update message, it checks whether the message is present in the message cache. If the message is not present in the message cache, it broadcasts the message, updates its group membership database and inserts the source node and sequence number of the received packet in the message cache.

2.5. ODMRP protocol functionality

ODMRP forwards multicast packets using mesh based forwarding group concept. It is an on demand protocol and uses soft state approach in maintaining group membership. Group membership and multicast routes are updated by the source on demand. When multicast sources have data to send but no route to the multicast group, it broadcasts *join query* message to the entire network periodically to refresh the membership information and multicast routes. When an intermediate node receives *join query* message, it stores source address and sequence number in the message cache to prevent duplicate processing of the received packet. The routing table is updated with source node id. If the received packet is not a duplicate and the *time to live* is greater than zero, node broadcasts the *join query* message again.

Once *join query* message reaches the node which belongs to the same multicast group, the node broadcasts *join reply* message. On receipt of *join reply* message, each node checks whether the next hop address matches its own id. If it matches, this node sets the forwarding group flag for the multicast group. It then rebroadcasts *join reply* message based on the match in its routing table. Thus join reply message gets propagated to the multicast source *via* the shortest path. This process creates multicast routes from source to receivers.

After setting up forwarding nodes, source can multicast packets to receivers. When a node receives data packet, it forwards the packet if it is a forwarding node and the packet is not a duplicate.

3. Performance Evaluation

This section presents the simulation results of the proposed algorithm. We compare the results of EHMRP with ODMRP multicast protocol. The metrics are packet delivery ratio, number of data packets transmitted per data packet received, number of control bytes transmitted per data byte received and number of total packets transmitted per data packet received.

3.1. Experimental Setup

We develop a simulator with Glomosim library [10] for comparing the performance of ODMRP and EHMRP. We use ODMRP code which is freely available with Glomosim 2.03 release. We fix bugs in *join reply* handling and doubly linked list deletion in ODMRP code and use for comparison purpose with our proposed protocol. Mobility prediction is not present in this ODMRP implementation. ODMRP parameter values used are shown in Table I.

We are using a basic ODMRP code. ODMRP and EHMRP both use same data flow path after destination forwarding nodes are established by *join query* and *join reply* exchange. The improved version of ODMRP incorporates reliable packet delivery with passive acknowledgment and setting up of alternate routes with mobility prediction using GPS. If the improved version of ODMRP code is used, it would enhance ODMRP performance as well as EHMRP performance. As we are interested in relative ranking of these two protocols with respect to scalability issues, use of basic ODMRP code is not an issue.

Parameter	*Value*
JOIN DATA refresh interval	3 sec
Acknowledgment timeout for JOIN TABLE	25 msec
Maximum JOIN TABLE retransmission	3

Table 1. Parameter Values for ODMRP.

Our simulation modeled a network of 50 mobile hosts placed randomly within a 1000 m × 1000 m. Radio transmission range used is 250 meters and channel capacity is 2 Mbps. A free space propagation model with a threshold cutoff is used in our experiments. All nodes share the same physical channel.

The IEEE 802.11 MAC with Distributed Coordinated Function (DCF) is used as the MAC protocol. We use default MAC code from Glomosim distribution.

There is no network partition throughout the simulation and each simulation is conducted for 600 seconds. Multiple runs are taken with different seed values for each scenario and collected data are averaged over those runs.

3.2. Traffic Pattern

CBR source: Constant bit rate traffic with application data size of 512 bytes is generated. Rate of the traffic is an adjustable parameter which is varied for different scenarios. The senders are chosen randomly from group members and group members are chosen randomly among all nodes in the network. The member nodes join the multicast group at the start of the simulation and continue to be a member till the end of the simulation. Start time for data transmission for each source starts with a random time uniformly distributed over [5, 25] sec. Data is transmitted till the end of the simulation time.

4. Simulation Results

We conduct several experiments to show the performance of our proposal for different type of network configurations.

4.1. Mobility Speed

In this experiment, each mobile node stays stationary for 10 seconds and starts moving to a randomly selected lo-

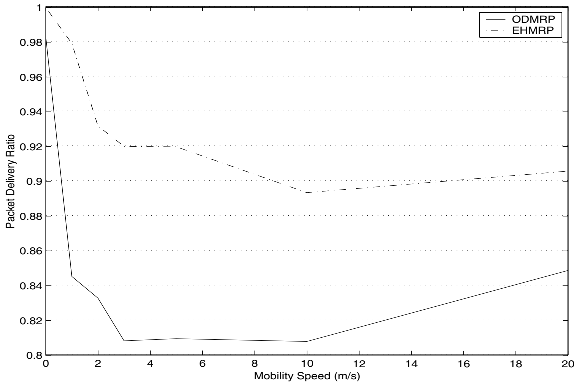

Figure 1. Packet delivery ratio for various mobility speed.

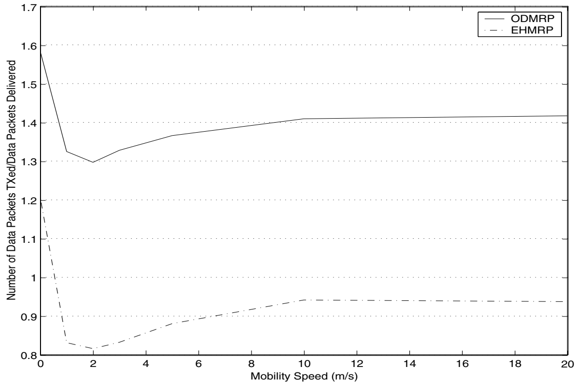

Figure 2. Number of data packets transmitted per data packet delivered for various mobility speed.

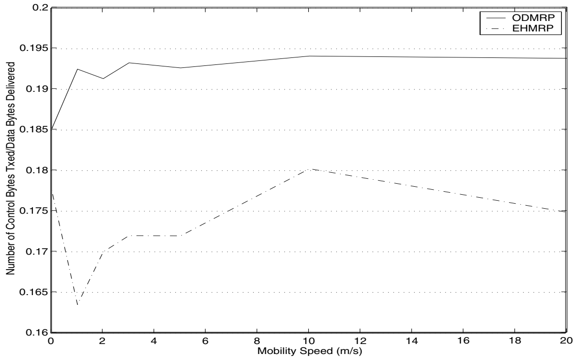

Figure 3. Number of control bytes transmitted per data byte delivered for various mobility speed.

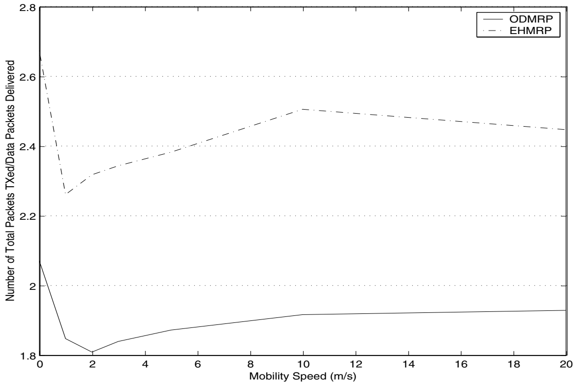

Figure 4. Number of all packets transmitted per data packet delivered for various mobility speed.

cation with a random speed uniformly distributed over [0, maxvalue] m/sec. Value of maxvalue is varied from 0 to 20 m/sec. 20 nodes are configured as multicast members of the same group and 5 sources transmit packets at 2 pkt/sec each.

Fig. 1 depicts packet delivery ratio of the two protocols under different mobility speeds. EHMRP outperforms ODMRP as the mobility speed increases by a decent margin of 10%. EHMRP maintains low overhead local cluster and uses limited broadcast while sending data packets using DDM path. ODMRP floods data packets while sending *join query* request and this results in more collision of the shared channel. EHMRP is more robust than ODMRP in handling node mobility.

Fig. 2 shows the number of data packets transmitted per data packet delivered for the two protocols. This ratio drops to a smaller value under low mobility conditions and then slightly increases with high mobility. EHMRP is more robust than ODMRP in high mobility situation. EHMRP uses DDM to forward data packets while sending *join query* requests and uses one hop limited broadcast. ODMRP uses mesh forwarding with duplicate checks. So redundancy in data forwarding is more in ODMRP than EHMRP.

Fig. 3 shows number of control bytes transmitted per data bytes delivered. As EHMRP uses DDM and one hop limited broadcast, it outperforms ODMRP based on this parameter. With high mobility the overhead for clustering algorithm used in EHMRP increases and so the value of this ratio also increases for EHMRP.

Fig. 4 shows total number of packets transmitted per data packet delivered for the two protocols. The value of this ratio drops under low node mobility and it increases slowly with high node mobility. As ODMRP is an *on demand* protocol, nodes running ODMRP do not transmit control packets until node has data to send. But EHMRP transmits control packets periodically to maintain the local clustering

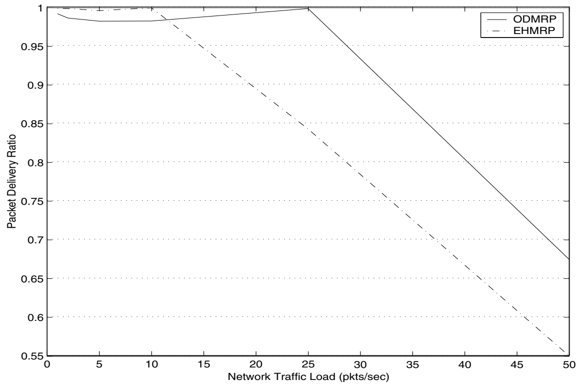

Figure 5. Packet delivery ratio for various network traffic load.

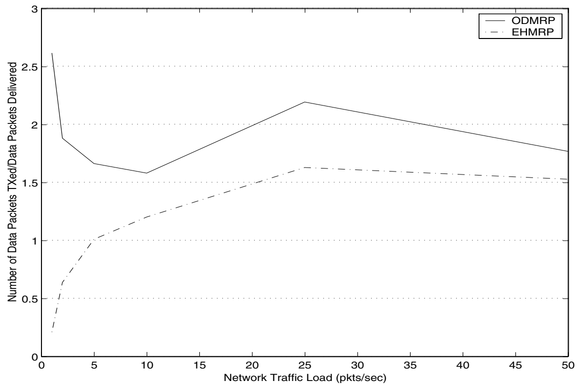

Figure 6. Number of data packets transmitted per data packet delivered for various network traffic load.

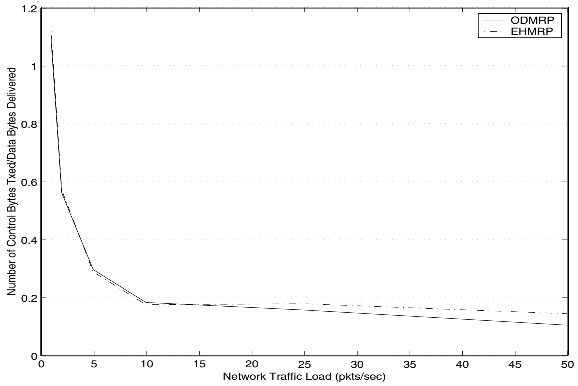

Figure 7. Number of control bytes transmitted per data byte delivered for various network traffic load.

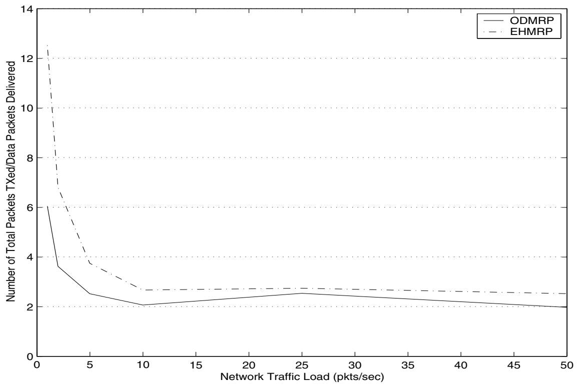

Figure 8. Number of all packets transmitted per data packet delivered for various network traffic load.

data. Hence the value of this ratio is 50% more for EHMRP than ODMRP but is within acceptable limit for EHMRP.

From the above results we conclude that EHMRP is more robust than ODMRP in high mobility situations.

4.2. Network Traffic Load

In this experiment the impact of network traffic load on the multicast routing protocols is studied. Each node stays stationary and 20 nodes are configured as multicast members of the same group. 5 sources transmit packets and total network traffic load is varied from 1 pkt/sec to 50 pkt/sec.

Fig. 5 illustrates packet delivery ratio of the two protocols under different network load scenarios. Both the protocol show similar trend in packet delivery ratio as the network load increases but ODMRP performs better. EHMRP performs slightly better than ODMRP till network load remains below 10 pkt/sec but ODMRP outperforms EHMRP by 15% under high network load. EHMRP uses DDM for data forwarding and the redundancy is low. With higher network load some data packets are dropped and as data replication information is contained with the data packet, next hop nodes do not receive some data packets to replicate. This leads to the decrease of packet delivery ratio for EHMRP at high network traffic load.

Fig. 6 depicts number of data packets transmitted per data packet delivered for the two protocols. The value of this ratio is lower for EHMRP than ODMRP. EHMRP uses DDM for data forwarding while sending *join query* packets. ODMRP uses mesh forwarding while sending *join query* packets.

Fig. 7 shows number of control bytes transmitted per data bytes delivered. As the network load increases the value of this ratio decreases. This ratio decreases slightly once network load reaches 10 pkt/sec value. Performance

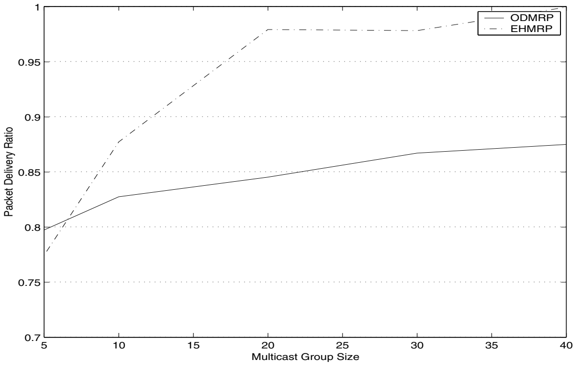

Figure 9. Packet delivery ratio for various multicast group size.

of both the protocols are very similar till network load stays below 15 pkt/sec. With higher network load ODMRP performs little better than EHMRP. As ODMRP uses mesh forwarding, it performs better in forwarding data packets in *join query* path. As EHMRP uses DDM for forwarding data packets, some nodes become bottleneck at high network traffic load.

Fig. 8 depicts total number of packets transmitted per data packet delivered for the two protocols. Both the protocols shows similar trends for this ratio. At low network load the value of this ratio is more for EHMRP because *join query* packets does not carry data and is small in size. So more number of *join query* broadcasts are successful at low traffic load for EHMRP. As EHMRP transmits control packet periodically to maintain its local clustering data and ODMRP is an *on demand* protocol, the value of this ratio is slightly higher for EHMRP than ODMRP for high network traffic load.

4.3. Multicast Group size

To study the impact of various multicast group sizes on multicast routing protocols, we varied multicast group sizes. Multicast group size is varied from 5 to 40 members. 5 sources transmit packets and total network traffic load is fixed at 10 pkt/sec. Each node stays stationary for 10 seconds and starts moving to a randomly selected location with a random speed uniformly distributed over [0, 1] m/sec.

Fig. 9 illustrates packet delivery ratio of the two protocols with different multicast group sizes. Packet delivery ratio increases for both the protocols as multicast group size increases. This is because *join reply* packets are lost less often with the increase in multicast group size. This is true for both ODMRP and EHMRP as this flow path is same for both the protocols. EHMRP sends *join query* without the data packet and so the size of *join query* packet is much less than ODMRP. Hence EHMRP outperforms ODMRP as multicast group size increases. As this metric is a measure of scalability of the protocol, EHMRP is more scalable than ODMRP.

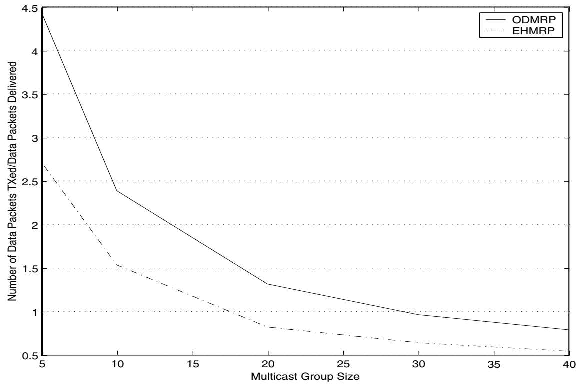

Figure 10. Number of data packets transmitted per data packet delivered for various multicast group size.

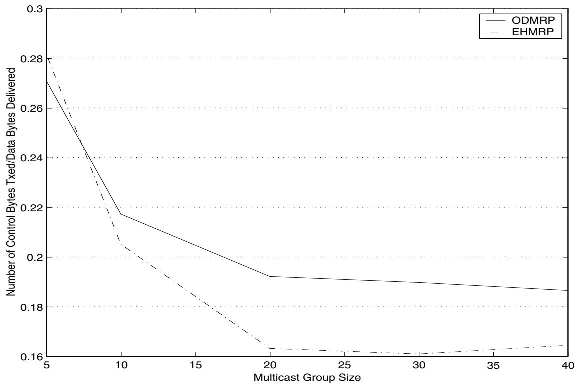

Figure 11. Number of control bytes transmitted per data byte delivered for various multicast group size.

Fig. 10 shows number of data packets transmitted per data packet delivered for the two protocols. Both the protocol show similar trends with the increase in multicast group size but EHMRP performs slightly better. The value of this ratio decreases as the multicast group size increases. This is because *join reply* packets are lost less often with the increase in multicast group size. So less data packets are forwarded through *join query* path.

Fig. 11 depicts number of control bytes transmitted per data bytes delivered. Both the protocol show similar trends. The value of this ratio decreases as *join reply* packets are lost less often with the increase in multicast group size. EHMRP outperforms ODMRP by around 15% based on this parameter value.

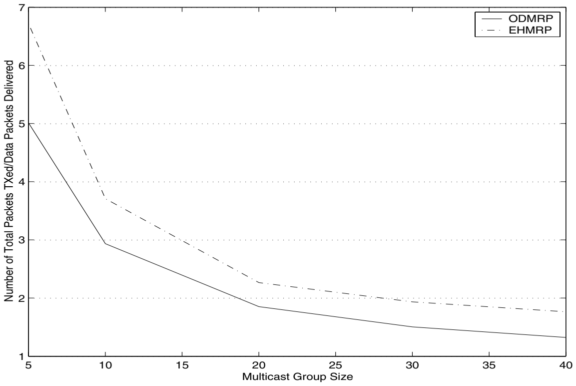

Figure 12. Number of all packets transmitted per data packet delivered for various multicast group size.

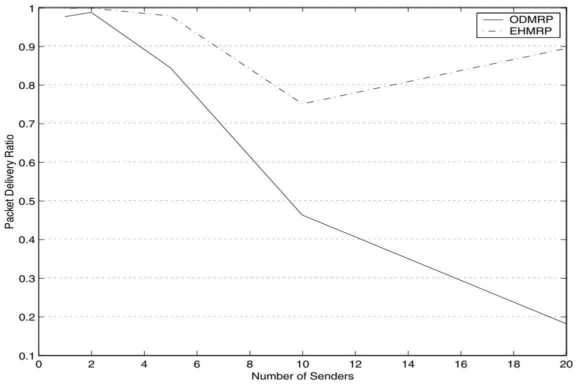

Figure 13. Packet delivery ratio for various number of senders.

Fig. 12 shows number of all packets transmitted per data packet delivered for the two protocols. Both the protocol show similar trends with the increase in multicast group size but EHMRP performs slightly better. The value of this ratio decreases as *join reply* packets are lost less often with the increase in multicast group size.

4.4. Number of Senders

In this scenario, we studied the impact of number of senders on multicast routing protocols. This gives a measure of scalability of the protocol. 20 nodes are configured as multicast members of the same group and total network traffic load is fixed at 10 pkt/sec. Node mobility is kept at a low value. Each node stays stationary for 10 seconds and starts moving to a randomly selected location with a random speed uniformly distributed over [0, 1] m/sec. Number of senders is varied from 1 to 20.

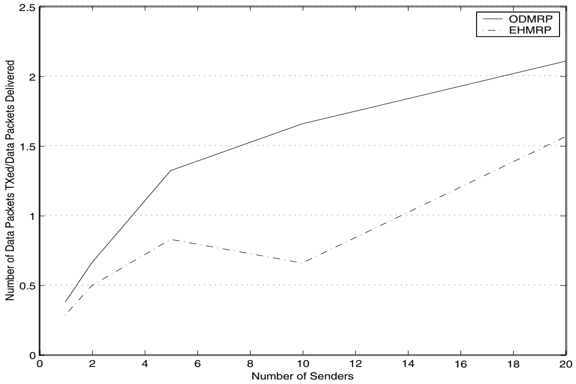

Figure 14. Number of data packets transmitted per data packet delivered for various number of senders.

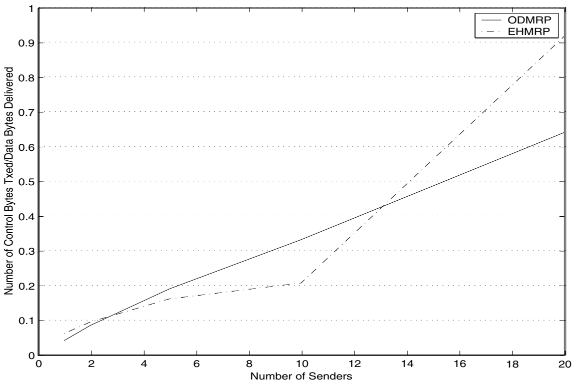

Figure 15. Number of control bytes transmitted per data byte delivered for various number of senders.

Fig. 13 illustrates packet delivery ratio of the two protocols with different number of senders. The value of this ratio decreases for both the protocols until the number of senders remains within a value of 10. Number of *join query* and *join reply* packets flowing in the network increases as the number of sender increases. This leads to more contention of the shared channel and some nodes start dropping *join reply* packet. Hence the value of this ratio decreases as the number of sender increases. As the number of senders increases beyond 10, ODMRP follows the same decreasing trend in performance.

For EHMRP with 20 senders, join replies are lost because of the *join query* flood storm but data packets are delivered to destination nodes efficiently through DDM path. EHMRP is more scalable than ODMRP in this regard.

Fig. 14 depicts number of data packets transmitted per data packet delivered for the two protocols. Both the proto-

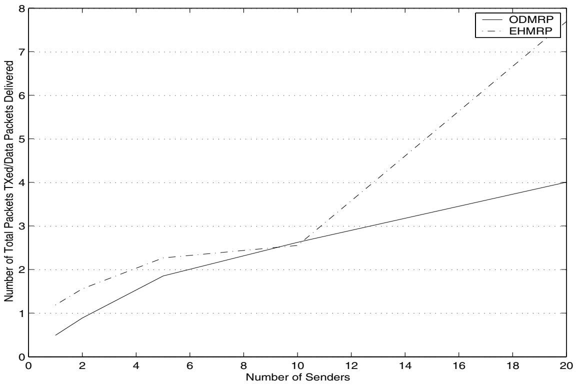

Figure 16. Number of all packets transmitted per data packet delivered for various number of senders.

cols show similar trend as the number of senders increase but EHMRP performs better. Number of *join query* and *join reply* packets flowing in the network increases as the number of sender increases. This leads to more contention of the shared channel and some nodes drop data packets. Hence this ratio increases with the increase in number of senders.

Fig. 15 shows number of control bytes transmitted per data bytes delivered. Both the protocols show similar trends as the number of senders increase. Number of *join query* and *join reply* packets flowing in the network increases as the number of senders increases. This leads to more contention of the shared channel and some nodes drop data packets and *join reply* packets. This leads to the increase of *join query* transmissions and hence the increase of this ratio value. For large number of senders delivery of data packets are reduced very much for ODMRP. For EHMRP data packets are delivered through DDM path while sending *join query* request. In this data delivery path, EHMRP incurs more data overhead as DDM headers are included with each data packet. The value of this ratio is slightly less for ODMRP than EHMRP for large number of senders.

Fig. 16 shows number of all packets transmitted per data packet delivered for the two protocols. The value of this ratio increases for both the protocols with the increase in number of senders. Number of *join query* and *join reply* packets flowing in the network increases as the number of sender increases. This leads to more contention of the shared channel and some nodes drop data packets and *join reply* packets. This leads to the increase of *join query* transmissions. So the value of this ratio increases with the increase in number of senders.

We observe ODMRP performance drops down sharply for large number of senders. We observe EHMRP outperforms ODMRP for large number of senders.

5. Conclusion

We have proposed an efficient hybrid mesh based multicast protocol for ad hoc networks. The key concept is to separate data forwarding path from *join-query* forwarding path by incorporating low overhead local clustering technique and forwarding data packets using DDM to solve the scalability issues of ODMRP. This protocol solves scalability issues of ODMRP by reducing control overhead and increasing multicast efficiency for different network scenarios. Through simulations we show packet delivery ratio is also improved by 20-50% for different network scenarios. The authors in [3] have shown that ODMRP is the best performing multicast routing proposal in the literature. This proposal solves scalability issues of ODMRP and outperforms all other multicast routing proposals in the literature including ODMRP for high mobility applications.

References

[1] S.J. Lee, M. Gerla, and C. C. Chiang, “On Demand Multicast Routing Protocol”, *In Proceedings of IEEE WCNC’99*, pp. 1298-1302, September 1999.

[2] L. Ji, and M.S. Corson, “Differential Destination Multicast - A MANET Multicast Routing Protocol for Small Groups”, *In Proceedings IEEE/INFOCOM’01*, April, 2001.

[3] S. Lee, W. Su, J. Hsu, M. Gerla, and R. Brigodia, “A Performance Comparison Study of Ad hoc Wireless Multicast Protocols”, *In Proceedings IEEE/INFOCOM’00*, March 2000.

[4] C. C. Chiang, M. Gerla, and L. Zhang, “Forwarding Group Multicasting Protocol for Multihop, Mobile Wireless Networks”, *ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing*, vol. 1, no. 2, pp. 187-196, 1998.

[5] J.J. Garcia-Luna-Aceves, and E. L. Madruga, “The Core-Assisted Mesh Protocol”, *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1380-94, August 1999.

[6] P. Sinha, R. Sivakumar, and V. Bharghavan, “MCEDAR: Multicast Core Extraction Distributed Ad hoc Routing”, *In Proceedings of IEEE WCNC’99*, pp. 1313-1317, September 1999.

[7] E. M. Royer, and C. E. Perkings, “Multicast Operations of the Ad-hoc On-Demand Distance Vector Routing Protocol”, *In Proceedings of ACM/MOBICOMM*, August 1999.

[8] S.K. Das, B.S. Manoj, and C.S.R. Murthy, “A Dynamic Core Based Multicast Routing Protocol for Ad hoc Wireless networks”, *In Proceedings of ACM/MOBIHOC*, June 2002.

[9] E. Bommaiah, M. Liu, A. MvAuley, and R. Talpade, “AM-Route: Ad hoc Multicast Routing Protocol”, Internet Draft, draft-manet-amroute-00.txt.

[10] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, “GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems”, *http://pcl.cs.ucla.edu/projects/domains/glomosim.html.*