A Dynamic Core Based Multicast Routing Protocol for Ad hoc Wireless Networks *

Subir Kumar Das
das@rts.iitm.ernet.in

B. S. Manoj
bsmanoj@cs.iitm.ernet.in

C. Siva Ram Murthy †
murthy@iitm.ernet.in

Computer Science & Engineering Dept.
Indian Institute of Technology
Madras, India

ABSTRACT

Ad hoc wireless networks are self-organizing, dynamic topology networks formed by a collection of mobile nodes through radio links. Minimal configuration, absence of infrastructure, and quick deployment, make them convenient for emergency situations other than military applications. Multicasting plays a very crucial role in the application of Ad hoc networks. As the number of participants increases, scalability of the multicast protocol becomes an important issue. Among the existing multicast protocols, On Demand Multicast Routing Protocol (ODMRP) [1], exhibits a high packet delivery ratio even at high mobility. But, ODMRP suffers from higher control overhead as the network size and the number of sources increase.

In this paper, we propose an efficient multicast routing protocol for Ad hoc wireless networks. This protocol reduces the control overhead by dynamically classifying the sources into *Active* and *Passive* categories. The control overhead is significantly reduced by about 30% compared to ODMRP, which contributes to the scalability of the protocol. We study the effectiveness of the proposed multicast routing protocol by simulation studies and the results show that the multicast efficiency is increased by 10-15% and packet delivery ratio is also improved at high network load.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols — Routing Protocols

General Terms

Design, Performance

Keywords

Ad hoc Wireless Networks, Multicast Routing,
Dynamic Core, Performance Evaluation

1. INTRODUCTION

Due to the rapid development in the mobile devices technology, wireless networks are becoming more popular. Wireless networks can be broadly classified into two types – infrastructure-based networks (for e.g., cellular networks) and Ad hoc networks. The former one uses fixed base stations, which are responsible for co-ordinating the communication between the mobile hosts (nodes). These base stations are interconnected by wired backbones, whereas, mobile nodes communicate with the base stations through the wireless medium. The latter one consists of mobile nodes that communicate with each other through the wireless medium, without any fixed infrastructure. Hence, there is no centralized mechanism to control the communication among the group of mobile nodes. As bandwidth is a scarce resource, efficient utilization of bandwidth is the most important issue in the Ad hoc environment. Since host mobility causes frequent and unpredictable topological changes, finding and maintaining routes in Ad hoc networks is a nontrivial task. Other issues such as the hidden terminal(s) effect [2] and the broadcast nature of the radio channel make routing in Ad hoc networks much more complex compared to that in wired networks.

Many routing protocols have been proposed for Ad hoc networks. They can be broadly classified into *table-driven* and *source-initiated on-demand* routing protocols. In table-driven routing protocols, each node maintains the routing information, and updates the same at regular intervals. Some existing table-driven routing protocols are Destination-Sequenced Distance-Vector (DSDV) routing [3], Clusterhead Gateway Switch Routing (CGSR) [4], and Wireless Routing Protocol (WRP) [5]. For updating routing tables, each node transmits control packets periodically, which constitute an inefficient use of network capacity. On the other hand, in source-initiated on-demand routing protocols, routes are obtained only when desired by the source. These protocols save bandwidth by avoiding periodic transmissions of control packets, but at the cost of augmented connection setup time. Some existing source-initiated on-demand routing protocols are Ad hoc On demand Distance Vector (AODV) routing [6], Dynamic Source Routing (DSR) [7], Tempo-

---

[*] This work was supported by the Department of Science and Technology, New Delhi, India.

[†] Author for correspondence.

MOBIHOC'02, June 9-11, 2002, EPFL Lausanne, Switzerland.

rally Ordered Routing Algorithm (TORA) [8], Associativity Based Routing (ABR) [9], and Signal Stability based Adaptive (SSA) [10] routing. Although most of the routing protocols are based on either the table-driven approach or the source-initiated on-demand approach, an attempt is made to combine the best of both in [11]. In Zone Routing Protocol (ZRP) [11], each node is associated with a routing zone. Within its routing zone, the node maintains the topology information by using a table-driven routing protocol, and out of its routing zone, a source-initiated on-demand routing protocol is used.

Ad hoc wireless networks find applications in civilian operations (collaborative and distributed computing), emergency search-and-rescue, law enforcement, and warfare situations, where setting up and maintaining a communication infrastructure is very difficult. In all these applications, communication and co-ordination among a given set of nodes is necessary. Multicast routing protocols play an important role in Ad hoc wireless networks to provide this communication. It is always advantageous to use multicast rather than multiple unicast, especially in Ad hoc environment, where bandwidth comes at a premium. Many multicast routing protocols for Ad hoc networks exist in the literature. Their fundamental differences lie in the approach used for initializing and maintaining the multicast group.

The rest of the paper is organized as follows. In Section 2, we review some commonly used terms, and give a brief survey of the related work. In Section 3, we provide the motivation for our work. In Section 4, we describe our multicast routing protocol. We present numerical results from the simulation studies of our multicast routing protocol in Section 5. Finally, we make some concluding remarks in Section 6.

2. RELATED WORK

Multicasting consists of concurrently sending the same message from one source to multiple destinations. It plays an important role in video-conferencing, distance education, co-operative work, video on demand, replicated database updating and querying, etc. Several multicast routing protocols have been proposed for Ad hoc networks, which are classified as either *mesh based* or *tree based*. In a mesh based multicast protocol, there may be more than one path between a pair of source and receiver, thus providing more robustness compared to tree based multicast protocols. In a tree based multicast protocol, there is only a single path between a pair of source and receiver, thus leading to higher multicast efficiency. The construction of a multicast tree can be done either from the source (*source-initiated*) or from a receiver (*receiver-initiated*).

The Ad hoc environment suffers from frequent path breaks due to mobility of nodes, hence an efficient multicast group maintenance is necessary. Maintaining the multicast group can be done by either *soft state approach* or *hard state approach*. In the soft state approach, the multicast group membership and associated routes are refreshed periodically which necessitate flooding of control packets. But, in the hard state approach, the routes are reconfigured only when a link breaks, thus making it a reactive scheme.

Some examples of tree based multicast protocols are Ad hoc Multicast Routing (AMRoute) [12], Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) [13], Bandwidth Efficient Multicast Protocol [14], Multicast operation of the Ad hoc On demand Distance Vector (MAODV) routing protocol [15], and Multicast Core-Extraction Distributed Ad hoc Routing (MCEDAR) protocol [16].

AMRoute [12] assumes the existence of a unicast routing protocol in the network environment but it is independent of a specific unicast routing protocol. This protocol has two main phases - mesh creation and tree creation. After formation of mesh by the logical core, it periodically creates a virtual multicast tree over the mesh. This multicast tree uses unicast tunnels to connect group members. Due to the underlying mesh, there is no need for frequent tree readjustments, thus providing robustness in a high mobility environment.

AMRIS [13] is an on-demand, source-initiated, shared tree based multicast protocol. In this protocol, each node in a multicast session generates session-specific *multicast session member id* (msm-id), after receiving the *NEW-SESSION* message from its parent node. The NEW-SESSION message transmission is initiated by a special node called *Sid*, at which the shared tree is rooted. The msm-id increases from the root towards leaf nodes radially, which indicates the flow of multicast data. The protocol uses periodic, short broadcast beacon packets to determine whether a link has been broken. Upon link break, it executes a *branch reconstruction* process to maintain the multicast tree.

Unlike soft state multicast protocols, the Bandwidth Efficient Multicast routing protocol [14] uses a hard state approach *i.e.,* there is no periodic transmission of control messages. Nodes join the multicast group through the nearest forwarding node, thus minimizing the number of added forwarding nodes and eventually leading to a high multicast efficiency. In [15], an on-demand multicast protocol has been proposed, which is the multicast extension of AODV [6]. MCEDAR [16] is the multicast extension of the Core-Extraction Distributed Ad hoc Routing [17] protocol. To provide robustness and efficiency, it uses an underlying mesh over which it constructs a forwarding tree.

Some of the existing mesh based multicast protocols are On Demand Multicast Routing Protocol (ODMRP) [18], Forwarded Group Multicast Protocol (FGMP) [19, 20], Core-Assisted Mesh Protocol (CAMP) [21], Neighbor Supporting Ad hoc Multicast routing Protocol (NSMP) [22], and Location-Based Multicast Protocols [23].

In contrast to the tree based concept, mesh based multicast protocols may have multiple paths between any source and receiver pairs, thus providing richer connectivity among the multicast members. The ODMRP [18] protocol is a mesh based protocol which uses a forwarding group concept for multicast packet delivery. Only the members of forwarding group forward data packets. For maintaining the multicast mesh it uses soft state approach.

Like ODMRP, FGMP [20] is also based on the forwarding group concept. But the major difference between them is that the former one is a source-initiated multicast protocol, while the latter one is receiver-initiated multicast protocol. Both FGMP and ODMRP protocols use control packets flooding to form the multicast mesh, thus resulting in considerable control overhead.

To eliminate flooding of control packets, CAMP [21] uses core nodes in the mesh. This protocol expands the idea of *core based tree*, [24] to form the mesh. But unlike the core based tree protocol, it contains more than one core. When any node wants to join the multicast group, it sends a *join Request* to a core node if none of its neighbor nodes are present in that particular multicast group. If all core nodes are unreachable, it uses an *expanded ring search* method to reach any group member. In contrast to ODMRP and FGMP, CAMP depends on the underlying unicast routing protocol.

In NSMP [22] protocol, maintenance of the mesh is done by *local route discovery i.e.,* when a source floods control information to refresh the route, it is forwarded only by mesh nodes and neighboring nodes (which are one hop away from any mesh node).

In Location-Based Multicast protocol [23], location information is used to reduce the control overhead. To deliver the data packets to all of the nodes in the same geographical region (it is called as *member region*), a limited flooding approach is used in this protocol. Before forwarding the data packets, a source defines a *forwarding zone.* A node forwards the data packets if it belongs to the forwarding zone.

3. MOTIVATION

We have mentioned various multicast routing protocols for Ad hoc environment in the previous section. Of these, ODMRP exhibits a high packet delivery ratio even at high mobility. In ODMRP [18], a source node which wants to initiate a multicast session, floods the *JoinReq* control packets to discover routes. When the receiver node (which wants to join the multicast group) receives this *JoinReq* packet, it builds a *JoinReply* packet and broadcasts it. The *JoinReply* packets are subsequently forwarded by the intermediate nodes along the reverse path to the source, thus establishing the route. A soft state approach is used to maintain the multicast group. This is done by periodically flooding the *JoinReq* control packets. Hence it provides robustness at the expense of increased control overhead, especially when the number of sources is large, as shown in [1]. In this paper, we propose an on-demand multicast routing protocol called Dynamic Core based Multicast routing Protocol (DCMP), which builds and maintains a *shared mesh*, *i.e.*, a mesh which is formed by a group of *core based trees* [24]. By exploiting the advantage of core based trees, it improves the scalability of the protocol. This is the motivation for our work.

Another existing multicast protocol is CAMP [21] which also builds and maintains a shared mesh. But the major difference is that DCMP is a source-initiated multicast protocol, whereas CAMP is a receiver-initiated multicast protocol. Unlike CAMP, our proposed multicast protocol, DCMP, is independent of any unicast routing protocol.

4. PROTOCOL DESCRIPTION

In DCMP the sources are classified into three categories

- *Active Sources*
- *Core Active Sources,* and
- *Passive Sources.*

Active sources are similar to sources in ODMRP which flood *JoinReq* control packets at regular intervals. Core Active sources are those Active sources which act as core for one or more *Passive sources.* In this paper,we have used the terms Core Active source and core node interchangeably. These core nodes are dynamic in nature and they are responsible for creating a shared mesh on behalf of the Passive sources which are associated with them. A Passive source does not transmit *JoinReq* control packet for creation of multicast mesh. A Passive source depends on a nearby Active source for forwarding its data packets. The maximum number of Passive sources that can be supported by a Core Active source is limited by a parameter called *MaxPassSize.* The hop distance between a Core Active source and a Passive source is bounded by the *MaxHop* parameter. These parameters basically discourage a large number of Active sources becoming Passive sources in the mesh. These help in maintaining the robustness of the mesh. This is because, mesh robustness basically depends upon the number of Active sources in the multicast group. As this number increases, robustness of the mesh increases, due to the transmission of the *JoinReq* packets periodically, by each Active source.

Multicast Group Address			
Passive Source ID	Next Node ID	CoreReq	Core Node ID
...	...	...	...

Figure 1: Format of PassReq packet

Multicast Group Address	Passive Source ID	PassiveSourceExistence timer
...	...	...

Figure 2: Format of PassSourceAddr table

In our DCMP protocol, when the source has data to send, it floods the *JoinReq* control packet. The *JoinReq* packet also contains an additional flag called as *CoreAcceptance flag*, the use of which is as follows. A core node may or may not be able to support more Passive sources due to the MaxPassSize parameter restriction. The *CoreAcceptance flag* is reset in the *JoinReq* packet if it (core node) cannot support more Passive source nodes. By checking this flag, near by Active source nodes come to know whether this core node can support them or not. This prevents unwarranted requests by nearby source nodes (which desire to change from Active to Passive) to the core node.

When any node receives a non-duplicate *JoinReq* control packet, it broadcasts the packet after storing the upstream node identification number (ID). A unique identification number is assumed for every node. When the *JoinReq* control packet is received by the receiver, it builds a *Reply* packet and sends it along the reverse path to the source. When intermediate nodes along the reverse path receive this *Reply* packet, they check the *Next Node ID* field in the *Reply* packet. If the node's ID matches with any of the entries of the field, then it sets its forwarding flag (FgFlag) and becomes a forwarding node for that particular multicast group. This node then builds a *Reply* packet and broadcasts it. In this way a route is established by the transmission of the *JoinReq* and the *Reply* packets.

When an Active source in a multicast group receives a *JoinReq* packet, it changes its status to a Passive source if all the following conditions are satisfied:

Multicast Group Address	Core Node ID	Passive Source ID

Figure 3: Format of Confirm packet

1. The *CoreAcceptance flag* is set.
2. Hop distance traveled by *JoinReq* is less than or equal to MaxHop.
3. The node ID of the source which receives a *JoinReq* packet (hereafter called as *ToBePassive* source) is less than the node ID of the source which sent the *JoinReq* packet (hereafter called as *ToBeCore* source).

If all the above conditions are met, then *ToBePassive* source sends a *PassReq* packet to the *ToBeCore* node, after setting the *CoreReq* field and putting its own ID in the *Passive Source ID* field of the packet. The format of *PassReq* control packet is shown in Figure 1. The setting of CoreReq field indicates the eagerness of *ToBePassive* source to turn from Active to Passive, if *ToBeCore* source node is willing to become core node for this *ToBePassive* source node.

After sending this *PassReq* packet, the *ToBePassive* source node prevents itself from either becoming core node for other source nodes or sending *PassReq* to other source nodes, by setting a lock flag. It then starts a *ConfirmWait* timer and waits for the *Confirm* packet from the *ToBeCore* source node.

When *PassReq* packet is received by an intermediate node, it stores the downstream node address in its *ConfirmRouteFind* table and then forwards the *PassReq* packet. This new entry made in the *ConfirmRouteFind* table is used to route back the *Confirm* packet to the *ToBePassive* source, when the intermediate node receives the same. But this entry is deleted from the *ConfirmRouteFind* table if the intermediate node does not get a *Confirm* packet within *ConfirmRouteDelete* time period.

When a *ToBeCore* source node receives this *PassReq* packet, it checks its *PassiveSupported* entry (*PassiveSupported* counts of the number of Passive sources being currently supported by the core node). If it is lesser than MaxPassSize, it accepts the request from the *ToBePassive* node by sending a *Confirm* packet to the *ToBePassive* source. Then, it increments the *PassiveSupported* counter and appends an entry for the *ToBePassive* source in the *PassSourceAddr* table. Thus, it becomes a forwarding node for the *ToBePassive* source node, *i.e.,* all the data packets from the *ToBePassive* source node will now be forwarded by this core node. The format of *PassSourceAddr* table and *Confirm* packet are shown in Figures 2 and 3, respectively.

After the above steps, if *PassiveSupported* counter at the core node is equal to MaxPassSize, then the flooding of the *JoinReq* packet from this node is done with the *CoreAcceptance flag* reset until the counter becomes less than MaxPassSize. When an intermediate node receives a *Confirm* packet, it sets its FgFlag and becomes a forwarding node. It forwards the *Confirm* packet as per the relevant entry in the *ConfirmRouteFind* table and deletes this entry from the table.

When the *ToBePassive* source receives the *Confirm* packet, it changes its status from Active to Passive source. A Passive source node will no longer flood *JoinReq* packets until it becomes an Active source. Data packets will be forwarded to its receivers through its core node.

After flooding the *JoinReq* packet, the core node expects a *PassReq* packet from each of its Passive sources. When such a Passive source gets a *JoinReq* packet from its core node, with hop distance less than or equal to MaxHop, it sends *PassReq* packet back to its core node. A Passive source node ignores the *CoreAcceptance flag*.

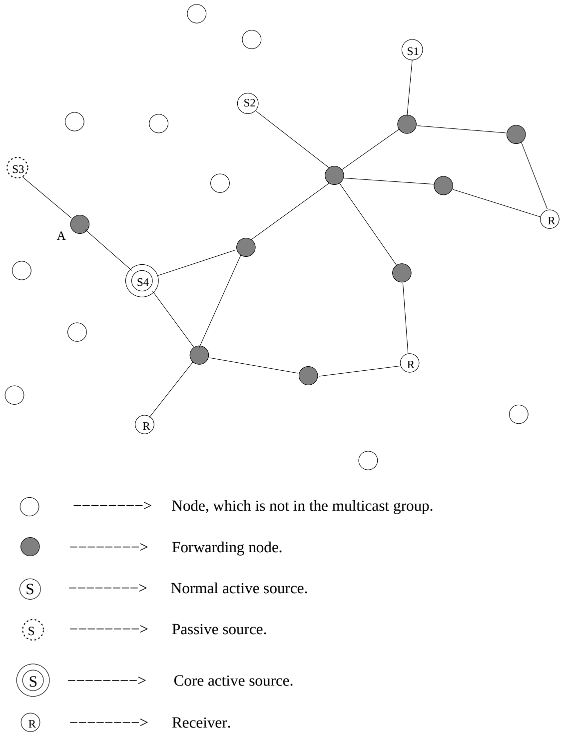

Figure 4: Mesh topology in DCMP

Whenever a core node receives the *PassReq* packet from one of its Passive sources, it resets *PassiveSourceExistence* timer (refer Figure 2) in the *PassSourceAddr* table for that Passive source.

If the *PassReq* packet does not reach the core node (either due to link breakage or due to collision), the *PassiveSourceExistence* timer expires. If this happens, the core node deletes the entry related to this Passive source from the *PassSourceAddr* table and also decrements the *PassiveSupported* counter.

In the event that the *Confirm* packet sent by the core node is unable to reach the *ToBePassive* source node or a Passive source node, the *ConfirmWait* timer expires. In that case, this source resets the lock flag and hence becomes an Active source.

Due to the mobility of nodes, it can happen that a Passive source will get a *JoinReq* packet from its core node with hop distance more than MaxHop. In this case, it sends a *PassReq* packet with the CoreReq field reset, and changes to an Active source. Basically this Passive source node wants to discontinue using its current core node as the forwarding node, as the hop distance involved is high. When a core node receives a *PassReq* packet without the CoreReq field set from its Passive source, it removes this Passive source entry from its *PassSourceAddr* table and decrements the *PassiveSupported* counter. At this point, if this counter becomes zero, the core node changes to an Active source. We now explain the proposed protocol with an example.

4.1 An Example

In Figure 4, there are four sources S1, S2, S3, S4, and three receivers, each indicated by R in the multicast group.

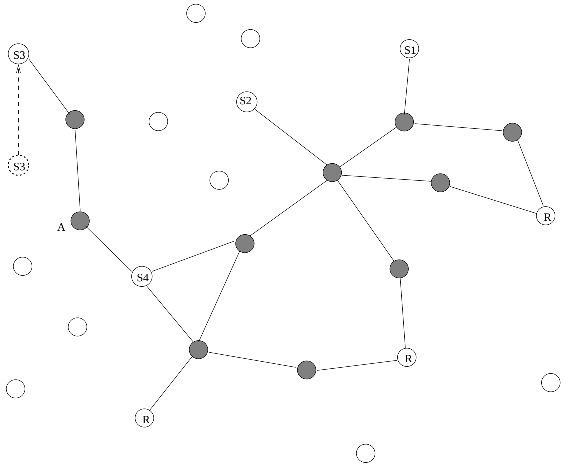

Figure 5: Topology change in DCMP due to movement of node S3

We can assign identification numbers of 1, 2, 3, and 4 to S1, S2, S3, and S4 without any loss of generality. MaxPassSize and MaxHop parameters are taken as 1 and 2, respectively.

Initially, to discover the receiver nodes, each source node floods *JoinReq* packets (with *CoreAcceptance flag* set). The packets will be received by receivers as well as sources. Upon receiving the *JoinReq* packet, each receiver R sends a *Reply* packet along the reverse path. When an intermediate node receives this *Reply* packet, it sets its FgFlag and forwards the *Reply* packet. In this way, routes between source and receiver nodes are established.

When S3 receives a *JoinReq* packet from S4, it checks the *CoreAcceptance flag*. Since, a) initially the *CoreAcceptance flag* is set in the *JoinReq* packet (sent by S4), b) the hop distance traveled by the *JoinReq* packet is 2, and c) ID of S3 is less than ID of S4, it satisfies all the three conditions prescribed above for status change. Hence, it (node S3) sends a *PassReq* packet to S4. In this way, source node S3 changes its status from Active source to *ToBePassive* source and S4 becomes *ToBeCore* node for S3.

After sending the *PassReq* packet, node S3 sets the lock flag so that it will not become core node for other source nodes and will not send *PassReq* packets to other source nodes. Even if S3 gets a *PassReq* packet from S2, a *Confirm* packet is not sent back to node S2.

When the intermediate node A receives the *PassReq* packet, it stores the ID of node S3 in its *ConfirmRouteFind* table and then forwards the packet to S4. After receiving the *PassReq* packet, S4 checks its *PassiveSupported* counter. If the counter is already 1 (as it might have got *PassReq* from Source S2), node S4 does not send a *Confirm* packet to node S3. Hence, at node S3, the *ConfirmWait* timer expires and S3 changes from *ToBePassive* to Active source node. On the other hand, if the counter is 0 at node S4, it sends a *Confirm* packet to S3. S4 then makes an entry for node S3 in its *PassSourceAddr* table, and also increments the *PassiveSupported* counter. Hence, S4 becomes a forwarding node, *i.e.,* data packets sent by S3 will be forwarded by node S4 to the end recipients. As the *PassiveSupported* counter is 1 now, future flooding of *JoinReq* packets by S4 is done with the *CoreAcceptance flag* reset.

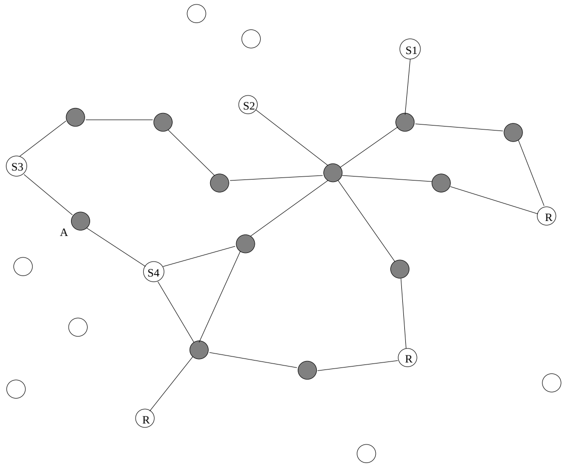

Figure 6: Mesh topology in ODMRP

When the intermediate node A receives a *Confirm* packet, it sets its FgFlag and forwards the *Confirm* packet to S3. After receiving this *Confirm* packet, the *ToBePassive* node S3 changes from Active source to Passive source. After this, whenever Passive source S3 receives a *JoinReq* packet from its core node S4, it sends a *PassReq* packet, regardless of the value of *CoreAcceptance flag* in the packet.

Let us now consider that the relative positions of the nodes have changed due to their mobility (Figure 5). Now, it can happen that S3 will get a *JoinReq* packet from its core node S4 with hop distance 3, which is more than MaxHop. Because of this, S3 wants to discontinue using S4 as the core node. Hence, S3 changes from *Passive* to Active source node and sends a *PassReq* packet with the CoreReq field reset. When S4 receives this *PassReq* packet from S3, it understands that S3 is too far from it, it deletes the entry corresponding to S3 from its *PassSourceAddr* table and decrements the *PassiveSupported* counter. This counter is 0 now, future flooding of *JoinReq* packets by S4 is done with the *CoreAcceptance flag* set.

Since in DCMP, *JoinReq* is sent only by Active Sources it results in less number of forwarding nodes in the multicast mesh compared to ODMRP. This is clear from Figures 4 and 6. In case of DCMP, the number of forwarding nodes in the multicast mesh is 10 whereas in case of ODMRP, the number of forwarding nodes is 12. Due to this reason the number of data packet transmissions in DCMP is less compared to ODMRP.

5. PERFORMANCE EVALUATION AND ANALYSIS

5.1 Analysis of Reduction in Control Overhead

We now analytically estimate the number of Passive sources and compare analytical and simulation studies later in Section 5.4.

5.1.1 Notation used

r: Transmission range.
L: Terrain range (assuming a square terrain with a side length L).
T_{sim}: Simulation time.

T_{ref}: *JoinReq* refresh time.
N: Number of nodes in the network.
S: Number of sources.
R: Number of receivers.
S_p: Average number of Passive sources.
S_a: Average number of Active sources.
M_h: MaxHop parameter.
M_p: MaxPassSize parameter.
P_n: Probability that a source is Active source.
P_p: Probability that a source is Passive source.
P_c: Probability that a source is Core Active source.
P_c^v: Probability that a source is Core Active source with vacancy.
P_c^{nv}: Probability that a source is Core Active source with no vacancy.

5.1.2 Control overhead estimation for ODMRP

With periodic *JoinReq* packets, the number of control packets generated in ODMRP in T_{sim} is $S.N.\frac{T_{sim}}{T_{ref}}$. Assuming that all the potential receivers receive the *JoinReq* packets and originate *Reply*, a total of $\sum_{i=1}^{S}\sum_{j=1}^{R} H_{ij}$ *Reply* packets will be generated where H_{ij} is hop distance between a source and receiver pair. Hence, the total number of control packets generated in ODMRP in time T_{sim} is

$$S.N.\frac{T_{sim}}{T_{ref}} + \sum_{i=1}^{S}\sum_{j=1}^{R} H_{ij} \qquad \text{—— (1)}$$

5.1.3 Control overhead estimation for DCMP

Assuming that every Active source generates the *JoinReq* control packet periodically, the number of *JoinReq* control packets generated in time T_{sim} is $S_a.N.\frac{T_{sim}}{T_{ref}}$.

In the ideal case all the potential receivers will receive the *JoinReq* and generate *Reply*. The number of *Reply* control packets generated in time T_{sim} is $\sum_{i=1}^{S_a}\sum_{j=1}^{R} H_{ij}$. The number of *PassReq* control packets generated in time T_{sim} is $(\sum_{i=1}^{S_p} M_{h i}).\frac{T_{sim}}{T_{ref}}$ and the number of *Confirm* control packets generated in time T_{sim} is $(\sum_{i=1}^{S_p} M_{h i}).\frac{T_{sim}}{T_{ref}}$. Hence, the total number of control packets generated in DCMP in time T_{sim} is

$$S_a.N.\frac{T_{sim}}{T_{ref}} + \sum_{i=1}^{S_a}\sum_{j=1}^{R} H_{ij} + 2.(\sum_{i=1}^{S_p} M_{h i}).\frac{T_{sim}}{T_{ref}} \qquad \text{—— (2)}$$

The reduction in control overhead is: (1) - (2) =

$$S_p.N.\frac{T_{sim}}{T_{ref}} + \sum_{i=1}^{S_p}\sum_{j=1}^{R} H_{ij} - 2.(\sum_{i=1}^{S_p} M_{h i}).\frac{T_{sim}}{T_{ref}} \qquad \text{—— (3)}$$

5.1.4 Estimation of number of Passive sources

From Equation (3) we can see that, the reduction in control overhead is proportional to the number of Passive sources S_p in the group.

A source node may be an Active source, Passive source or Core Active source. Hence we have

$$P_p + P_n + P_c^v + P_c^{nv} = 1 \qquad \text{—— (4)}$$

The number of sources within $M_h.r$ hop distance from any source node is $Ns = (S-1).\Pi.(M_h.r/L)^2$. The probability that a source node will become Passive source (P_p) is

$$1-(1-P_n+P_c^v)^{Ns/2} \qquad \text{—— (5)}$$

$$\text{where } P_c^{nv} = (P_p)^{M_p} \qquad \text{—— (6)}$$

By substituting P_p from Equation (6) we can get S_p, where $S_p = S.P_p$ —— (7)

5.2 Simulation Environment

We evaluated the performance of our proposed scheme by carrying out various simulation studies. The simulation model was built around GlomoSim [25] developed at the University of California, Los Angeles using PARSEC [26]. The IEEE 802.11 DCF is used as the MAC protocol. The free-space propagation model [27] is used at the radio layer. In the radio model, we assumed that the radio type was radio-capture.

In our simulation model, 50 mobile nodes move within a 1000m × 1000m area. The random-way-point model implemented in GlomoSim [25] is used in simulation runs and the pause time is taken as 10 seconds. The radio transmission range used is 250 meters. Channel capacity is assumed as 2Mbits/sec. Constant Bit Rate (CBR) model is used for data flow and each data packet size is taken as 512 bytes. The network traffic load is kept at 10 packets/sec throughout the simulation. Active sources flood *JoinReq* packets at intervals of 3 seconds. Sources and receivers are chosen randomly and join the multicast session at the beginning and remain as members throughout the simulation. The multicast group size is taken as 5 and 20 for small and large multicast groups, respectively. Each simulation is run for 200 seconds of simulation time and the final results are averaged over 20 simulation runs. We have used same simulation parameters for both DCMP and ODMRP unless otherwise specified.

5.3 Metrics

The performance evaluation metrics used in simulation are as follows:

- **Data Packet Delivery Ratio**: The percentage of data packets received by the receivers.
- **Number of Control Packets Transmitted per Data Packets Delivered**: This metric represents the degree of control overhead.
- **Number of Data Packets Transmitted per Data Packets Delivered**: This metric represents the multicast routing efficiency.

5.4 Simulation Results

5.4.1 Impact of MaxHop and MaxPassSize Parameters

In this experiment, the number of sources is taken as 5 for both of the multicast group sizes. The mobility is kept at 20m/s. The variation of number of Passive sources as a function of MaxHop size with different MaxPassSize is shown in Figures 7 and 8 for small and large multicast group sizes, respectively. It is clear from the figures that with increasing MaxHop, the number of Passive sources increases. This is because, a larger area can be covered with a single Core Active source. Similarly, with increase in MaxPassSize, number of Passive sources increases. Even though the trends

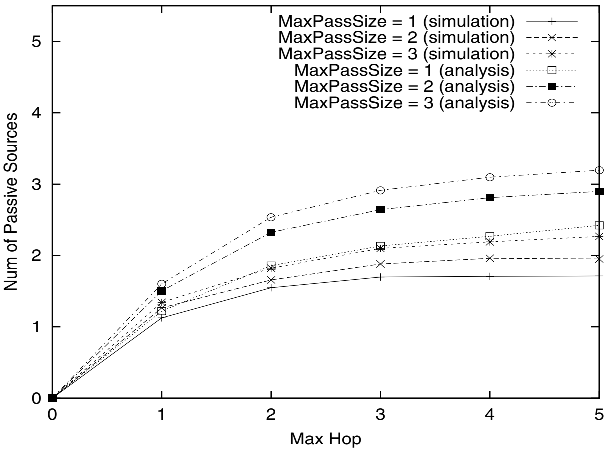

Figure 7: Number of Passive sources Vs MaxHop with varying MaxPassSize (for small multicast group)

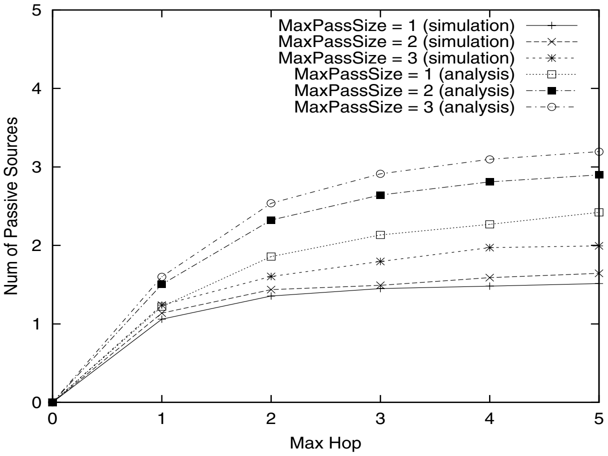

Figure 8: Number of Passive sources Vs MaxHop with varying MaxPassSize (for large multicast group)

of analytical results are same as that of simulation results, they deviate because simulation is more realistic. For example, loss of a *Confirm* packet due to collision could alter the results of simulation.

The impact of MaxHop size and MaxPassSize on the number of Active sources is evident from Figures 9 and 10 for small and large multicast group sizes, respectively.

Since with increasing MaxHop and MaxPassSize parameters the number of Passive sources increases, the control overhead decreases, which is clear from Figures 11 and 12.

Figures 13 and 14 show the data packet delivery ratio as a function of MaxHop parameter with different MaxPassSize. With increasing MaxHop and MaxPassSize parameters, the number of Passive sources increases. Hence, the redundancy of paths in the mesh decreases, resulting in reduction in packet delivery ratio.

Based on the above results, we have chosen MaxHop and MaxPassSize as 2 for the rest of our experiments.

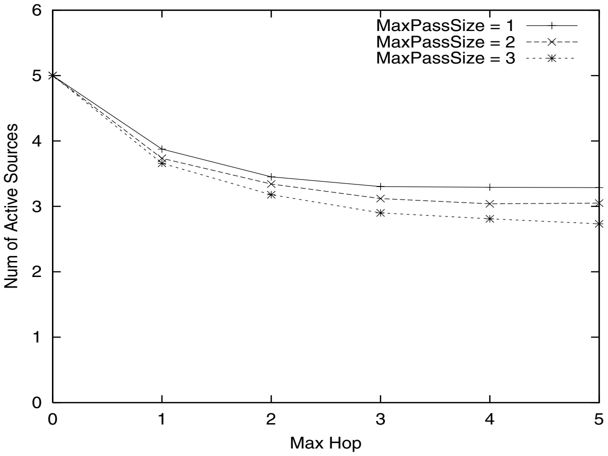

Figure 9: Number of Active sources Vs MaxHop with varying MaxPassSize (for small multicast group)

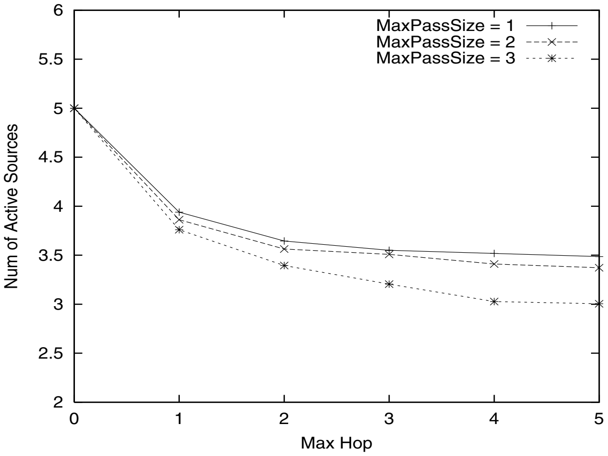

Figure 10: Number of Active sources Vs MaxHop with varying MaxPassSize (for large multicast group)

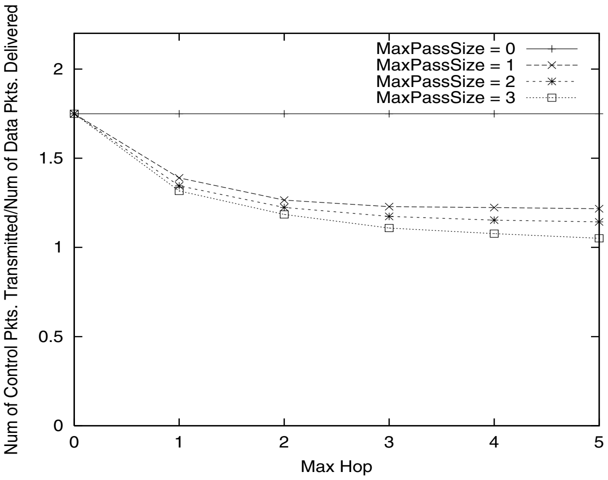

Figure 11: Control overhead Vs MaxHop with varying MaxPassSize (for small multicast group)

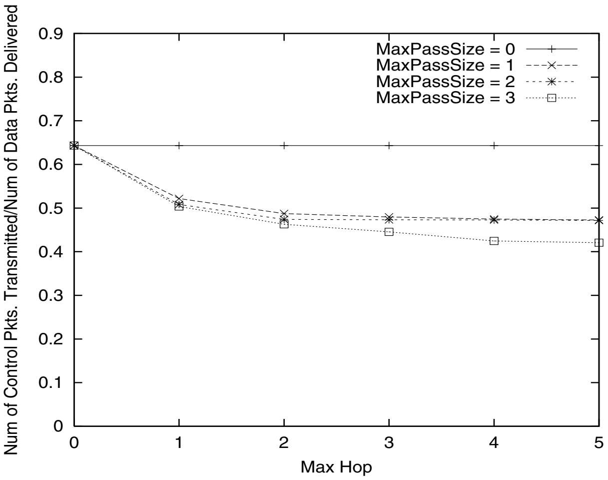

Figure 12: Control overhead Vs MaxHop with varying MaxPassSize (for large multicast group)

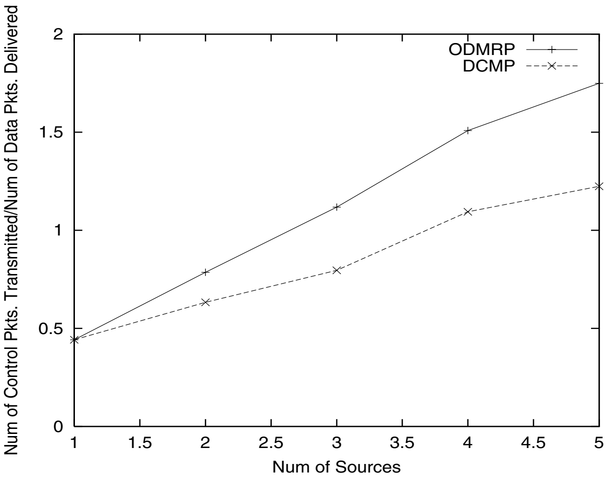

Figure 15: Comparison of control overhead for small multicast group

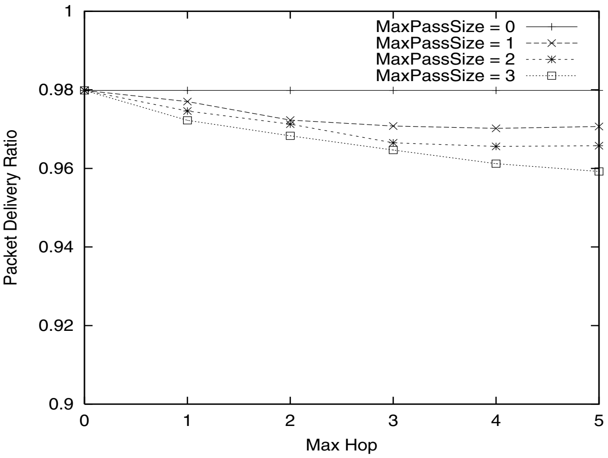

Figure 13: Packet delivery Vs MaxHop with varying MaxPassSize (for small multicast group)

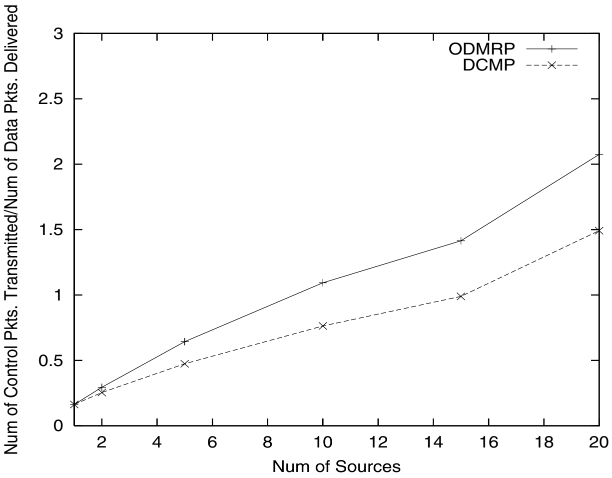

Figure 16: Comparison of control overhead for large multicast group

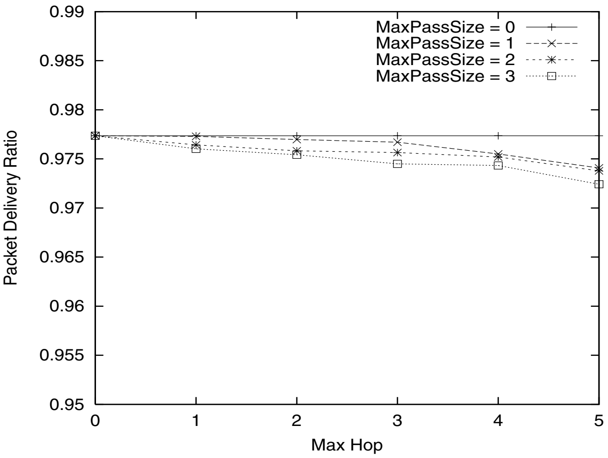

Figure 14: Packet delivery Vs MaxHop with varying MaxPassSize (for large multicast group)

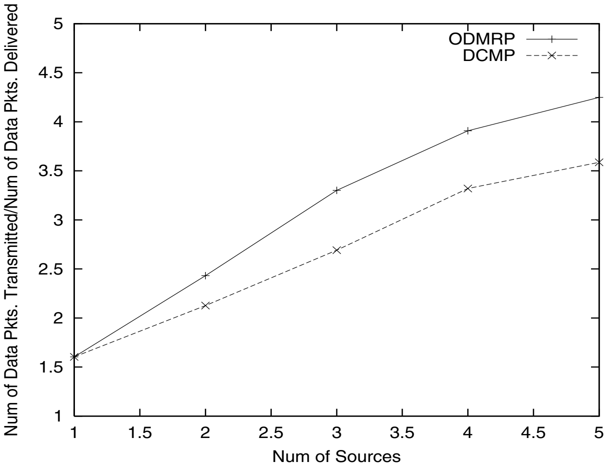

Figure 17: Comparison of number of data transmissions for small multicast group

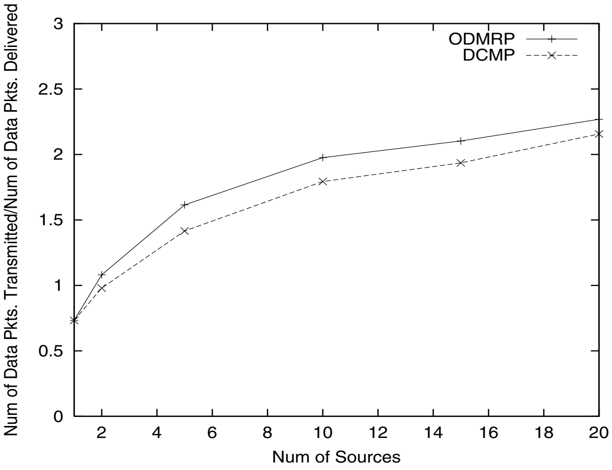

Figure 18: Comparison of number of data transmissions for large multicast group

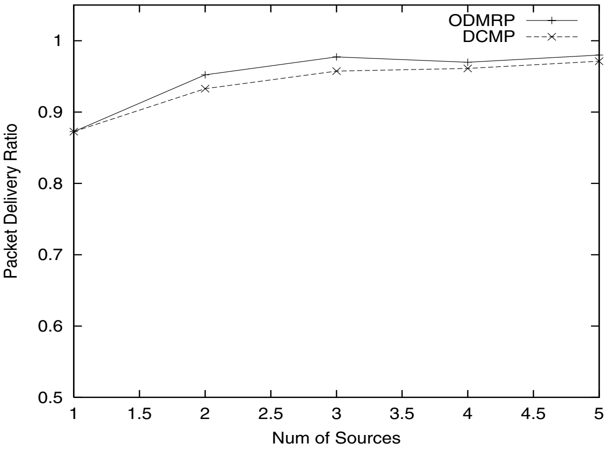

Figure 19: Comparison of packet delivery ratio for small multicast group

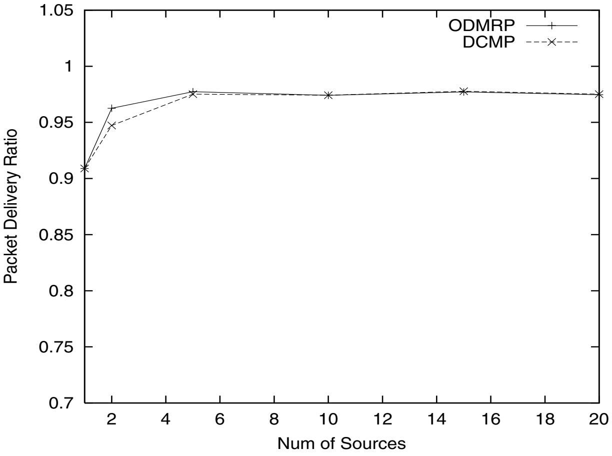

Figure 20: Comparison of packet delivery ratio for large multicast group

5.4.2 *Impact of Number of Sources*

In this experiment also, mobility has been kept constant at 20 m/sec.

The variation of control overhead as a function of the number of sources for small and large multicast group sizes is shown in Figures 15 and 16, respectively. As expected, when the number of sources increases, the control overhead increases in both the cases. However, in case of DCMP, the increase in control overhead is markedly less compared to that in ODMRP (about 30 %). This is due to the fact in DCMP, flooding of the *JoinReq* packets is done only by Active sources, whereas in ODMRP, all sources need to flood the *JoinReq* packets. We also observe from the figures that as the number of sources increases, the gain of DCMP over ODMRP in terms of control overhead also increases. This is because, with an increase in the number of sources, the probability that sources come nearer to each other (and hence becoming Passive) is more.

Figures 17 and 18 show the number of data packets transmitted per data packet delivered as a function of the number of sources. Since, DCMP creates a mesh with less number of redundant routes, it transmits a lesser number of data packets compared to ODMRP (about 10-15 % less). Referring back to the example in Figures 4 and 6, we see that in DCMP there are 10 forwarding nodes in the mesh, whereas in ODMRP there are 12 forwarding nodes in the mesh. Figure 17 shows that as the number of sources increases, advantage of DCMP over ODMRP in terms of number of data transmissions also increases. But for a large multicast group, this difference is about constant. This is due to the fact that there are many forwarding nodes in a large multicast group. Hence, if some source nodes become Passive, there will not be much difference in the number of forwarding nodes.

We plot the packet delivery ratio as a function of the number of sources in Figures 19 and 20 for small and large multicast groups, respectively. We observe that for large multicast group size, the packet delivery ratio of DCMP is almost same as that of ODMRP. But, for small multicast groups, the maximum difference is about 2%. We also observe that in DCMP the loss of data packets due to collision is less compared to that in ODMRP. This can be explained by the fact that in DCMP there is lesser amount of transmission of control and data packets than in ODMRP.

5.4.3 *Impact of Mobility*

In this study, the number of sources is taken as 5 for both multicast group sizes and the node speed is varied from 0 m/s to 20 m/s. Figures 21 and 22 show control overhead as a function of mobility. Since the DCMP protocol uses a soft state approach to maintain the multicast mesh, the control overhead is about constant with varying mobility.

The number of data transmissions as a function of mobility is shown in Figures 23 and 24 for small and large multicast groups, respectively. As we observe, without mobility, the number of data packet transmissions is high. It is because of the fact that dropping of *Reply* packets is less (while traversing along reverse path), so many intermediate nodes become forwarding nodes. But as mobility increases (beyond 10 m/s), again data packet transmission slightly increases. This is due to more frequent link failures, causing more nodes to unnecessarily become temporary forwarding nodes.

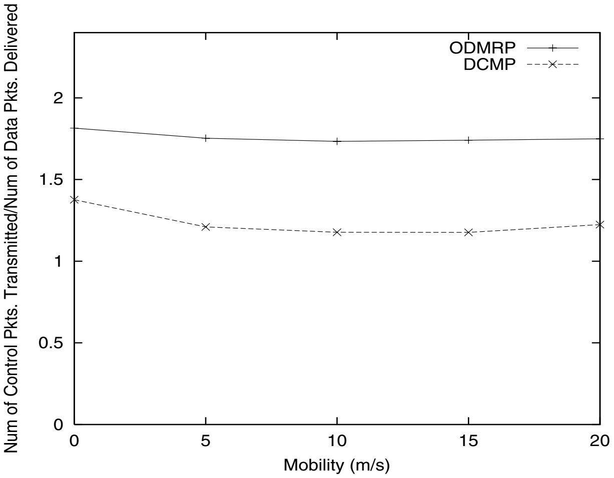

Figure 21: Comparison of control overhead for small multicast group

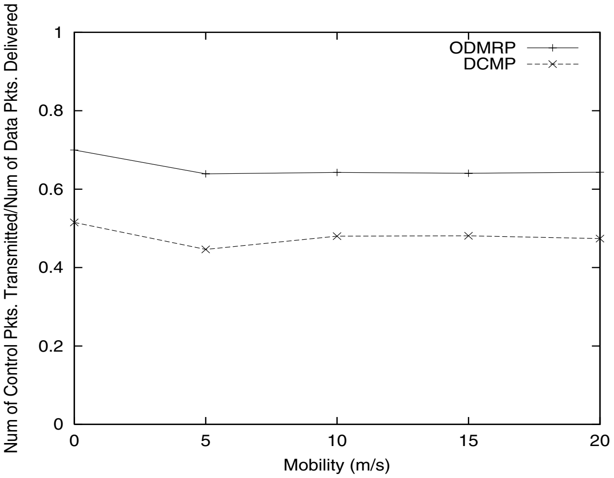

Figure 22: Comparison of control overhead for large multicast group

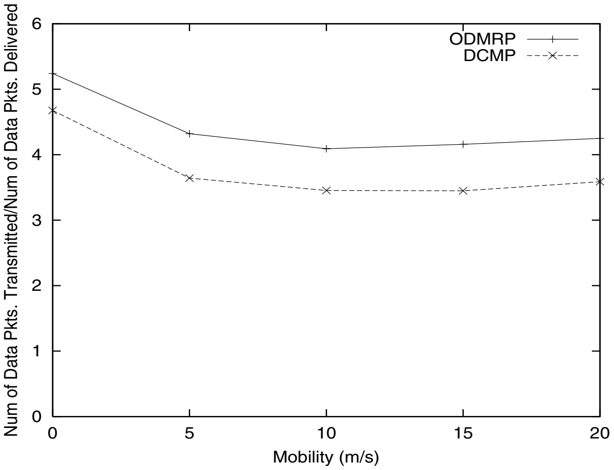

Figure 23: Comparison of number of data transmissions for small multicast group

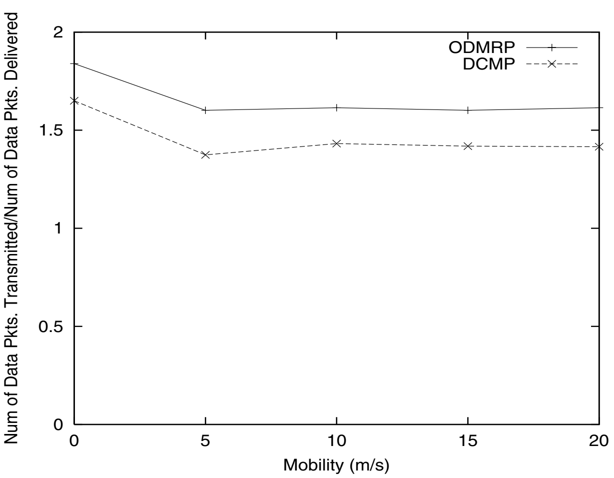

Figure 24: Comparison of number of data transmissions for large multicast group

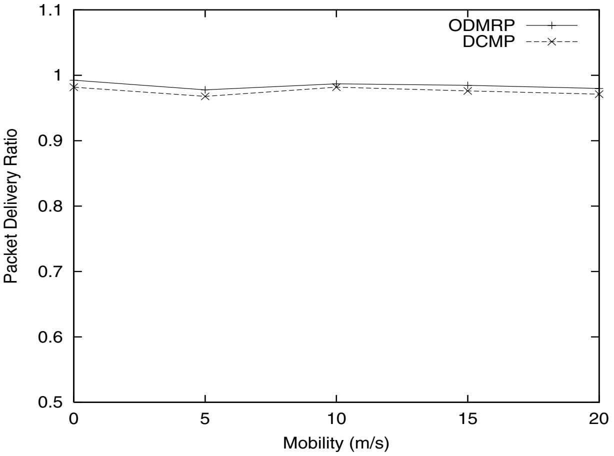

Figure 25: Comparison of packet delivery ratio for small multicast group

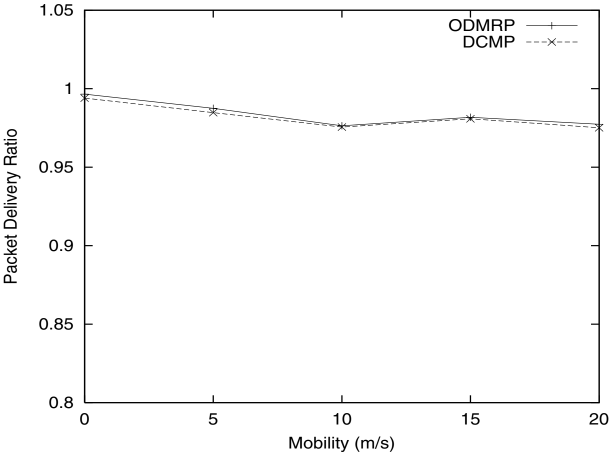

Figure 26: Comparison of packet delivery ratio for large multicast group

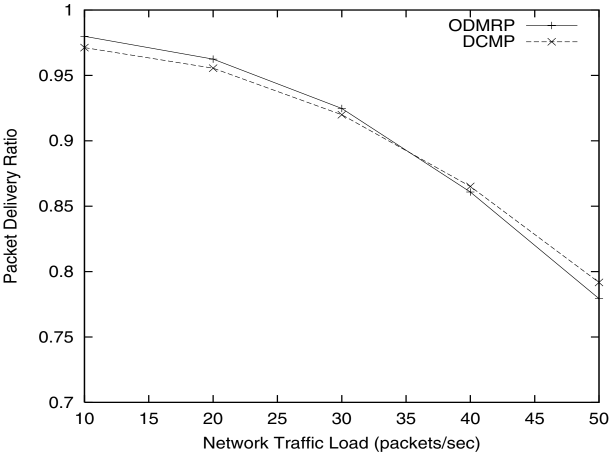

Figure 27: Comparison of packet delivery ratio for small multicast group

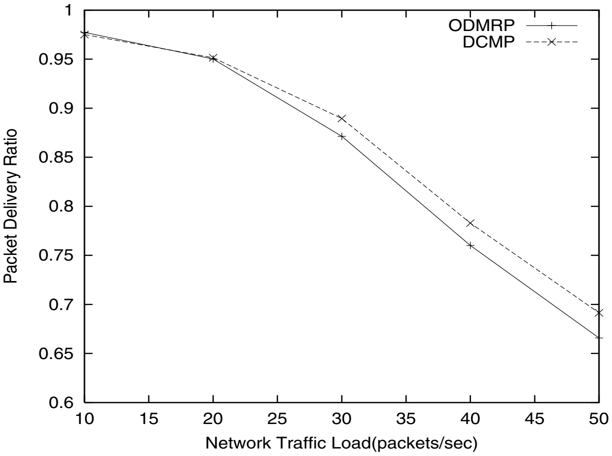

Figure 28: Comparison of packet delivery ratio for large multicast group

Packet delivery ratio as a function of mobility is shown in Figures 25 and 26. As we observe for large multicast group, packet delivery ratio of DCMP is about the same as that of ODMRP. But for small multicast group, packet delivery ratio is about 2% less compared to ODMRP, at high mobility. For the large multicast group, there are many forwarding nodes in the multicast group. Hence, even if some source nodes become Passive, it does not reduce the robustness of the mesh that much. But, in the case of small multicast group, when a source becomes Passive, it affects mesh robustness slightly.

5.4.4 Impact of Load

In this simulation, the number of sources is taken as 5 for both the multicast group sizes and mobility is kept at 20m/s. The packet delivery ratio Vs network traffic is shown in Figures 27 and 28 for small and large multicast group sizes, respectively. Since in DCMP the number of control packet transmissions is less compared to ODMRP and hence data packet losses due to collisions are also less, resulting in more data packet delivery at high load.

6. CONCLUSIONS

We have proposed an efficient, mesh based, on-demand multicast protocol, DCMP, for Ad hoc networks, where the multicast topology is shared mesh. The key concept in this protocol is to make some sources Passive, which then forward data packets through their core nodes. The major advantage of this protocol is its increased scalablity. This can be mainly attributed to the reduced control overhead. We implemented DCMP using GlomoSim and the simulation results show that there is a 30% reduction in control overhead, while the multicast efficiency is increased by 10-15%, at the cost of a small (2%) reduction in packet delivery ratio for light network loads. We also find that packet delivery ratio is improved at high load.

7. REFERENCES

[1] S. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad hoc Wireless Multicast Protocols", *Proceedings of IEEE INFOCOM 2000*, pp. 565-574, March 2000.

[2] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks", *Proceedings of ACM SIGCOMM'97*, pp. 39-49, September 1997.

[3] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Proceedings of ACM SIGCOMM'94*, pp. 234-244, September 1994.

[4] C. -C. Chiang, H. K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", *Proceedings of IEEE SICON'97*, pp. 197-211, April 1997.

[5] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", *ACM Mobile Networks and Applications Journal: Special Issue on Routing in Mobile Communication Networks*, vol. 1, no. 2, pp. 183-197, October 1996.

[6] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing", *Proceedings of IEEE WMCSA'99*, pp. 90-100, February 1999.

[7] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks", *Mobile Computing, Kluwer Academic Publishers*, Chapter 5, pp. 153-181, 1996.

[8] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", *Proceedings of IEEE INFOCOM'97*, pp. 1405-1413, April 1997.

[9] C. K. Toh, "Associativity Based Routing for Ad hoc Mobile Networks", *Wireless Personal Communication*, vol. 4, no. 2, pp. 1-36, March 1997.

[10] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi, "Signal Stability Based Adaptive Routing for Ad hoc Mobile Networks", *IEEE Personal Communications*, vol. 4, no. 1, pp. 36-45, February 1997.

[11] Z. J. Haas, M. R. Pearlman, and P. Samar, “Zone Routing Protocol (ZRP)”, *Internet Draft*, draft-ietf-manet-zrp-04.txt, January 2001, Work in progress.

[12] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, “AMRoute: Ad hoc Multicast Routing Protocol”, *Internet-Draft*, draft-talpade-manet-amroute-00.txt, August 1998, Work in progress.

[13] C. W. Wu, Y. C. Tay, and C. K. Toh, “Ad hoc Multicast Routing Protocol Utilizing Increasing id-numberS (AMRIS) Functional Specification”, *Internet-Draft*, draft-ietf-manet-amris-spec-00.txt, November 1998, Work in progress.

[14] T. Ozaki, J. B. Kim, and T. Suda, “Bandwidth Efficient Multicast Routing Protocol for Ad hoc Networks”, *Proceedings of IEEE ICCCN’99*, pp. 10-17, October 1999.

[15] E. M. Royer and C. Perkins, “Multicast using Ad hoc on Demand Distance Vector Routing”, *Proceedings of ACM Mobicom’99*, pp. 207-218, August 1999.

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

[17] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: Core Extraction Distributed Ad hoc Routing”, *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 8, pp. 1369-1379, August 1999.

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

[19] 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.

[20] C. -C. Chiang and M. Gerla, “On Demand Multicast in Mobile Wireless Networks”, *Proceedings of IEEE ICNP’98*, October 1998.

[21] 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. 1380-1994, August 1999.

[22] S. Lee and C. Kim, “Neighbor Supporting Ad hoc Multicast Routing Protocol”, *Proceedings of ACM Mobihoc 2000*, pp. 37-50, August 2000.

[23] Y. -B. Ko and N. H. Vaidya, “Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms”, *Proceedings of IEEE WMCSA’99*, February 1999.

[24] A. Ballardie, P. Francis, and J. Crowcroft, “Core Based Tree (CBT): An Architecture for Scalable Inter-Domain Multicast Routing”, *Proceedings ACM SIGCOMM’93*, pp. 85-95, October 1993.

[25] 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*.

[26] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H. Y. Song, “PARSEC: A Parallel Simulation Environment for Complex Systems”, *IEEE Computer*, vol. 31, no. 10, pp. 77-85, October 1998.

[27] T. S. Rappaport, *Wireless Communications Principles and Practice*, Prentice Hall PTR, New Jersey, 1996.