UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

Efficient Multicast Routing in MANETs Using Prefix Labels

Permalink

<https://escholarship.org/uc/item/76s9d77n>

Author

Garcia-Luna-Aceves, J.J.

Publication Date

2009-08-03

Peer reviewed

Efficient Multicast Routing in MANETs Using Prefix Labels

J.J. Garcia-Luna-Aceves^{†*}

[†] Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
Email: jj@soe.ucsc.edu

Dhananjay Sampath^{*}

^{*}Computer Engineering Department
University of California, Santa Cruz
Santa Cruz, CA 95064
Email: dsampath@soe.ucsc.edu

Abstract—We introduce Prefix-based Multicast Routing (PMR), the first approach to multicast routing in MANETs in which the network-wide dissemination of control information is independent of the number of groups and sources per group. PMR establishes a labeled directed acyclic graph (LDAG) rooted at an elected node in the MANET and assigns a prefix label to each node denoting its network location relative to the root of the LDAG. Routes between any two prefix labels are implicit in the labels themselves. The sources and receivers of a multicast group use a consistent hashing function to map the identifier of the group (e.g., an IP multicast address) onto the group-prefix-label. The node whose prefix label is the closest match to the group-prefix-label serves as the core of the group. As in prior receiver-initiated multicast approaches, receivers join a multicast group by sending join requests towards the group core, and multicast sources simply forward their multicast data packets towards the cores of the groups. We use simulation experiments to compare PMR with ODMRP and MAODV for different mobility scenarios with varying number of nodes, groups and receivers. The results clearly show that PMR is far more efficient than traditional multicast routing approaches, even in relatively small networks.

I. INTRODUCTION

The objective of a multicast routing protocol is to establish routing structures between the sources and receivers of multicast groups, such that they can communicate with each other even when they do not enjoy direct physical connectivity. In a MANET, the multicast routing structures linking multicast sources with the receivers of the target multicast groups can be trees or meshes, and a critical aspect in the design of such a protocol consists of limiting the amount of bandwidth consumed in building the necessary trees or meshes.

Interestingly, as Section II describes, the traditional approach adopted by the multicast routing protocols proposed and implemented to date for MANETs, consists of building multicast trees or meshes by either having the core of each multicast group, or each source of each multicast group, disseminate control packets to the entire MANET. This network-wide dissemination of control information per multicast group, is an approach that does not work well in large-scale MANETs or large MANETs with many multicast groups. However, it is actually a necessity resulting from adopting the IP multicast model for the maintenance of multicast groups, together with the use of IP or MAC addresses for packet forwarding. More specifically, IP and MAC addresses are really names,

especially when they denote multicast groups, in that, they are assigned to nodes independently of their relative location in the MANET. In the IP multicast model, the constituency of a multicast group is not known to any one node, and the only identifiers used for multicast forwarding are the names of the multicast groups. Accordingly, the only way for either the core or a source of a multicast group to update potential receivers, of its relative location, is by sending updates to the entire network. Receivers can then send join requests towards the core, or towards sources of a multicast group, to build a multicast tree or mesh. Multicast data packets can then be forwarded over this routing structure, using the name of the group.

The main contribution of this paper is the introduction of the first approach to multicast routing in MANETs in which the network-wide dissemination of routing information grows only linearly with the number of nodes in the network and independently of the number of multicast groups or sources per group. This makes it far more efficient than traditional sender- or receiver-initiated approaches. In this paper, we assume that the MANET is connected, and leave the handling of network partitions and disruption-tolerant multicasting for future work.

Section III presents the Prefix-based Multicast Routing (PMR) protocol, which adopts the IP multicast model and enables receiver-initiated multicasting for any number of multicast groups. PMR first establishes a labeled directed acyclic graph (LDAG) rooted at an elected node in the MANET, and uses prefix labels for forwarding of control and multicast data packets. A prefix label is assigned to each node denoting its network location relative to the root of the LDAG. Routing between any two prefix labels is implicit owing to the ordering with respect to the root of the LDAG. Multicast sources and receivers use a consistent hashing function to map the name of a multicast group (e.g., an IP multicast address) onto its group prefix label. The node with a prefix label that is the closest to the group prefix label becomes the core of the multicast group. Based on this simple indirection, PMR supports receiver-initiated multicasting as in CBT [1] or CAMP [3]. Receivers build a multicast routing tree by sending join-requests towards the group-prefix-label, and a source similarly forwards data packets towards the group-prefix-label as well.

Section IV compares the performance of PMR with that of

ODMRP [7] and MAODV [10] in MANETs of 100 nodes. We selected these protocols as benchmarks because they are good representatives of multicast routing based on meshes or trees and their implementations are readily available. Furthermore, other multicast routing approaches exhibit similar signaling overhead. The results of our simulations show that PMR is much more efficient than ODMRP and MAODV, even for relatively small MANETs with just a few multicast groups. This is due primarily to the fact that PMR eliminates the network-wide dissemination of control packets by the cores or sources of multicast groups.

II. RELATED WORK

Multicast routing protocols for MANETs build either multicast trees or meshes over which multicast data packets are forwarded. The signaling they use to build such multicast routing structures can be classified as sender-initiated or receiver-initiated. In the receiver-initiated approach, introduced in CBT [1], called shared-tree approach, only one node—called the *core* of the group—originates the dissemination of information about a multicast group reaching all other nodes, and receivers send explicit requests towards the core to join the group. Sources forward multicast data packets towards the core, and the packets are then multicast once they reach any node in the multicast tree or mesh. In contrast, source-based or sender-initiated schemes have each multicast source initiate the dissemination of state information that reaches all nodes in the network. Many multicast routing proposals exist for MANETs (e.g., see [8]), and due to space limitations we discuss only a very small fraction to highlight the novelty of our approach.

MAODV [10] maintains a shared tree for each multicast group consisting of receivers and relays. Sources acquire routes to the group on demand as in on-demand unicast routing. Receivers join the shared tree by means of a special route request (RREQ) packet. Any node belonging to the shared multicast tree can answer the RREQ with a route reply (RREP). A sender joins a group through the neighbor that reports the freshest route in a RREP with the minimum hop count to the tree. A node that does not belong to the multicast group must first send a non-join RREQ, which is treated like a RREQ to reach the group.

ADMR [4] maintains a source-based multicast tree for each sender of a multicast group. A new receiver performs a network-wide flood of a multicast solicitation packet when it needs to join the multicast group. Each source replies to the solicitation and the receiver sends a receiver join packet to each source that answers the solicitation. Each source-based tree is maintained by periodic keep-alive packets from the source, which allow intermediate nodes to detect link breaks in the tree by the absence of data or keep-alive packets. A new sender also sends a network-wide flood to allow existing group receivers to send receiver joins to the source. MZR [14] maintains source-based trees, like ADMR, but performs zonal routing; and hence the dissemination of control packets is less expensive.

In ODMRP [7], group membership and multicast routes are established and updated by the sources. Each multicast source broadcasts join queries periodically, and these are disseminated to the entire network to establish and refresh group membership information. When a join query reaches a multicast receiver, it creates and broadcasts a join-reply to its neighbors stating a list of one or more forwarding nodes. Nodes receiving a join-reply listing them as part of forwarding groups forward the replies with its own list of forwarding nodes. A join-reply is propagated by each forwarding group member, until it reaches a multicast source via the selected paths. This process establishes and updates the routes from sources to receivers and builds a mesh of nodes, the forwarding group. A source can multicast data packets to multicast receivers via selected routes and forwarding groups. Many ODMRP extensions have been proposed (e.g., [6], [8]), which differ on how flooding is reduced. However, the order of the signaling overhead in these schemes is the same as in ODMRP.

CAMP [3] avoids the need for network-wide disseminations from each source to maintain multicast meshes by using a core per multicast group. 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 drawbacks of CAMP are that it needs the pre-assignment of cores to groups and a unicast routing protocol to maintain routing information about the cores. PUMA [13] uses a receiver-initiated approach similar to that of CAMP, and implements a distributed algorithm to elect one of the receivers of a group as the core of the group, and to inform each router in the network of at least one next-hop to the elected core of each group.

With few exceptions, all prior schemes require the network-wide dissemination of signaling traffic from cores or multicast sources in order to support multicast routing that adheres to the IP multicast model. Recently, however, a few solutions have explored the use of geographical coordinates in multicasting. GMR [11] supports multicast routing as an extension of geographical unicast routing; however, the coordinates of the destinations are assumed to be known, which is not scalable, as it requires that information to be disseminated throughout the network. On the other hand, HRPM [2] is an example of multicast routing based on geographical coordinates that avoids the need for cores or sources to send control traffic to the entire network by means of geographic hashing and hierarchical routing. HRPM assumes that each node knows its own geographic location, which can be attained by using GPS [9] at every node. HRPM partitions the area of the MANET into equal size cells, and uses geographic hashing to determine access point (AP) coordinates and rendezvous point (RP) coordinates. The node closest to the RP coordinates becomes the rendezvous point of the group, and the node with the closest coordinates to the AP coordinates of a cell becomes the access point for the cell. Signaling can be directed to the RP to join and leave the multicast group by geographic routing. The key advantage of this scheme is that it reduces the multicast signaling overhead by eliminating the need for flooding from

Fig. 1. Overview of PMR Operation

cores (i.e., RPs). The disadvantage of the scheme is the need to implement geographic routing and the use of GPS.

III. PREFIX-BASED MULTICAST ROUTING (PMR)

A. Overview

PMR support multicast routing adhering to the IP multicast model, where group membership is not known by multicast sources and packet forwarding is based on the identifiers of multicast groups. Like other prior approaches [1], [3], [13] it implements a receiver-initiated approach to multicast routing. However, PMR does so while eliminating the need for each core of a multicast group to disseminate control information to the entire network, as well as the need for geographical coordinates. As we have stated, we assume a MANET without partitions.

The basis for the operation of PMR is the distributed establishment of a labeled directed acyclic graph (LDAG) in the MANET. Nodes establish and maintain dynamically an LDAG by exchanging *Hello* messages among neighboring nodes periodically. The root node of this LDAG is elected dynamically such that: (a) Each node is assigned a *prefix label* denoting the relative location of the node with respect to the root of the LDAG; (b) the prefix labels of a source and a destination define implicitly one or multiple valid routes between the two nodes; and (c) node mobility, link or node failures and addition of new nodes have limited impact on the prefix labels already assigned to other nodes.¹

The election of the root of the LDAG in PMR does not require any feedback to the root, all signaling is neighbor-to-neighbor, and the root of the LDAG is refreshed periodically. Each node transmits a *Hello* periodically to its neighbors with a sequence number created by the root to determine its freshness. This soft state is used to update the prefix labels of nodes as the MANET topology changes. The prefix labels used in PMR are formed by concatenating the labels assigned to all the ancestors of a node. In this fashion, a *Hello* originated at the root percolates through the MANET. The *Hello* sent by a node also lists the identifier and prefix labels of its one-hop neighbors, as well as the multicast groups to which the node

belongs. Over time, each node knows the identifiers and prefix labels of the nodes in its two-hop neighborhood. Figure 1(a) illustrates the resulting prefix labels of the nodes in an example MANET. Simple prefix routing determines one or multiple valid routes between any node and a target prefix label.

All nodes use a common hash function, such as SHA-1, to map any multicast group name into a group-prefix-label. In this paper, we assume that the hash function is such that the prefix labels corresponding to different group identifiers are uniformly distributed throughout the network.² Establishing receiver-initiated multicast routing in PMR is very simple based on this simple indirection. The core of a multicast group is simply defined to be that node whose prefix label is the closest match in its own two-hop neighborhood, to the group-prefix-label that is obtained by hashing the identifier of the target multicast group. The actual node serving as the core of a multicast group is not known to the group, only the group-prefix-label is. Nodes serve as cores of groups only as long as their own prefix labels are the best match for the corresponding group-prefix-labels.

To join a multicast group, a multicast receiver simply hashes the identifier of the multicast group and obtains the group-prefix-label. It then directs its join-request towards the node that matches it the best. The receiver then sends a join-request message towards the resulting group-prefix-label using simple prefix routing. As the join-request message is forwarded towards the core, the reverse path to the receiver is activated and relaying nodes become a part of the multicast forwarding tree for the group. Figure 1(b) illustrates the creation of a shared multicast tree for a multicast group whose group-prefix-label is '01,' which is also the prefix label of node *b* and thus makes that node the core of the multicast group. In the example, nodes *a*, *d*, *e*, *p*, *m*, *n* and *o* are already participating in the multicast routing tree, and nodes *r* and *i* are new receivers attempting to join. The join-request from node *r* is answered by node *n*, while the request from node *i* needs to traverse all the way to the core of the group, node *b*.

To send information to a multicast group, a source node that is not part of the shared multicast tree of the group simply hashes the identifier of the target multicast group to obtain the

¹For simplicity, the rest of this paper and our implementation of PMR assumes that a single root node is used for the establishment of the LDAG of a MANET. However, multiple roots could be used for added reliability.

²SHA-1 chooses each of the prefix labels randomly with equal probability, making it a fair distribution of labels in the virtual space.

group-prefix-label where the data packets should be forwarded. The source then forwards its multicast data packets towards the corresponding group-prefix-label using simple prefix routing. When the packet reaches any node in the shared multicast tree, the packet is multicast over the tree. Figure 1(c) illustrates the forwarding of multicast data packets. In the example, node k is a source that is not part of the shared multicast tree of the group, and hence forwards its multicast data packet to node c , which is the next hop in the prefix route between node k (with prefix label '023') and the group-prefix-label '01.' When the multicast data packet reaches node a , it is multicast over the shared tree

B. Prefix Labels

Let Σ be the alphabet containing a finite number of symbols and Σ^* be the set of all strings over Σ such that $|\Sigma| \geq 2$. Every node labels its links to each of its neighbors with a letter w from Σ . If w_i represents the letter assigned to the i^{th} link of any node then, $w_i \mapsto \{w_i \in \Sigma \mid w_i \neq w_{i+1} \forall i \leq d-1\}$, where d is the degree of the node. Hence, a unique letter is assigned to each link connecting any node to its neighbors. The labeling logic labels each node in the LDAG in a *breadth first* fashion. Given that the LDAG can be organized as a k -ary tree, with k being the degree of the LDAG, each child (up to k), is assigned a prefix label Λ as defined below:

Prefix Label: A prefix label Λ for node y is a word in Σ^* such that $\Lambda = \Lambda_{parent} \odot l'$, where Λ_{parent} is the prefix label obtained from the parent and $\odot$ is a concatenation operator where Λ_{parent} is concatenated with a unique suffix over k different choices from Σ to form Λ

Consider the example from Figure 1(a). After being elected as the root of the LDAG, node a begins labeling its neighbors. Each child node, c , b , e and d is given a prefix label with respect to the root node, such that their prefix matches that of the root and have unique suffixes from the set $\{1, 2, 3, 4\}$. It follows from the above definition that the prefix label of a node Λ uniquely identifies the node in a given LDAG. These labels define a *predecessor-successor* relation in the LDAG that lets a node choose its next-hop and route implicitly with very minimum state.

C. Information Stored and Exchanged

Each node maintains a *neighbor table* (NT) and a *two-hop neighborhood table* (TNT). The NT of node i contains an entry for each immediate neighbor of node i , and each entry states the identifier and prefix label of a neighbor; the most recent sequence number received from the neighbor; and a list of multicast groups to which the neighbor belongs, with each entry in the list indicating if node i was selected by its neighbor to reach the core of the group. The TNT of node i contains an entry for each two-hop neighbor of node i , and each such entry states the node identifier and prefix label of the neighbor. Each node also maintains a *multicast group table* listing information for each multicast group to which the node belongs; the information stored for each group includes: the identifier and group-prefix-label of the group, a

Fig. 2. PMR's Adaptive update timer

group sequence number, and the next-hop towards the core of the group.

Each node also maintains a *packet cache* listing information about multicast data packets heard recently, and a *signaling cache* that stores and aggregates update information to be sent in the next *Hello*.

PMR uses a single type of control message called a *Hello* to carry out its signaling. A *Hello* sent by node a contains the following information: (a) An LDAG identifier consisting of the node identifier of the root of the LDAG and the last sequence number created by the root, (b) the prefix label and node identifier of the node, (c) a sequence number created by the sending node, (d) the list of node identifiers and prefix labels for neighbors one and two hops away, and (e) group information list (*GIL*).

The *GIL* in a *Hello* consists of a list of one or more entries describing the state of the sending node with respect to a multicast group. Each entry in the *GIL* specifies: (a) the identifier and group-prefix-label of a multicast group; (b) an action being taken for the group (join-request, join-reply, quit request, or member); (c) the node identifier and prefix label of the next hop selected to reach the core of the group; and (d) the node identifier of one or more nodes to which a join-reply is intended.

Hellos are transmitted periodically by a node to all its immediate neighbors. The periodicity of *Hellos* is determined by the nature of the updates that need to be conveyed. Two separate timeout values, a long-timeout (LT) and a short-timeout (ST) are maintained by each node. Updates to the neighborhood information or *GIL* are aggregated until either timeout expires. Any event that changes the next hop towards the core of a group for which the node is active fires the short-timeout. When the ST expires, the node sends its *Hello* with the aggregated updates stored in its signaling cache, and the node takes the steps needed to maintain multicast trees. The long-timeout is fired when the topology is relatively static and fewer messages are needed to maintain the multicast trees of various groups. Figure 2(a) shows an example of the adaptive update timers. These timers ensure that the frequency with which a node transmits its *Hellos* is a function of the topology changes around the node.

D. Implicit Routing Using Prefix Labels

Determining the next hop towards a destination prefix label is straightforward. A node can apply simple greedy strategy by choosing the link to one of its neighbors that matches the destination prefix label to the maximum extent, i.e., *maximum prefix matching*. In addition, the node uses its two-hop neighborhood information to discover short-cuts, should they exist, to target prefix labels. This greedy logic does not encounter a local-minima because nodes are uniquely labeled and no two nodes can be assigned the same label. Hence, there is always some node that has a closer prefix to a destination prefix label.

Given that the LDAG is labeled correctly and if each node knows the prefix labels of all its neighbors, a greedy routing scheme (with sequence numbers) routes a packet from a source to a destination correctly. We omit the proof for brevity.

E. Joining, Leaving, and Maintaining Groups

Group membership is maintained using simple soft-state logic through the exchange of *Hellos* periodically. Although each *Hello* specifies control information in the *GIL* for one or multiple multicast groups, the steps taken for a given multicast group are independent of other groups, and hence we discuss the group maintenance activity in PMR by focusing on a single group.

A receiver interested in joining a particular multicast group hashes the name of the group to obtain the corresponding group-prefix-label. It then sends a join-request for the group in the *GIL* of its *Hello* stating the identifier and prefix label of the neighbor node it chooses along one of its implicit routes to the group-prefix-label.

A core node of the multicast group G receives a join-request for that group in the *Hello* from a neighbor x . It then, sends a join-reply entry for group G in the *GIL* of its own *Hello* and states the identifier of node x as the recipient of the reply. The same action is taken by a node that is a member of multicast group G when it receives a join-request for that group in the *Hello* from a neighbor stating that the node is the next hop to the core of the group. The core of a group G becomes a member of the group and starts sending a ‘member’ entry for group G in the *GIL* of its *Hellos* after receiving the first join-request for the group.

If node x is not a member of a multicast group G and receives a join-request from a neighbor stating the node as the next hop to the core of group G , then node x sends a join-request of its own in the *GIL* of its next *Hello*. Its join-request states its choice of next hop to the core of G along one of its implicit routes to the core.

A ‘member’ entry for group G in the *GIL* of the *Hello* from a node serves as an update of the multicast state for all its neighbors. It also helps nodes decide when to forward multicast data packets.

A multicast receiver that does not serve as a relay and chooses to leave a group sends a quit request to its current choice of next hop towards the core. A relay node that is a member of the group and receives a quit request creates its

own quit request if it is not a receiver and is not connecting any other neighbor to the core of the group. A node does not require a reply to a quit request and simply stops including an entry for group G in its *Hellos*.

F. Routing of Multicast Packets

Routing of multicast data packets from sources that are not part of the multicast tree of a node is accomplished by simple prefix routing. In this case, a source node forwards the packet to the neighbor whose prefix label provides the maximum prefix match for the group-prefix-label. The same applies to a relay node that is not on the shared multicast tree of the destination group.

If IP packets were used for forwarding at the network layer, the original multicast packet can be encapsulated into a unicast packet whose header states the same source identifier, and a destination identifier that corresponds to the next hop neighbor along the implicit path to the core of the group.

Routing of multicast packets by nodes that belong to the shared multicast tree of a given group is done in a similar fashion. If the node receives a multicast data packet for a group to which it belongs, the node retransmits the packet if it has on-tree neighbors (i.e., neighbors that have reported a ‘member’ or join-request action for the same group) other than the node from which the data packet is received, provided the packet has not been processed before.

G. Resilience to Topology Changes and Mobility

The prefix label of each node changes as nodes move around, and hence the path from a multicast receiver or relay node n to the core of a group G may break. However, the actions taken by nodes after detecting the need to change a multicast routing tree are always localized, because they are based on the prefix labels they are assigned in the LDAG of the network.

The LDAG is updated continuously with *Hellos*, and old prefix labels are kept for a label-transitioning period. This reduces the loss of packets because they are forwarded along an outdated prefix route. If the root of the LDAG fails, a new node is elected as the root in a way that priority to become root is given to nodes whose neighbors include the largest number of neighbors of the prior root. In addition, when a new root is elected, nodes keep their old prefix labels for an extended period of time.

Since node n knows the prefix labels and group memberships of all its neighbors, it can issue a join-request for group G as soon as it detects that it has lost connectivity to its next hop towards the core of group G (which includes the case of a new root forcing the relabeling of many nodes) and can select any neighbor that is a member of G and whose prefix label is closer to the group-prefix-label for G than its own prefix label. Similarly, a node n may decide to leave a group G after losing connectivity with neighbors it helped to connect with the core of the group.

If a node serving as the core of a group G were to fail or move away from the prefix label that is the best match for the

Fig. 3. Performance of PMR, ODMRP and MAODV with increasing pause times.

Fig. 4. Performance of PMR, MAODV and ODMRP with increasing number of concurrent sources.

group-prefix-label for G , then the node in the neighborhood that has the new best matching prefix label becomes the new core of the group. The only special action needed in this case occurs when this node, say node b , is not a current member of group G but has received *Hellos* from the previous core stating a *GIL* that includes a ‘member’ entry for group G , which means that there are active receivers for the group. In this case, node b becomes a member of group G and starts sending a ‘member’ entry for group G in the *GIL* of its *Hellos*. In contrast, the failure of a core in CBT, CAMP or PUMA would require network-wide signaling to elect a new node as the core of a group.

IV. PERFORMANCE EVALUATION

A. Simulation Setup

We compared PMR with MAODV and ODMRP, two of which are good examples of tree-based and mesh-based approaches and help evaluate PMR’s relative performance. We used a discrete event simulator called QualNet ver. 4.0 [12] for our simulation study.

Each of the simulations were instrumented with a network of nodes over a terrain 600m long and 900m wide. Nodes were placed randomly within a unit block of 5% the dimensions of the terrain and each of these unit blocks were distributed uniformly. We chose two distinct mobility models, random-waypoint and mobility traces from Dartmouth University Campus [5]. In the random-waypoint model, the way points were distributed over a convex region within the terrain to prevent abnormal occurrence of nodes with high degree. The

velocity distribution varied from 1 to 10 m/s uniformly and each node paused for a specific time interval selected from a random distribution with mean 30ms.

The radios were instrumented with a CSMA medium access control and the data rate, sensitivity and power were set to correspond to the chosen terrain in such a way that, under a static scenario, the entire network remained connected with minimal interference. Limiting the power of radios ensured that the radio-range was correspondingly low and did not cover the entire network. No path loss was simulated in the interest of simplicity of our study.

We simulated up to 100 nodes in these networks, and the number of active flows was set to 250 in each of these runs. The interval of each of these simulations were distributed exponentially with mean equal to 1/20th of the duration of simulation. Each protocol was also run at separate pause times from static to continuously mobile scenarios. The performance of all protocols was studied for different mobility rates with increasing pause times from 0 to the duration of the simulation (i.e., a static topology). The duration of the simulation was set to 300 sec. We ran the simulations over 10 random seeds and normalized these results in order to remove any topological artifact resulting from the node placement strategy.

All of the multicast routing protocols have scenario specific parameters that were made the same to ensure sufficient likeness for comparison. ODMRP and MAODV refreshed their routing mesh/trees every 3 seconds. Maximum group timeouts for join queries were set at 9 seconds.

In each of our experiments, we used three metrics in our

Fig. 5. Performance of PMR, MAODV and ODMRP with increasing number of groups, in a static topology and group mixes that contain a multiplicative increase in total members but divided randomly among different groups.

Fig. 6. Delivery ratio and control overhead of PMR, MAODV and ODMRP with the Dartmouth Campus trace mobility

comparison: The total number of control packets, the delay measured per flow from one end to the other, and the packet delivery ratio (defined as the number of data packets delivered per total number of messages sent).

B. Random Waypoint Mobility

In this scenario nodes move according to the random-waypoint mobility model. We evaluated the performance under different node speeds, and node pause times were varied from the case where all nodes were always moving to the static scenario. Each multicast group contained 20 members and sources of these groups sent 10 packets per second. The multicast groups were assigned unique global addresses. Figures 3(a) to 3(c) show the results for this scenario. We can see that PMR delivers 20% more than MAODV and 10% more than ODMRP while at the same time incurring 20% less overhead. Note that, for low pause-times of up to 50 seconds, PMR manages to deliver packets 3 to 4 seconds faster as well. While under higher-pause times, it is hard to beat network-wide floods in terms of latency, PMR managed to outperform both protocols in the combined performance of delivery, overhead and latency.

PMR performs better than MAODV and ODMRP, because of the reduced amount of control overhead generated by PMR, and in particular the absence of periodic network-wide flooding of control packets. In MAODV, whenever a receiver disrupts the shared-tree by leaving and joining at a different location, it floods the network with a RREQ. On the other-hand, in ODMRP, a multicast source floods the entire network

periodically with a join-query message. In contrast, disruptions due to mobility trigger very few updates in PMR, and control messages are propagated throughout the network. This reduces any congestion that may occur owing to network wide floods and improves delivery significantly.

C. Increasing The Number of Concurrent Sources

In this scenario, the node mobility was set to the random-waypoint mobility model, but the node speed was kept constant throughout the experiment by setting the pause time to 30 seconds. As the number of sources increase, the load on the network increases. With the same number of groups, each node receives a larger number of multicast packets. Figure 4(a) to 4(c) show the performance of the three multicast protocols for this scenario. It is apparent that the performance of ODMRP and MAODV degrade considerably as the number of sources increases, because of the corresponding increase in signaling traffic created by each source. In contrast, PMR generates far less control overhead and attains much smaller end-to-end delays and equal or better delivery ratios than the other two protocols.

D. Increasing Number of Multicast Groups

In this scenario, the topology of the network is static and the number of multicast groups is increased from 1 to 6. Each of these groups was divided such that the total members in the groups increased by a factor, corresponding to the number of groups. However, the size of each group was determined uniformly at random.

Therefore, in the first experiment with 1-group, the group size was set to 10 members. In the second experiment with 2-groups, the group size was set to 20 members, and the constitution of each of the group was determined randomly such that it totaled 20 members. In this case, group-1 was setup with 8 nodes and group-2 with 12. Similarly the number of groups were varied from 1 to 6. The average distances between each group member was set to vary from 3 hops to 7 hops. This helped us to observe the effect of having wide-spread group members vs. spatially co-located ones.

We observe from Figures 5(a) to 5(c) that ODMRP performs better for smaller groups that are spatially co-located. However, as the number of groups increases, its performance degrades independently of how the groups are located in the network. PMR outperforms both MAODV and ODMRP as the number of groups increases, and the performance of PMR is very similar for any number of multicast groups. These results should be expected given that the cores of groups do not generate any signaling traffic.

E. Dartmouth Campus Trace Mobility

Most realistic mobility scenarios differ from random mobility in that they follow a pattern and the rate of change of positions is slower. To address more realistic mobility patterns in our comparison, we used the mobility traces of the Dartmouth Campus from the CrawDad repository. Data for the traces were collected from access points deployed across the campus and were derived from the syslog traces. Associations with mobile laptops were used to determine location information and the movement history data set was then parsed into a mobility file for the QualNet simulator. We ran the simulations with a similar setup as the one described for the random-waypoint mobility scenario. While the trace files were for a larger number of nodes, a subset of 100 nodes was derived and simulated.

Figures 6(b) and 6(a) show the node speed in the x-axis and the delivery ratio and control overhead in the y-axis, respectively. All three protocols deliver a larger number of packets compared to what they deliver under the random-waypoint mobility scenario. PMR delivers 5 to 6% more packets than ODMRP and 25% more packets than MAODV. MAODV suffers from frequent broadcasts of network-wide RREQs as in the previous scenario, but as Figure 6(a) shows, its overhead starts to decrease at about 0.7m/s; this is because of the caching of RREQs seen in MAODV and the gratuitous RREPs generated by existing members of the multicast trees. On the other hand, ODMRP floods the network with the join-query packets, and the corresponding join replies are lost and retransmitted as nodes start to move around, which results in the overhead increase observed at around 0.7 m/s.

V. CONCLUSIONS

We presented PMR, which is the first multicast routing scheme for MANETs in which no network-wide dissemination of control information regarding multicast groups is needed. PMR achieves this by using prefix labels as an indirection

method in the establishment of multicast routing trees. All the signaling involved in PMR is localized to the neighborhood of a node, even after node or link failures disrupt multicast trees. Simulation experiments running in Qualnet were used to illustrate the performance advantages of PMR over two traditional approaches to mesh-based and tree-based multicasting in MANETs, namely ODMRP and MAODV. In this paper, we considered the use of multicast routing trees built using prefix labels and applied this technique to connected MANETs. Applying prefix labels to mesh-based multicasting and adapting this approach to accommodate network partitions and disruption-tolerant multicasting are the subject of our future work.

VI. ACKNOWLEDGMENTS

This work was supported in part by the U.S. Army Research Office (ARO) under grants W911NF-05-1-0246 and SC20070363, by the National Science Foundation under grant CNS-0435522, by DARPA through Air Force Research Laboratory (AFRL) Contract FA8750-07-C-0169, and by the Baskin Chair of Computer Engineering. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

- [1] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees (cbt). *SIGCOMM Comput. Commun. Rev.*, 23(4):85–95, 1993.
- [2] S. M. Das, H. Pucha, and Y. C. Hu. Distributed hashing for scalable multicast in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(3):347–362, 2008.
- [3] J. J. Garcia-Luna-Aceves and E. L. Madruga. The core-assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1380–1394, Aug 1999.
- [4] J. G. Jetcheva and D. B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *ACM MobiHoc '01*, pages 33–44, New York, NY, USA, 2001. ACM.
- [5] D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD trace set dartmouth/campus/movement (v. 2005-03-08). Downloaded from <http://crawdad.cs.dartmouth.edu/dartmouth/campus/movement>, Mar. 2005.
- [6] S. Lee and C. Kim. Neighbor supporting ad hoc multicast routing protocol. In *ACM MobiHoc '00*, pages 37–44. ACM, 2000.
- [7] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-demand multicast routing protocol. In *Proc. of the IEEE Wireless Comm. and Net. Conf., 1999. WCNC.*, pages 1298–1302 vol.3, 1999.
- [8] C. S. R. Murthy and B. Manoj. *Ad Hoc Wireless Networks*. 2004.
- [9] B. Parkinson and J. J. Spiker. *Global Positioning System: Theory and Applications, Volume 1*, volume 163. xx 1996.
- [10] E. Royer and C. Perkins. Multicast ad hoc on-demand distance vector (maodv) routing, 2000.
- [11] J. Sanchez, P. Ruiz, and I. Stojmenovic. Gmr: Geographic multicast routing for wireless sensor networks. In *IEEE SECON 2006*, 2006.
- [12] S. N. Technologies. Qualnet. <http://www.scalable-networks.com/>.
- [13] R. Vaishampayan and J. J. Garcia-Luna-Aceves. Efficient and robust multicast routing in mobile ad hoc networks. *IEEE MASS 2004*, pages 304–313, Oct. 2004.
- [14] X. Zhang and L. Jacob. Multicast zone routing protocol in mobile ad hoc wireless networks. *Local Computer Networks, Annual IEEE Conference on*, 0:150, 2003.