Multicasting Along Meshes in Ad-Hoc Networks

Ewerton L. Madruga and J.J.Garcia-Luna-Aceves
{madruga,jj}@cse.ucsc.edu
Computer Engineering Department
Baskin School of Engineering
University of California
Santa Cruz, CA 95064

Abstract— **The Core-Assisted Mesh Protocol (CAMP) is introduced for multicast routing in ad-hoc networks. CAMP generalizes the notion of core-based trees introduced for internet multicasting into multicast meshes that have much richer connectivity than trees. A shared multicast mesh is defined for each multicast group; the main goal of using such meshes is to maintain the connectivity of multicast groups even while network routers move frequently. CAMP consists of the maintenance of multicast meshes and loop-free packet forwarding over such meshes. Within the multicast mesh of a group, packets from any source in the group are forwarded along the reverse shortest path to the source, just as in traditional multicast protocols based on source-based trees. CAMP guarantees that, within a finite time, every receiver of a multicast group has a reverse shortest path to each source of the multicast group. It uses cores only to limit the traffic needed for a router to join a multicast group; the failure of cores does not stop packet forwarding or the process of maintaining the multicast meshes.**

I. INTRODUCTION

With few exceptions, the methods used today for supporting many-to-many communication (multicasting) efficiently in computer networks involve routing trees. The basic approach consists of establishing a routing tree for a group of routing nodes (routers). Once a routing tree is established for a group of routers, a packet or message sent to all the routers in the tree traverses each router and link in the tree only once. Multicast routing trees (multicast trees for short) are being used extensively for multicast routing in computer networks and internets [1], [8], [9], and have also been proposed for wireless multi-hop networks [2], [4].

The topology of an ad-hoc network can be very dynamic due to the mobility of routers and the characteristics of the radio channels. Although tree-based multicast routing is very attractive for wired networks and the Internet because of its simplicity, we argue that it is not as applicable to ad-hoc networks with dynamic topologies. Maintaining a routing tree for the purposes of multicasting packets when the underlying topology changes frequently can incur substantial control traffic. This paper focuses on multicast communication in ad-hoc networks and presents a generalization of routing trees into graphs that have more connectivity than trees and yet prevent long-term or permanent routing loops from occurring. We call these routing graphs *multicast meshes*. The key contributions of this paper consist of proving that it is possible to establish and maintain routing structures for multi-point communication in an ad-hoc network that are far more resilient than trees and can make efficient use of communication resources, without the need to first flood an entire network or internet with either data packets (like DVMRP or PIM-DM do), or control packets (like the Forwarding Group Multicast Protocol (FGMP) does [3]).

Section II introduces the main design principles of the Core-Assisted Mesh Protocol (CAMP), which builds and maintains shared multicast meshes, and routes packets from any group source over the shortest from source to receivers defined in the group's mesh. Section III to IV describe in more detail the creation and maintenance of multicast meshes in CAMP. Section V describes the results of simulation experiments used to study CAMP's performance compared to tree-based multicasting and a different mesh approach in a dynamic topology. Section VI provides our concluding remarks.

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Grant F30602-97-2-0338, by the University of California under a MICRO grant sponsored by Raytheon and by CNPq-Brazil.

II. OVERVIEW OF CAMP

CAMP [11] differs from most prior multicast routing protocols in that it builds and maintains a *multicast mesh* for information distribution within each multicast group. A multicast mesh is a subset of the network topology that provides at least one path from each source to each receiver in the multicast group. CAMP ensures that the shortest paths from receivers to sources (called reverse shortest paths) are part of a group's mesh. Packets are forwarded through the mesh along the paths that first reach the routers from the sources, i.e., the shortest paths from sources to receivers that can be defined within the mesh. CAMP does not predefine such paths along the mesh. A router keeps a cache of the identifiers of those packets it has forwarded recently, and forwards a multicast packet received from a neighbor if the packet identifier is not in its cache. The key difference between a mesh and a tree structure is how data packets are accepted to be processed. A router is allowed to accept unique packets coming from any neighbor in the mesh, as opposed to trees where a router can only take packets coming from routers with whom a *tree branch* has been established. Therefore, keeping the branch information updated is one extra challenge protocols based on trees have to face in a mobility scenario.

Because a member router of a multicast mesh has redundant paths to any other router in the same mesh, topology changes are less likely to disrupt the flow of multicast data and to require the reconstruction of the routing structures that support packet forwarding. Figure 1 illustrates the differences between a multicast mesh and the corresponding shared multicast tree; routers that are members of the multicast group are dark. The multicast mesh and tree shown in the figure include routers that have host receivers, hosts that are senders and receivers, and routers that act only as relays. Router g is the last receiver to join the multicast group, and does so in the multicast mesh through either router f or h; consequently, router c does not become a member of the mesh.

CAMP extends the basic receiver-initiated approach introduced in the core-based tree (CBT) protocol [1] for the creation of multicast trees to enable the creation of multicast meshes. Cores are used to limit the control traffic needed for receivers to join multicast groups. In contrast to CBT, one or multiple cores can be defined for each mesh, cores need not be part of the mesh of their group, and routers can join a group even if all associated cores become unreachable.

A host first determines the address of the group it needs to join as a receiver. The host then uses that address to ask its attached router to join the multicast group using IGMP [7]. Upon receiving a host request to join a group, the router sends a join request towards a core if none of its neighbors are members of the group; otherwise it simply announces its membership using either reliable or persistent updates. If cores are not

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1999**	2. REPORT TYPE	3. DATES COVERED **00-00-1999 to 00-00-1999**
4. TITLE AND SUBTITLE **Multicasting Along Meshes in Ad-Hoc Networks**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**5**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

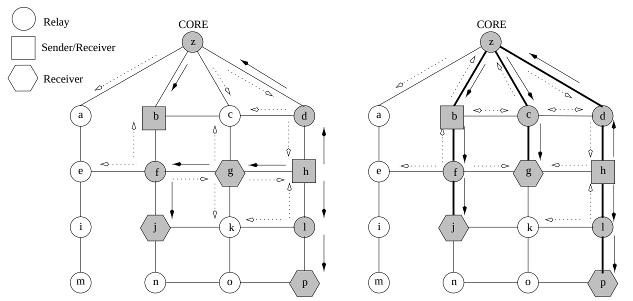


Fig. 1. Traffic flow from router h in a multicast mesh (left) and in the equivalent multicast shared tree (right).

reachable from a router that needs to join a group, the router broadcasts its join request using an *expanded ring search* (ERS) that eventually reaches some group member. When one or multiple responses are sent back to the router, it chooses any of these responses to use as a path to the mesh. The mappings of multicast addresses to (one or more) core addresses are disseminated from each core out to the network as part of group membership reports.

The Core-Assisted Multicast Routing Protocol provides also an alternate way for routers connected to sender-only hosts to join the mesh. Whenever a router senses multicast packets originated at a host directly attached to it, this designated router will join the mesh in *simplex* mode if it's not a member yet. The simplex join request, just as a regular duplex join request, will travel towards one of the available cores and is acknowledged in the same fashion. The conceptual difference is that data packets should travel in only one direction: from the sender-only host to the mesh and not the opposite. This is an attempt to contain data traffic closer to the areas of the mesh where receivers are present. A router can leave the group when there are no other hosts or routers depending on it simply by advertising the change in group membership to their neighbors. More details about simplex joins as well as the handling of topology changes are presented in previous work [11].

A router leaving a multicast group issues a *quit notification* to its neighbors, who in turn can update their data structures. No acknowledgments are requested for quit notifications, because in contrast to multicast routing trees, multicast meshes do not dictate the paths taken by multicast packets. Quit notifications are sent as part of multicast routing updates.

The Forwarding Group Multicast Protocol (FGMP) [3] and the On-demand Multicast Routing Protocol (ODMRP) [13] also build a variation of meshes. However, to establish group meshes, they require for control packets to be flooded in an ad-hoc network. The difference between these two protocols is who starts the flooding — in the former, the receivers, and in the latter, the senders. This approach is acceptable only in small networks. In contrast, the use of cores in CAMP eliminates the need for flooding, unless all cores are unreachable from a connected component.

III. ROUTING INFORMATION AND DATA FORWARDING IN CAMP

Each router maintains a routing table (RT) built with the unicast routing protocol. This table is also modified by CAMP when multicast groups need to be inserted or removed. CAMP assumes the existence of a *beaconing* protocol, usually embedded into the unicast routing protocol or available as a separate network service.

At router i, the RT made available to CAMP specifies, for each destination j, the successor (s^i_j) and the distance to the destination (D^i_j).

Other than the unicast routing table, CAMP relies on these data structures:

- CAM^g_i : table mapping cores to multicast groups.
- $CORES_g$: set of routers acting as cores to multicast group g.
- $CACHE_i$: cache of multicast data packet control information.
- MRT_i : the multicast routing table, containing the set of groups known to router i.
- AT^g_i : table containing anchor information pertaining to router i. This table is split in two subsets:
  - A^g_i : list of neighbors that have router i as their anchor for multicast group g.
  - $A2^g_i$: list of neighbors who are anchors to router i in multicast group g.
- N^g_i : router i's list of neighbors that are known to be members of the multicast group g.
- LS^g_i : list of senders that are directly attached to router i and send data traffic to multicast group g.
- LR^g_i : list of receivers directly attached to router i, who want to receive data packets from multicast group g.
- $PEND^g_i$: list of either join or simplex join requests to multicast group g originated at or forwarded by router i for whom acknowledgment is pending.
- $PENDPJ^g_i$: list of push join requests to multicast group g originated at or forwarded by router i for whom acknowledgment is pending.
- BK^g_i : list used for periodic "book-keeping" of senders and associated anchors.

The packet-forwarding cache $CACHE_i$ maintains information of packets recently processed by router i. The main role of the packet forwarding cache is to avoid packet replication by keeping track of packets already received by the router. Caching packets is only feasible for low-bandwidth channels. Although restricted to symmetric networks, an alternative to packet caching is the use of reverse path forwarding [6], where routers only accept data packets from their successor to the packet source.

The table AT_i has an entry for each of the multicast groups in which router i is a member. For each multicast group g, an entry in the AT specifies those neighbors that router i uses as its *anchors* for the group, and whether the router has any local host that is a source or receiver of the group. An anchor for router i in group g is a neighbor router that is a successor (next hop) in the reverse shortest path to *at least one source* in the group g. Therefore, a router determines its anchor to a given source by using the unicast routing table. In the example shown in Figure 1, router f uses router g as an anchor for the group because of source h, if g is the next hop to h in RT. Note that a router does not maintain anchor information for each source in a group.

When MRT_i or AT_i is updated, router i sends a multicast routing update (MRU) to all its neighbors reporting changes in its group membership and anchors per group. An MRU contains one or more entries, and each entry specifies a multicast group address, an op-code (quit notification or duplex/simplex membership notification) and, when update includes a membership notification, a list of anchors the router depends on for this multicast group. The main objective of communicating anchor information among routers is to prevent routers that are required by their neighbors to forward multicast packets from leaving groups prematurely.

Detecting the failure or addition of a link to a neighbor is part of the routing protocol used in conjunction with CAMP. For CAMP to work correctly, it is necessary for the associated routing protocol to work correctly in the presence of router failures and network partitions. This implies that CAMP cannot be used in conjunction with a routing protocol based on the distributed Bellman-Ford algorithm such as the routing protocol of the DARPA packet radio network [14]. However, there are several recent examples of routing protocols that can be used in conjunction with CAMP [5], [15], [16].

The basic packet forwarding scheme in CAMP consists of trying to forward multicast data packets along the paths within the mesh that first reach the member routers from the sources. A router receiving a multicast packet without errors from a neighbor router accepts the packet only if:

- The router is a member of the multicast group specified in the packet, which is determined from the router's MRT.
- If the router is a duplex router, the packet's sequence number is not in the packet-forwarding cache.
- if a simplex router, the packet's sequence number is not in the packet-forwarding cache and the neighbor sending the packet is also a simplex router.

When a router accepts a packet, it adds its sequence number and the identifier of the source to its packet forwarding cache. This step prevents the same packet from being accepted more than once by the router, provided that the entries in the cache persist longer than the time it takes for packets to revisit a router.

IV. Heartbeats and Push Joins

CAMP ensures that all the reverse shortest paths between sources and receivers are part of a group's mesh by means of *heartbeat* and *push join* (PJ) messages.

Periodically, every single entry in the packet forwarding cache is verified. The router looks up its RT to check whether the neighbor that relayed the packet is the reverse path to the source for every cache entry. A heartbeat or a PJ is sent towards every source stored in the cache that had the number of packets coming from the reverse path under a threshold.

CAMP determines two types of push join acknowledgments — regular ACK, sent by duplex members and ACK_SIMPLEX, sent by simplex members. Given the fact that simplex mesh members do not accept packets coming from duplex members, it's important that there's no interleave of duplex and simplex routers between the initiator of a push join request and the router directly attached to the source. When acknowledgments start coming back from the source, duplex members will always send regular ACKs, and simplex members will become duplex when they receive a regular ACK. Therefore, if there's at least one duplex mesh member in the path from initiator to the source, all nodes from that duplex member all the way to the initiator must become duplex if they're not yet.

V. Performance Comparison

A. Protocols Used for Comparison

In large ad-hoc networks, no multicast protocol proposed to date that is based on sender-initiated joining is scalable with the number of nodes in the network or the number of sources and groups in the network. Examples of this type of protocols based on routing trees are DVMRP and PIM-DM; an example of this type of protocols based on graphs other than trees is FGMP [2]. The reason that these protocols are not scalable is that sources must flood either data packets or control packets to *all* the network in order to establish a routing structure. If the network size is large, or the number of groups and sources per group is large, this approach is not applicable.

To date, CAMP is the only multicast routing protocol not based on trees that avoids flooding of data or control packets to establish the routing structure for a group. Therefore, for comparative purposes, we implemented a simple tree-based protocol that can be used to capture all the features of the main tree-based multicast protocols with receiver-based joining proposed or implemented to date. Also implemented was

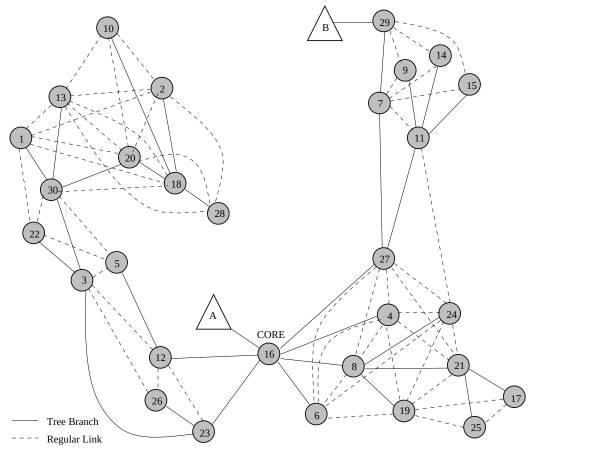


Fig. 2. Network topology used in experiments: Lines connecting nodes indicate the initial tree obtained with CBT, all routers are receivers of the multicast group, Router 16 is the core, and A and B are sources.

the On-demand Multicast Routing Protocol (ODMRP) [13]. The objective of the simulation experiments is to illustrate that the mesh approach used by CAMP is more robust than shared tree structures and is more scalable than the flood-based mesh approach used in ODMRP and FGMP.

We implemented a shared-tree multicast routing protocol that is similar to CBT in that it uses a single core and uses that tree to forward packets. A router in this protocol, which we denote by WTP (wireless tree-based protocol), forwards data packets only when they come from one of the children or parent of the router in the tree rooted at the core. The tree maintenance part of WTP extends the conventional shared-tree protocols like CBT and PIM-SM. In WTP, a router re-establishes its connection to the tree by looking for a new parent as soon as it detects that its previous parent has moved away.

B. Experiments

The interesting aspects for performance comparison between CAMP and the other multicast protocols are the average delays, percentage of packet loss incurred due to node mobility, and the number of control packets received by each node. The percentage of packets lost at a receiver is simply the amount of packets sent by the traffic source that was not seen by the specific receiver. Therefore, the smaller the percentage is, the better the protocol behaves. Obviously, the average packet delay measured at each receiver excludes lost packets.

We ran a number of experiments to study this aspect of CAMP's performance and to compare it against the other multicast approaches. Figure 2 shows the topology of the dynamic network used in the simulations. The network has 30 routers, numbered from 1 to 30, and two senders, A and B. The solid links shown in the diagram illustrate the initial shared tree computed dynamically in the simulation. The dashed links represent the connectivity among nodes. All nodes in the simulation of the multicast routing protocols are receivers. In CAMP, this means all nodes are *duplex* members. *Router 16* was chosen as core for all simulations.

Experiments ran for 350 seconds and the same conditions were applied to the simulation runs for CAMP, WTP and ODMRP; specifically, the same number of packets was sent from the given source, the same pattern of router mobility was applied, and the same MAC and routing protocols were used. The simulations used a single broadcast channel, so that the transmission of a node is received by all its neighbors. The floor acquisition multiple access (FAMA) protocol[10] was used to access the broadcast channel, and the wireless internet routing protocol

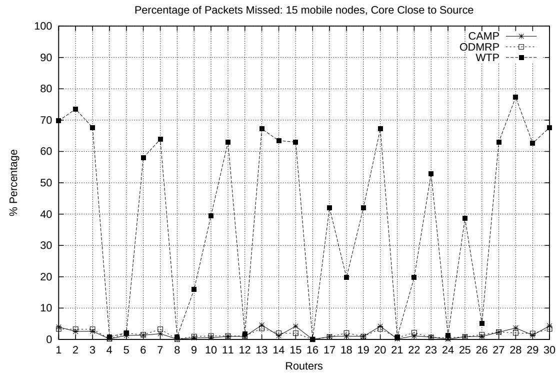


Fig. 3. Number of packets lost by routers when 15 nodes are mobile, and traffic comes from source A, directly connected to the core.

(WIRP) [12] with *hop count* as distance metric was used to generate the unicast routing-table entries at routers for CAMP and WTP. ODMRP does not need a unicast routing protocol. Radio links are bidirectional. The timers of updates in CAMP and sender advertisement in ODMRP determine how fast the network adapt to topology and group membership changes. They are both set to three seconds.

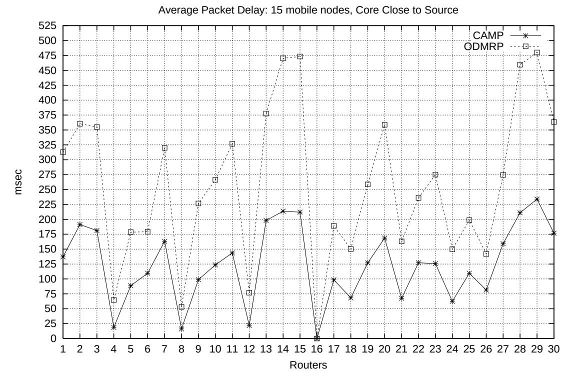
A number of experiments were run regarding mobility. For the sake of brevity, the results will be illustrated only by the experiments where 15 routers were moving through the network. The mobile nodes are routers 1, 3, 6, 7, 9, 10, 17, 18, 19, 20, 23, 25, 27, 28 and 30. The speed at which mobile nodes moved randomly in all simulations was 67.5 miles/hr (30 meters/sec).

In the experiments, data traffic is originated either by source A, which is directly attached to the core (*router 16*), or by both source A and B, which is attached to *router 29*. In the experiments where the source of data traffic is sender A, the load was 4 packets/second. In the experiments where both senders A and B transmitted packets, each one sent 2 packets/second to try to keep the same number of data packets in the network.

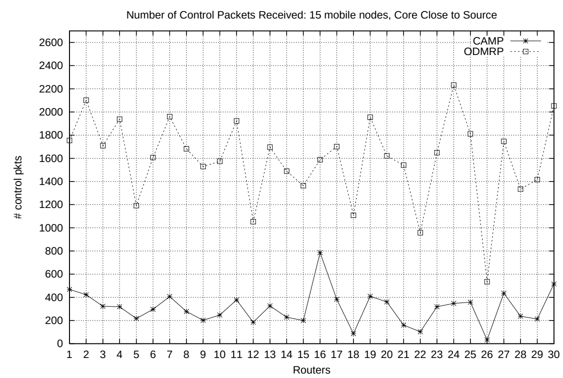
Not surprisingly, WTP was the protocol that performed the worst in the experiments. Figure 3 shows the different outcome between WTP and the mesh-based protocols regarding packet losses. WTP attempts to reconnect the tree as soon as possible every time a router loses its parent in the shared tree. Every time the unicast routing protocol warns WTP about a neighbor being removed from the unicast routing table, the protocol sends a join request to the new successor to the core, trying to re-establish its connection to the tree. The same trend shown in Figure 3 for packet losses was observed in all experiments we ran. In such a context, the comparison of average packet delays between the shared-tree protocol and the mesh-based protocols cannot be made, since the averages for the routers running WTP is computed based in much less data packets than in CAMP and ODMRP. Therefore, for the sake of brevity, we do not include WTP results in the following figures.

The reason for the poor behavior of WTP is the strong dependency it has on consistent unicast routing tables to provide a loop-free shared tree. WIRP [12], the unicast routing protocol used in the experiments, may create temporary loops shortly after links go down. Because WTP makes decisions regarding tree reconnection shortly after links go down, the shared tree becomes vulnerable to loops, which leads to the larger packet-loss rate. This fact shows the difficulties brought up when packet forwarding is dictated by a strict delivery structure like a shared tree in a dynamically changing environment. Protocol behavior in the presence of temporary loops in unicast routing also illustrates the survivability of mesh protocols.

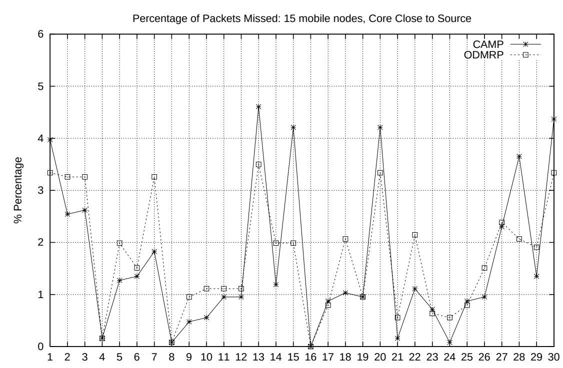
Figures 4 and 5 summarize the comparison between CAMP and



(a)



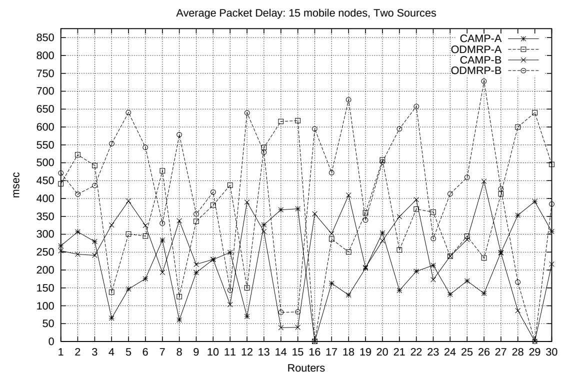
(b)



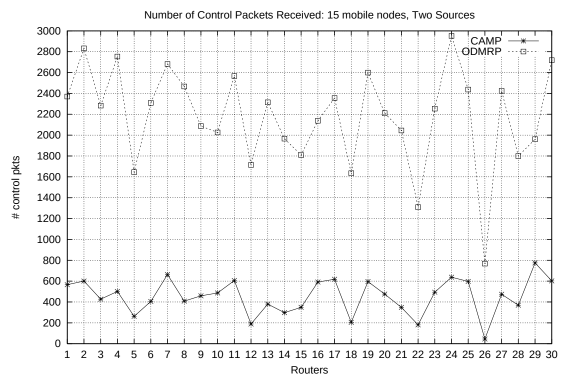
(c)

Fig. 4. Average packet delay (a), number of incoming control packets (b) and percentage of missed data packets (c) for routers in a network of 30 nodes, where 15 of these nodes are mobile. Data traffic from source A, which is the one close to the core.

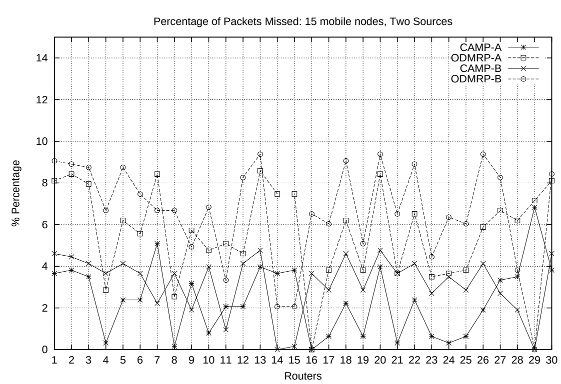
ODMRP over the different experiments we have run. Dotted lines represent ODMRP and solid ones represent CAMP. Figure 4(a) shows that CAMP renders smaller delays than ODMRP in the case of a single source sending 4 packets/second and 15 nodes moving in the network. And the main reason for this difference in average is shown in Figure 4(b). The longer delays incurred in ODMRP is a consequence of



(a)



(b)



(c)

Fig. 5. Average packet delay (a), number of incoming control packets (b) and percentage of missed data packets (c) for routers in a network of 30 nodes, where 15 of these nodes are mobile. Data traffic from both sources A and B. In this figure, "CAMP-A" represents the results for routers running CAMP due to traffic from Source A.

the flooding of control packets per source needed in ODMRP. The number of control packets received by CAMP routers is just a small fraction of the number seen by ODMRP routers. For the same traffic load and mobility patterns, both protocols perform similarly, when there's one source of data traffic, as shown by Figure 4(c). As the number of sources grows, CAMP performs even better than ODMRP, as shown in Figure 5. In Figure 5(a), one can observe that, like routers 1, 2 and 3, almost half of the routers in the network show shorter delays for both senders A and B when running CAMP. As illustrated by Figure 5(c), as far as packet losses are concerned, CAMP loses consistently fewer packets when more than one source send data packets. Those aspects of the protocols' performance illustrate that meshes can be used effectively as multicast routing structures without the need for flooding of control packets.

VI. Conclusions

We have introduced the first multicast routing protocol based on a routing structure other than trees that does not require flooding an entire network with control or data packets to set up its routing structure. CAMP consists of the maintenance of multicast meshes and loopless packet forwarding over such meshes. Within the multicast mesh of a group, packets from any source in the group are forwarded along the shortest paths defined with the mesh from the source to the receivers. Simulation experiments show that mesh-based protocols easily outperform tree-based multicast protocol in dynamic networks. Experiments show that CAMP scales very well, because it does not require sources or receivers to flood the entire network with control or data packets as long as there are cores available. Our comparison with ODMRP shows that the receiver-initiated approach used for mesh joining in CAMP performs and scales better than the sender-initiated approach.

References

[1] A. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing," *Proc. ACM SIGCOMM'93*, pages 85–95, October 1993.

[2] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Adhoc Multicast Routing Protocol," Internet Draft, http://www.ietf.org/internet-drafts/draft-talpade-manet-amroute-00.txt

[3] C. Chiang and M. Gerla, "On-Demand Multicast in Mobile Wireless Networks," *Proc. IEEE ICNP 98*, Austin, Texas, October 14-16, 1998.

[4] M.S. Corson and S.G. Batsell, "A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Overview of Initial Route Construction," *Proc. IEEE INFOCOM '95*, Boston, MA, April 1995.

[5] M.S. Corson and V. Park, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[6] Y.K. Dalal and R.M. Metcalfe, "Reverse Path Forwarding of broadcast packets." *Communications of the ACM*, 21(12):1040-1048, December 1978.

[7] S. Deering, "Host extensions for IP multicasting," RFC-1112, August 1989.

[8] S. Deering, C. Partridge and D. Waitzman, "Distance Vector Multicast Routing Protocol," RFC-1075, November, 1988.

[9] S. Deering et al. "An Architecture for Wide-Area Multicast Routing," *Proc. ACM SIGCOMM'94*, Cambridge, MA, 1994.

[10] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proc. ACM SIGCOMM 97*, Cannes, France, September 14-18, 1997.

[11] J.J. Garcia-Luna-Aceves and E. Madruga, "A Multicast Routing Protocol for Ad-Hoc Networks", *Proc. IEEE INFOCOM'99*, New York, NY, June, 1999.

[12] J.J. Garcia-Luna-Aceves, C.L. Fullmer, E. Madruga, D. Beyer and T. Frivold, "Wireless Internet Gateways (WINGS)", *Proc. IEEE MILCOM'97*, Monterey, California, November 2–5, 1997.

[13] M. Gerla, et al., "On-Demand Multicast Routing Protocol," Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-odmrp-00.txt, November, 1998.

[14] G.S. Lauer, "Packet Radio Routing," *Chapter 11. Routing in Communications Networks.* (M.Streenstrup, ed.), Prentice-Hall, Englewood Cliffs, NJ, 1995.

[15] C.E. Perkings and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers." *Proc. ACM SIGCOMM'94*, London, UK, 1994.

[16] M. Spohn and J.J. Garcia-Luna-Aceves, "Scalable Link-State Internet Routing," *Proc. VI IEEE International Conference on Network Protocols (ICNP'98)*, Austin, TX, October, 1998.