Distributed Hashing for Scalable Multicast in Wireless Ad Hoc Networks

Saumitra M. Das, *Member*, *IEEE*, Himabindu Pucha, *Member*, *IEEE*, and Y. Charlie Hu, *Senior Member*, *IEEE*

Abstract—Several multicast protocols for mobile ad hoc networks have been proposed, which build multicast trees by using location information that is available from the Global Positioning System (GPS) or localization algorithms and use geographic forwarding to forward packets down the multicast trees. These *stateless* multicast protocols carry encoded membership, location, and tree information in each packet and are more efficient and robust than stateful protocols (for example, ADMR and ODMRP), as they avoid the difficulty of maintaining distributed state in the presence of frequent topology changes. However, current stateless multicast protocols are not scalable to large groups because of the per-packet encoding overhead, and the centralized group membership and location management. We present the Hierarchical Rendezvous Point Multicast (HRPM) protocol, which significantly improves the scalability of stateless multicast with respect to the group size. HRPM consists of two key design ideas: 1) hierarchical decomposition of a large group into a hierarchy of recursively organized manageable-sized subgroups and 2) the use of distributed geographic hashing to construct and maintain such a hierarchy at virtually no cost. Our detailed simulations demonstrates that HRPM achieves significantly enhanced scalability and performance due to hierarchical organization and distributed hashing.

Index Terms—Wireless networks, multicast, mobile ad hoc networks, scalable routing, hashing.

1 Introduction

A mobile ad hoc network (MANET) consists of a collection of wireless mobile nodes that dynamically form a temporary network without the use of any existing network infrastructure or centralized administration. In such a network, since nodes are often not within the radio transmission range of each other, each node operates not only as a host but also as a router, forwarding packets for other mobile nodes.

Multicast is a fundamental service for supporting collaborative applications among a group of mobile users [1]. Unlike in the wired Internet, multicast in MANETs is faced with a more challenging environment. In particular, multicast in MANETs needs to deal with node mobility and, thus, frequent topology changes, a variable quality wireless channel, constrained bandwidth, and low memory and storage capabilities of nodes. Additionally, unlike in the wired Internet, nodes in a MANET can be modified at the network layer to provide group communication support. This reduces the need for overlay-based group communication that has been popular in the Internet.

Numerous multicast protocols have been proposed for multicast in MANETs. These include traditional tree-based or mesh-based protocols such as MAODV [2], ADMR [3], and ODMRP [4]. Some multicast protocols use an overlay-based approach such as AMRoute [5] and PAST-DM [6]. Finally, certain protocols such as MCEDAR [7] are backbone-based protocols. More recently, stateless protocols such as DDM [8], HDDM [9], and RDG [10] have also been proposed. All these protocols either rely on underlying unicast routing schemes (for example, [5] and [8]) or expend great effort to maintain a distributed multicast routing structure (for example, [3] and [4]). Both factors affect the scalability of these protocols.

Recently, several location-based multicast protocols for MANETs have been proposed [11], [12], [13], which neither assume any unicast routing scheme nor build any distributed multicast routing structure. These protocols build multicast trees by using the location information available from the Global Positioning System (GPS) [14] and use geographic forwarding to forward packets down the multicast trees. Sharing the stateless nature of geographic forwarding, these protocols are *stateless*, as they carry encoded membership and location, and tree information in each packet so that the multicast membership and routing state do not have to be distributed as in traditional multicast protocols such as ADMR and ODMRP. Stateless protocols are more robust and potentially more efficient than stateful protocols, as they avoid the difficulty of maintaining distributed state in the presence of frequent topology changes in MANETs. Their efficiency over stateful protocols is more pronounced when the multicast group has a sparse membership, in which case the stateful protocols have to employ a high percentage of nonmember forwarding nodes (that is, large state) to maintain the tree or mesh.

However, because of their stateless nature, previous location-based multicast protocols suffer from limited scalability in terms of the group size. Conceptually, stateless location-based multicast protocols are not scalable to large groups, because they encode group membership in the

- *The authors are are with the Center for Wireless Systems and Applications, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907. E-mail: {smdas, hpucha, ychu}@purdue.edu.*

Manuscript received 18 May 2006; revised 11 Dec. 2006; accepted 10 Apr. 2007; published online 3 July 2007.
Recommended for acceptance by I. Stojmenovic.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0130-0506.
Digital Object Identifier no. 10.1109/TPDS.2007.70731.

1045-9219/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

header of each data packet. In fact, previous location-based protocols are explicitly proposed for small groups.

In this paper, we study the scalability aspect of location-based multicast, in particular the group (membership and location) management in location-based multicast protocols. A well-known general approach for reducing the load of managing a large multicast group is to partition the large group into hierarchically organized subgroups of manageable sizes. The immediate consequence of distributing membership management is that the protocol becomes stateful. Therefore, the key question here is whether there is a way to leverage the concept of hierarchical membership management *without* incurring the high cost associated with maintaining a distributed state in mobile nodes.

We present the Hierarchical Rendezvous Point Multicast (HRPM) protocol, which meets the above criterion and significantly improves the scalability in the group size of previous location-based multicast protocols. HRPM leverages two key techniques: distributed *mobile geographic hashing* and *hierarchical decomposition of multicast groups*. Given a data item and a location, mobile geographic hashing maps (routes) the data item to the node whose geographic location is *currently* closest to the location. Thus, mobile geographic hashing allows multicast group members to agree upon a fixed rendezvous point (RP; and the current node associated with it) as the group manager (root) without incurring any overhead, for example, in keeping track of an otherwise mobile group root. This, in turn, allows the multicast protocol to maximally leverage stateless geographic forwarding for efficient group membership and location management. Furthermore, mobile geographic hashing can recursively be used to enable a hierarchy of subgroups that are of manageable size. The manageable size allows the multicast inside each subgroup to satisfy a per-packet tree-encoding overhead constraint. Group management under such a hierarchy is extremely lightweight, as the RP subgroup roots are effectively "stationary." This property of virtual stationarity allows HRPM to effectively avoid the high cost that is associated with maintaining a distributed state in mobile nodes.

We first study the performance of HRPM as compared to previously proposed location-based multicast protocols. The results demonstrate that for large groups (up to 250 members), HRPM significantly improves the scalability of previous location-based multicast protocols.

We then compare HRPM to ODMRP, a topology-based multicast protocol that is scalable to large groups. In this comparison, we find that HRPM is comparable to ODMRP in performance as the group size increases. However, HRPM significantly outperforms ODMRP as the network size is increased (up to 1,000 mobile nodes). In addition, HRPM outperforms ODMRP when a large number of groups (up to 45) or a large number of sources per group exist.

In summary, leveraging stateless geographic forwarding for data delivery and distributed hashing for group and location management allows HRPM to scale well in terms of the group size, the number of groups, the number of sources, and the size of the network.

The rest of this paper is organized as follows: Section 2 formulates the location-based multicast problem. Section 3 presents the detailed design of HRPM. An analysis of the key design parameters of HRPM is presented in Section 4. Section 5 presents the simulation studies. Section 6 summarizes related work, and finally, Section 7 concludes the paper.

2 Preliminaries

The multicast problem deals with the transmission of information from a node to all members of a group while optimizing a certain application-specific metric such as bandwidth cost or delay. In a MANET with positioning systems such as GPS [14], each node can determine its own geographic location. Location information can be exploited to provide location-based multicast. These protocols encode the membership and tree information in each packet so that the membership and the forwarding state are not distributed as in multicast protocols such as ADMR or ODMRP. In the following, we discuss the three components of a location-based multicast protocol for MANETs:

1. *Group membership and location management.* An efficient scheme for the management of group membership and locations is critical to the efficiency and scalability of location-based multicast, since nodes are continuously moving in a MANET. To manage the group membership, group members can multicast their membership/locations to all other group members [12] or send their updates to an agreed-upon root so that the group members can then contact the root to obtain updated information. Moreover, the location of either the group members [12] or all the nodes in the network [11] are required, depending on the nature of the multicast tree used.
2. *Multicast tree construction.* Once the group membership and location information are obtained, the source of the multicast can construct a multicast tree by using either an overlay tree [12] that consists of only group member nodes or a physical tree [11] that consists of group member nodes and other nodes en route between the member nodes. Many graph algorithms exist for the construction of such multicast trees. These tree construction algorithms exploit the correlation between geometric distance and network distance (the number of routing hops) that longer geometric distance implies more network hops [12], and they use geographic distances between nodes as edge weights.
3. *Data delivery.* The data delivery mechanism depends on the nature of the tree and the location/member management scheme used. A physical tree can efficiently be encoded at the header of a data packet. Such data packets can be delivered via source routing [11], as the tree contains all the intermediate nodes. In case of an overlay multicast tree, based on the group/location management scheme, there can be two approaches to data delivery: 1) If the locations of the group members are known only to the source of the multicast tree, the destinations and the locations of the group members need to be encoded in the packet header at the source. And, 2) if every group member

knows every other group member's location, only the destinations are encoded in the packet header (since each intermediate overlay node can fill in the locations and decide how the packet could be forwarded). This reduces the per-packet encoding overhead. However, this requires intermediate overlay nodes in the tree to acquire such location information via other means, for example, updates directly from the destination nodes. Moreover, in case of an overlay multicast tree, as the tree members may not be within direct reach of each other, geographic forwarding is needed to deliver data packets along the overlay links.

In this paper, we use a greedy geographic forwarding algorithm as the routing protocol. Each node periodically announces its IP address and location to its one-hop (within the radio transmission range) neighbors by broadcasting BEACON packets. Each node maintains the IP and location information of its neighbors. Each packet being routed contains the destination address in the IP header and the destination's location (x and y-coordinates) in an IP option header. To forward a packet, a node consults its neighbor table and forwards the packet to its neighbor that is closest in geographic distance to the destination's location. Note that the above greedy geographic forwarding can lead to a packet reaching a node that does not know any other node that is closer to the destination than itself. This indicates a hole in the geographical distribution of nodes. Recovering from holes can be achieved using *face routing* (first proposed in [15] and extended in GPSR [16] and GOAFR+ [17]).

3 Hierarchical Rendezvous Point Multicast

In this section, we describe the design of HRPM. HRPM incorporates two key design concepts: 1) using hierarchical decomposition of multicast groups and 2) leveraging geographic hashing to efficiently construct and maintain such a hierarchy.

Hierarchical routing [18] is a well-known approach to reducing the protocol states in a large scale network. The per-packet encoding overhead of a stateless location-based multicast protocol grows with the group size as $O(G)$, where G is the multicast group size. Thus, an increase in G severely limits the usability of such protocols. The main design goal of HRPM is to limit the per-packet overhead to an application-specified constant (ω), irrespective of the increase in G. The value of ω is a parameter of HRPM and can be adjusted based on the amount of overhead that can be tolerated by an application. To achieve this, HRPM recursively partitions a large multicast group into manageable-sized subgroups, in which the tree-encoding overhead satisfies the ω constraint. This partitioning is achieved by geographically dividing the MANET region into much smaller *cells*. Such cells form a hierarchy, with the root representing the entire region. Every cell in the hierarchy has an *Access Point* (AP), and the entire region has an *RP*. All members in a leaf cell of the hierarchy form a subgroup and are managed by that cell's AP. Groups of APs are recursively managed, that is, by the APs of their parent cells. ω is an application parameter, and we discuss how HRPM adjusts the hierarchy to meet this ω constraint in Section 4.

The fact that both RPs and APs are logical entities is central to the design of HRPM. If such a logical entity is associated with a specific node (IP address), keeping track of the RP/AP would require an external location service or some flooding-based mechanisms due to mobility in MANETs. This can potentially incur high overhead. To avoid such overhead, HRPM disassociates the RP/AP from any specific node by adopting the concept of *geographic hashing* that was previously proposed for data storage in static sensor networks [19]. Given a data item, geographic hashing maps that data item to a geographic location (x, y), whereas geographic routing is then used to route the data item to the node whose geographic location is closest to (x, y). Since in MANETs, different mobile nodes can become the closest to a fixed location over time, *mobile geographic hashing* in HRPM extends geographic hashing via a continuous handoff process, which ensures that the data item is always stored on the node that is currently closest to the location. Thus, if the members of a group/subgroup use an agreed-upon hash function to hash the multicast group identifier (GID) and obtain the RP/AP location for the group/subgroup, all group management messages can be routed to the RP/AP by leveraging geographic forwarding.

In the following, we describe the details of HRPM group management, tree construction, and data delivery.

3.1 Group Management

We first introduce the concept of RP group management (RPGM), which assumes a flat geographic domain. We then introduce the hierarchical domain decomposition of a multicast group and describe how RPGM can recursively be applied in a hierarchy of subdomains.

3.1.1 Rendezvous Point Group Management

RPGM allows multicast group members to leverage geographic hashing for efficient group management. Fig. 1a shows RPGM in a flat geographic domain. Any node that wants to join a multicast group first hashes the GID to obtain the RP's location in the physical domain of the network using a hash function:

$$H(GID) = (x, y) \; where \; x, y \in MANET \; region.$$

This hash function takes the GID as input and outputs a location (x and y-coordinates) contained in the region. Note that we assume that this is a well-known hash function that is known by nodes that enter the network through external means or by using some resource discovery processes.

After obtaining the hashed RP location for the group that it wants to join, the node sends a JOIN message that is addressed to this hashed location. This JOIN message is routed by geographic forwarding to the node that is currently closest to the hashed location in the network. This node is the designated RP at this time. Since there is only one such node at any given time, the JOIN messages from all the group members converge at a single RP in a distributed fashion without global knowledge. Fig. 1a depicts the JOIN message from node A that is being routed to the RP.

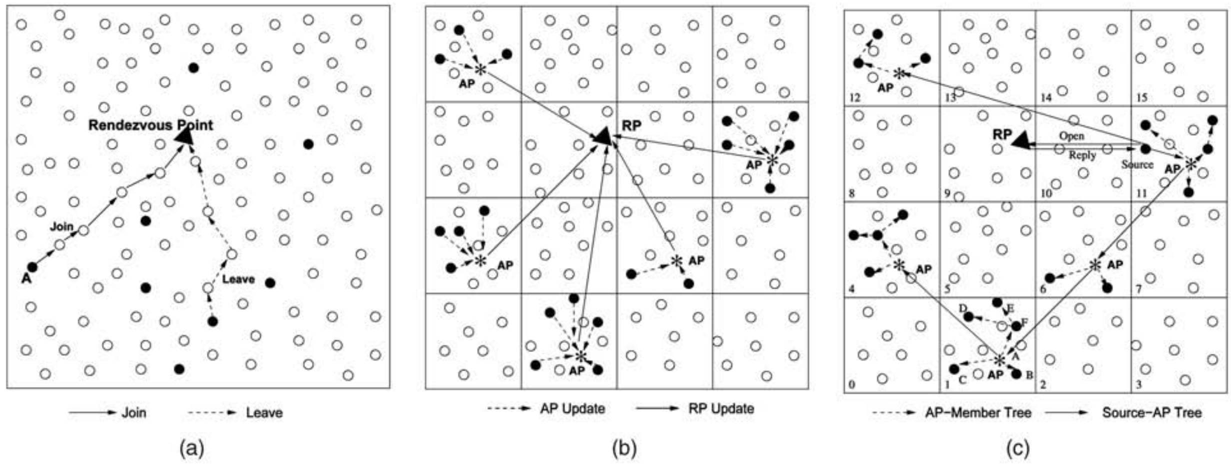

Fig. 1. Group management, tree construction, and data delivery in HRPM. (a) RPGM in HRPM $(d = 1)$. (b) Location updates in HRPM $(d = 4)$. (c) Data delivery in HRPM $(d = 4)$.

Note that computing the hashed location assumes that all nodes know the approximate geographic boundaries of the network. Such boundary information may be preconfigured at nodes before deployment or discovered by using some simple protocols. This assumption is consistent with the literature [20], [19]. Finally, we do not restrict the scope of the hash function (the ranges possible for the x and y output of the hash function) so as to have enough randomness in the location returned and consequent nice properties of load balancing: the RPs and APs for different groups will be spread out in a cell. Although it is true that hashing to a scoped area (say, a scaled-down version of the overall area) may minimize detours, it can also cause a hot spot at the center of the network. In general, random flows in an area cross the center, which cause the center to be already loaded. Thus, the hash function is allowed to return random x and y-coordinates in the area of the network without restriction.

3.1.2 Virtual Hierarchical Organization

To hierarchically apply RPGM described above, HRPM partitions the geographic domain into d^2 equal-sized square subdomains called *cells*, where d is the decomposition index. The partition can be recursively repeated until each cell consists of a manageable-sized subgroup of members. For ease of explanation, we restrict our following discussion of HRPM to two levels, as shown in Figs. 1b and 1c. We defer the description of how HRPM dynamically adjusts the d value according to the group size and why a two-level hierarchy is sufficient till Section 4.

In case of a two-level hierarchy, the members of each subgroup, that is, in each leaf cell of the hierarchy, choose an AP in the cell by using the same geographic hashing of the GID, except that the hashed location is scaled to be inside the cell. The APs then coordinate with the RP for the group. We extend the hash function for locating APs and the RP for a particular multicast group as follows:

$$H(GID, d, myLoc) = (x, y) \ where \ x, y \in Cellregion.$$

Here, d is the decomposition index, and $myLoc$ is the current location of the node invoking the function. Fig. 1b depicts the network partitioning for $d = 4$, in which case the region is divided into 16 cells. For the special case of $d = 1$ (Fig. 1a), only one cell exists in the region, and the function outputs the hashed location of the RP. Note that the procedure for electing the AP is essentially the same as that of electing the RP, except that it operates in a subspace of the original area. Finally, the APs for different groups in a cell may be different (since the $GIDs$ may hash to different locations); that is, there is not just one AP per cell.

3.1.3 Hierarchical Rendezvous Point Membership Management

To join a hierarchically decomposed multicast group, a node first generates the hashed location for the RP and sends a JOIN message to the RP, which is the same as in the flat-domain scenario. After receiving the value of the current decomposition index d of the hierarchy from the RP, the joining node invokes the hash function with d and its current location to compute the hashed location of the AP of its cell. The node then starts periodically sending LOCATION UPDATE packets to its AP. Such location updates are soft state and serve as a subgroup membership update; that is, if an AP stops receiving location updates from a member, it assumes that the member has migrated to another cell.

Upon receiving (or not receiving) a location update from each member, the AP summarizes the membership inside its cell as nonempty (or empty) and further propagates to the RP whenever the membership switches between empty and nonempty. The cells in which no group members exists do not have any active APs, and consequently, no updates from these cells are sent to the RP, further reducing the update congestion at the RP, as shown in Fig. 1b.

The state that the RP needs to keep about the group is just a bit vector of d^2 bits, with each bit representing whether a member exists in a particular cell or not. Thus, the RP can easily encode a large number of APs. For example, 256 APs from 256 cells can be encoded in 32 bytes. Thus, for a large multicast group, a two-level HRPM reduces the state required at the RP to d^2 bits while requiring the (leaf) AP in each cell to only maintain the addresses and locations of $\frac{G}{d^2}$ nodes on the average, where G is the original size of the multicast group.

The frequency of the location update determines the accuracy of the knowledge at the RP/APs and, consequently, the accuracy of the multicast tree. We use threshold-based updates where each node initiates a LOCATION UPDATE whenever it moves 100 m from the location of the last update. This is similar to the strategies used in location services for MANETs (for example, [20]).

When a node moves into a new cell, it does not immediately send an update to the new AP. Its previous AP can continue routing data using geographic forwarding. When the node moves a certain distance (that is, 100 m) from the location of its last update, it will send a new update to the AP in the new cell.

Note that the group management architecture of HRPM also needs to deal with the situation when nodes of a group are close to each other; that is, there is locality in the group membership. In such a case, extra overhead is incurred in sending control messages to an RP that may be far from the cluster of group members. Fortunately, a hierarchy is also useful in this scenario, since a group with locality will primarily send updates to a small set of APs in the clustered cells where the group members are located. The RP is only sent one update from each AP that indicates the existence of members in its cell. Each source only needs to retrieve a bit vector from the RP once to perform data delivery, which will locally be done through the nearby APs. Thus, when the group membership has geographic locality, HRPM incurs minimal overhead in using an RP. We believe that this small overhead is justified, given the overall overhead reduction made possible by using a virtual hierarchy.

3.1.4 Hierarchy Maintenance

As nodes move, the RP or an AP for a particular group may change as some other nodes become the closest to the hashed location of the GID. Thus, a handoff protocol is required to maintain geographic hashing. The current RP/AP on the receipt of any BEACON packet (used in geographic forwarding) checks whether this neighbor is currently closer to the hashed location. If so, the current RP/AP performs a handoff procedure that transfers the state of the multicast group/subgroup to the neighbor. This neighbor now becomes the RP/AP. Note that this process is transparent to the multicast group members.

In rare instances, messages that are sent to the RP/AP from different nodes may not converge at a single node. This could be due to the loss of BEACON packets, which causes inconsistencies in the view that each node has of its neighborhood. This convergence problem is solved as follows: when the first group management message *for a new group* NG arrives at a node A, which discovers that it is the closest to the hashed location, the node initiates a *converge* operation for the packet by buffering the packet and initiating an expanding ring broadcast search for any other node that also thinks that it is the current RP/AP node for the group NG. This search is limited to two hops, since any other potential RP/AP node is expected to be close by due to the geographic hash. If another node that acts as an RP/AP node for the group NG is located, the current node A relays the buffered packet to this RP/AP node for further processing, along with its own fresh location, so that the current RP/AP node can perform a handoff procedure, if appropriate, to A. This way, the consistency of RP/APs are maintained on the rare occasions that convergence does not occur.

3.1.5 Adaptivity and Per-Group Architecture

Another important design choice of HRPM is *adaptive per-group hierarchies*; that is, each group operates with its own virtual hierarchy based on its group size. Note that each group automatically has logically and potentially physically separate nodes that serve as RPs and APs. The per-group hierarchy is motivated by the fact that, depending on the group size G, there exists a trade off between the level of hierarchical partitioning required and the path length traveled by the location updates. The larger the number of levels in the hierarchy, the more the detours that location updates and data packets need to take to reach the RP. For small groups, since the amount of aggregation required is low and there is no hot spot at the RP, the hierarchy imposes overhead without adequate gain in performance. For large groups, increasing the levels of the hierarchy results in lower congestion at the RP and reduced encoding overhead in data packets. In summary, HRPM uses per-group hierarchy construction to allow choosing suitable hierarchy depths for groups of different sizes.

As will be discussed in Section 4, HRPM uses the RP to coordinate the construction of dynamic per-group hierarchies according to the changing group size. However, as will be explained in Section 4, the hierarchy depth rarely needs to be increased to beyond two levels.

3.2 Tree Construction and Data Delivery

HRPM provides a framework for scalable group management in location-based multicast, in which any tree construction algorithm of choice can be utilized based on the application metrics. For the performance study in this paper, we assume the use of a specific overlay tree construction algorithm that minimizes the bandwidth cost. The source of the multicast uses geographic distances between the multicast group members as edge weights to build an overlay graph, and then, a minimum spanning tree of the overlay graph (that is, an overlay tree) is built by using MST algorithms (for example, Prim's [21] or Kruskal's [21]). In Section 4.4, we evaluate different tree construction algorithms and show that such an overlay MST makes the best trade-off between bandwidth efficiency, computational cost, and location management overhead.

To send a data packet, the source sends an OPEN SESSION message to the RP and receives the membership group vector from the RP. The membership vector is of size d^2 bits, with a "1" bit for each cell that contains any group members. This vector is cached by the source. The RP differentially updates (sending only the changes) the source whenever the RP receives a change in membership notification from an AP. Once the group vector is received, the source can build a virtual overlay tree (the $Src \rightarrow AP$ tree) by assuming each active AP as a vertex in a topology graph. The tree is *virtual*, since the source does not need to know the actual AP node in each cell: it just needs to hash the GID in the AP's cell to put in a virtual vertex in the topology graph.

Multicast data packets are first delivered down the $Src \rightarrow AP$ tree. In this phase, the encoding overhead is the bitmap of d^2 bits (denoting which APs have active group members) that is inserted into every data packet. When a data packet reaches an AP, the AP performs the following operations:

1. It forwards the data packet on the remainder of the $Src \rightarrow AP$ tree below it by reconstructing the remaining tree based on the bitmap that it receives.
2. It constructs an $AP \rightarrow Member$ overlay tree to distribute this packet to the members of the group (the data packet has a destination group ID) in its cell by using the member locations that it stores. The AP then encodes the list of group members and their locations under each branch of the overlay tree in each data packet sent along that branch. On the average, the number of group members in a cell is $\frac{G}{d^2}$, where G is the group size. The packet is then delivered to the nodes down the $AP \rightarrow Member$ tree, with each node recomputing a tree of the remaining destinations in the list. Note that the size of this multicast header reduces as the packets travel down the tree and the depth of the remaining multicast tree reduces.

Example. Fig. 1c shows an example of data delivery in HRPM for a multicast group that only has group members in cells 1, 4, 6, 11, and 12. A multicast source receives a group vector with bits 1, 4, 6, 11, and 12 set from the RP, since only those cells contain group members and, consequently, active APs. It then constructs a virtual topology graph containing all the active APs, and builds a $Src \rightarrow AP$ multicast tree containing the active APs. The multicast data packet is then sent out on each branch (two at the source): one toward the AP in cell 12 with bitmap [0000000000001000] and one toward the AP in cell 11 with bitmap [0100101000010000]. On receiving the packet, the AP in cell 12 flips the bit that corresponds to its cell and finds no further APs, so it only sends the packet to the members in its cell. In contrast, when the AP in cell 11 receives the packets, it flips bit 11 and finds three remaining APs in the bitmap. It then constructs a tree to these three APs, encodes a new bitmap (without its own bit), and sends the packet. It also sends the packet to the members in its own cell. To see how the $AP \rightarrow Member$ tree works, consider the AP in cell 1. It constructs a tree with group members (B–F) and splits the packet three ways in the first branch of the tree. Node F can see the remaining group members in its branch (E, D) when it receives the packet and thus reconstructs the tree to send the packet forward.

Multiple sources. Multiple sources for a single group work similarly: each source retrieves the bitmap from the RP and constructs its independent $Src \rightarrow AP$ tree. Each AP can cache and reuse the $AP \rightarrow Member$ tree (saves computation) for delivering packets for multiple sources to members in its cell.

Since the primary focus of this paper is on multicast routing and group management, we do not address reliability and security issues due to the lack of space. As with all multicast protocols, the malicious operation of nodes or the failure of nodes can cause service disruptions. Mechanisms for dealing with these problems are part of our future work.

3.3 Dealing with Sparse Topology

A fundamental problem that has been researched with regard to geographic forwarding is the occurrence of local maxima while greedily forwarding packets. In such a situation, a packet is received by a node whose transmission range does not cover the destination location yet does not know of any other neighbor that is closer to the destination location than itself. Local maxima are more likely to occur in sparse network deployments. Even when the overall deployment is not sparse, certain regions of the network may be sparse due to nonuniform node distribution. Local maxima are also referred to as a *hole* [15] in the literature.

To enable geographic routing when local maxima occur, *face routing* was proposed (first in [15] and then extended in [16] and [17]) to route along the face of a planarized topology surrounding the hole graph until greedy forwarding can be invoked again or the destination is reached. Recent work [22] has extended the algorithms for face routing to take realistic radio transmission characteristics such as asymmetric links and nonideal ranges into account. Such schemes can also be easily incorporated into the geographic forwarding component of HRPM.

Similar to previous geographic unicast routing protocols, HRPM also needs to deal with *holes* in the network topology. Our implementation of HRPM uses GPSR [16] as the underlying geographic forwarding protocol to recover from holes. Holes can occur in the following cases during the operation of HRPM:

1. **Routing to a node.** This scenario occurs when the $AP \rightarrow Member$ overlay tree is being traversed for data delivery to the individual group members. In this case, the problem is similar to that faced by unicast geographic routing protocols, and thus, the normal protocol operations of GPSR (that is, distributed planarization followed by face traversal) are used to route the packet to the destination node, thereby avoiding the hole. This is expected to work, unless the network is partitioned.
2. **Routing to a hashed location.** Routing to a hashed location in HRPM occurs during the routing of JOIN, LEAVE, and LOCATION UPDATE messages to the RP/AP and during data delivery to the APs by using the $Src \rightarrow AP$ tree. Holes that occur whenever a message is routed to a hashed location have to be dealt with differently from when a message is being routed to a specific node. In the latter case, face routing is triggered whenever a node does not have the destination node in its table and does not know a neighbor that is closer to the destination node. On the other hand, dealing with a hole while routing to a hashed location

is more complicated, since when a node encounters a hole, it needs to distinguish whether the hole is en route from the sender to the hashed location or the hashed location is inside the hole.

We modify HRPM to deal with local maxima when routing to a hashed location as follows: a node X that detects local maxima stores the sequence number of the packet and starts face routing (perimeter forwarding mode in GPSR). During perimeter forwarding, the packet may be switched back to the greedy mode (if a node discovers itself to be closer to the hashed location than the point of entry into the current face). If this happens, the packet will continue to be normally routed. If the packet traverses around the face and comes back to X, then X becomes the RP. All subsequent packets are routed in this manner and are expected to reach the current RP despite a sparse topology.

3.4 Other Communication Primitives

In this paper, we focus on a design of HRPM for enabling multicast operation. Apart from multicast, anycast and manycast [23] are also useful communication primitives for MANETs. The group and location management architecture of HRPM can easily be leveraged for manycast and anycast services. For example, HRPM can be extended to provide manycast service, which delivers data to any k of G group members by constructing a tree consisting of k group members at the source of the manycast. In the nonhierarchical case, this k-member tree can trivially be constructed. In the hierarchical case, k APs (with group members) are selected to forward the message to. These selected APs then deliver the message to at most one group member in each of their cells.

HRPM can also be easily extended to provide anycast service. In this case, each node needs to contact its AP in the lowest level of the hierarchy for that group. The AP checks if it can locate a group member in its *cell*. If one exists, it is notified; otherwise, the anycast request is forwarded up the hierarchy to a higher level AP. This is recursively done until an anycast recipient is found. Note that this architecture allows for the anycast request to travel to a nearby anycast group member and exhibits good locality properties.

4 Analysis

In this section, we analyze the depth of the HRPM hierarchy and the choice of the decomposition index d.

4.1 Choice of d and Hierarchy Depth

We first show how HRPM chooses the decomposition index d that satisfies certain per-packet encoding overhead constraints. We then show that a two-level HRPM hierarchy is sufficient to support a very large multicast group. To simplify the analysis, we assume a random uniform distribution of N nodes in the geographic domain, the existence of G group members, and cells having about the same number of group members. For simplicity, we assume that the MANET region is a square of side length l and that each cell is a square of side length k.

In a two-level HRPM hierarchy, the $Src \rightarrow AP$ tree that is rooted at the source maximally has d^2 members (due to d^2 cells), and the per-packet encoding overhead is $d^2/8/f$ bytes, where f is the average fan-out of the overlay tree at the root. Each $AP \rightarrow Member$ tree has, on the average, $\frac{G}{d^2}$ members, and thus, the per-packet encoding overhead is at most $C \cdot \frac{G}{d^2}/f$ bytes, assuming that the cost of encoding the node identifier and locations is C. Note that as the data packet descends either type of overlay tree, the tree encoding overhead decreases as the remaining subtree becomes much smaller. Since the nodes within each cell are assumed to be uniformly distributed, that is, similar to the APs, the overhead in the two kinds of trees are expected to decrease in a similar fashion, and thus, we can focus on comparing the very first packet(s) that depart the tree roots.

Since the design goal of HRPM is to limit the per-packet encoding overhead, for example, to be less than ω bytes (or a fixed percentage of the payload), the partitioning of the network region into cells is governed by two constraints. The first constraint requires that the worst case encoding overhead in the $AP \rightarrow Member$ tree $C \cdot \frac{G}{d^2}/f$ be less than ω bytes. Assuming the worse case fan-out from the tree root of 1, the constraint becomes

$$C \cdot \frac{G}{d^2} \leq \omega. \quad (1)$$

The second constraint dictates that the worst-case encoding overhead in the $Src \rightarrow AP$ tree $\frac{d^2}{8}/f$ is also less than ω bytes. Thus, the constraint becomes

$$\frac{d^2}{8} \leq \omega. \quad (2)$$

In HRPM, the RP selects a particular decomposition index d based on the group size G and the MANET region side length l subject to the above constraints. Since all group JOIN and LEAVE messages reach the RP, it knows the group size G. The RP evaluates (1) to choose a d value that is just large enough to satisfy the constraint. It then checks if this value of d satisfies (2). In this case, HRPM forms a two-level hierarchy with decomposition index d. As an example, consider a multicast group of size 125. Using (1) and $\omega = 96\ \mathrm{bytes}$ (20 percent of 512 bytes), we have $d = 3.95 \approx 4$. As this value of d satisfies (2), HRPM will divide the network into 16 grids, with the RP having a constant encoding overhead of 2 bytes.

When the multicast group grows to be large enough that no choice of d can satisfy both (1) and (2) for a particular ω, HRPM increases the level of the hierarchy to 3 or higher. Effectively, the depth of the hierarchy should be the smallest h that satisfies (2) and

$$C \cdot \frac{G}{d^{2(h-1)}} \leq \omega. \quad (3)$$

In a depth h hierarchy, the top level remains to be a $Src \rightarrow AP$ tree, followed by $(h-2)$ levels of $AP \rightarrow AP$ trees, and the bottom level consists of $AP \rightarrow Member$ trees.

Based on the above analysis, for a reasonably small ω, a two-level hierarchy can support multicast groups that are larger than any deployable MANET today. For example, assume that the per-packet overhead is restricted to be below 20 percent of the payload size of 512 bytes, that is, around 100 bytes. Since 12 bytes are needed to encode a node identifier and its x- and y-coordinates, $C = 12$. Using (2), the maximum d that can be supported by the RP is 27 for $\omega = 96$. Substituting this value of d in (1) results in $G \approx 5800$; that is, a two-level HRPM hierarchy can support up to 5,800 group members while limiting the per-packet encoding overhead to be under 20 percent. Note due to the fan-out at the tree roots and the shrinking tree size during tree descent, the average per-data packet overhead is expected to be much lower than ω.

In a MANET that is densely populated with multicast group members, G can be large, l can be very small, and the large d value chosen by (1) can result in a small cell size k ($k = \frac{l}{d}$). When k is smaller than twice the radio range 500 m, HRPM limits k to be not smaller than 500 m and hence limits the corresponding value of d. The reason is that below this point, the area that an AP has to deal with is too small. In such a small cell area, if the group members are still too many to locally satisfy the ω constraint, data can easily be delivered by using localized ODMRP, since all nodes can be reached with two-hop broadcast transmissions. This is further discussed in Section 4.3.

When the RP decides to adjust d due to changes of the group size, it multicasts a NOTIFY message that contains the new d value to all member nodes, that is, via the current hierarchy. Upon receiving such a message, each member node generates the hashed location for its new AP and starts sending updates to that AP. The new APs then send the aggregated membership to the RP.

4.2 Trade Off between Encoding Overhead and Delay

There exists a trade off between the number of partitioned cells and the detours in the tree and, consequently, the delay in data delivery. In general, the more the partitions, that is, the larger the d, the longer the average detours that data packets will take before reaching group members. HRPM chooses the minimum d that satisfies both (1) and (2) to improve the forwarding cost (FC) and delay while satisfying the ω constraint. We experimentally evaluate this trade-off in the next section.

4.3 Dealing with Nonuniform Node Distribution

The above analysis and technique for choosing d suffices for uniform node distribution. When certain areas of the network have a higher concentration of group members (hot spots), the encoding overhead in those regions, that is, corresponding cells, may be unacceptable. HRPM deals with such a nonuniform node distribution with an additional mechanism as follows: if an AP detects that its group membership is larger than the ω constraint, it uses localized ODMRP to deliver packets in the cell. We call this *localized ODMRP* because ODMRP control packets (and, thus, the mesh constructed) are restricted to nodes within the cell's area by disallowing nodes whose geographic location is outside the cell to operate on the packet. This way, packets can more efficiently be delivered in dense cells by using wireless multicast advantage without incurring encoding overhead.

Thus, cells with high density of members due to nonuniform distribution can be accommodated easily in HRPM. When the membership in a congested cell becomes manageable (due to members moving out) later on, the AP will detect the change from the updates and switch to the normal unicast-tree-based data delivery.

4.4 Choice of Tree Construction Technique

HRPM multicast involves the construction of a tree that is rooted at the source and contains at least all the multicast group members. A first-cut approach is to construct a tree by using global knowledge of the locations of all nodes V in a MANET (both group members and nonmembers), that is, a Steiner tree. The Steiner tree problem[1] has been shown to be NP-Complete, and many heuristics have been proposed, which provide an approximate solution in polynomial time. For example, the TM heuristic [24] provides an approximation in $O(N^2)$ time, where N is the number of nodes in the MANET. The work in [11] DSM proposes such an approach, in which, given global knowledge of locations and group membership, a source can construct an approximate Steiner tree by using heuristics to perform multicast. However, such an approach requires the flooding of location and group membership information of each node to all nodes in the network in order to allow the construction of the Steiner tree at any source. Thus, this approach potentially limits the scalability of multicast.

A second approach is to construct an overlay minimum spanning tree (that is, a tree that spans the group members without involving intermediate nodes).[2] This approach is advantageous because 1) it reduces the group management overhead by managing the membership and location of only the G group members and 2) the overlay tree can be built by using computationally simpler algorithm such as Prim's or Kruskal's MST algorithms. However, the overlay tree can potentially be less bandwidth efficient than a Steiner tree that was constructed using both group member and nonmember nodes.

To evaluate which tree construction algorithm provides the best trade-off between bandwidth efficiency, delay, and computational complexity, we performed simulation experiments that compare the performance of three tree construction algorithms: 1) an overlay minimum spanning multicast tree built by using an MST algorithm, 2) a Steiner tree built by using the TM heuristic, and 3) a low-delay multicast tree in which the shortest paths (with the lowest accumulated weight edges) are used to deliver data to each group member built by using Dijkstra's single-source shortest path algorithm. Each tree construction algorithm was evaluated over 1,000 randomly generated sample network topologies of different sizes. The edge weights between a pair of nodes were set to the geographic distance between the pair of nodes.

Fig. 2 depicts the average bandwidth consumed (measured in the number of physical links on which a transmission is required), the average distance, and the maximum distance to any node in the multicast tree (measured in the number of hops). The results show that as the network size is increased, the bandwidth efficiency of the overlay and Steiner trees are very close, and both are much better than the shortest path tree. We also found that for a given network size, as the group size increases, there is a slight gain in the Steiner tree, since there are more

1. Given an undirected graph $G(V, E)$, a subset of vertices $\mathrm{R} \subseteq \mathrm{V}$, and an integer K, is there a subtree of G that includes all vertices in R and contains at most K edges?

2. An overlay tree is, in fact, a heuristic for the Steiner tree problem [25].

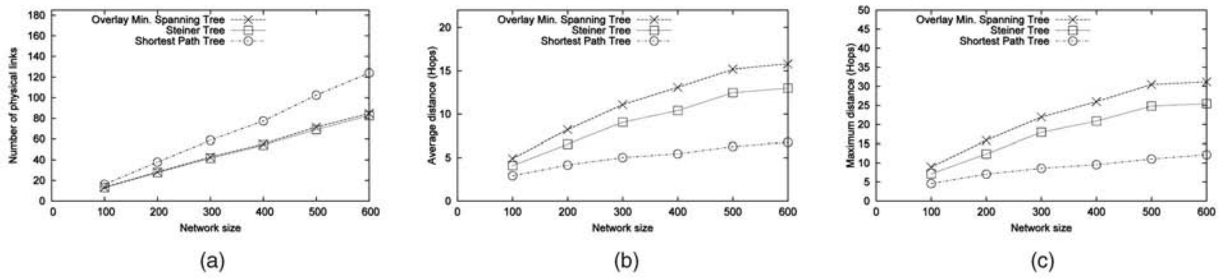

Fig. 2. Bandwidth consumption and delay of various tree building algorithms. Multicast group size for each network size is 10 percent of the network size. (a) Average bandwidth consumed. (b) Average distance (in hops). (c) Maximum distance (in hops).

opportunities of using nongroup members to improve the bandwidth efficiency. However, the gains observed are not significant enough to warrant the requirement that every node knows every other node's location, since this incurs high continuous overhead. Since Steiner tree construction using the TM heuristic is also more computationally expensive as the network size and the group size increase, the delays associated with the computations were also not acceptable.

In summary, due to the reduced location management overhead and acceptable bandwidth efficiency observed, HRPM uses an overlay MST to construct both the $Src \rightarrow AP$ tree and each of the $AP \rightarrow Member$ trees. The Src/AP constructs a graph, with edge weights being the geographic distances between group members. It then constructs an overlay MST by using Prim's algorithm, which has a complexity of $O(m^2)$, where m is the number of group members in the $Src \rightarrow AP$ tree, bounded by $max(d^2, \frac{G}{d^2})$. The $AP \rightarrow Member$ tree built at each active AP has, on the average, $\frac{G}{d^2}$ overlay edges. The $Src \rightarrow AP$ tree has, in the worst case, d^2 overlay edges, which happens when every cell has active group members.

Note that analyzing the goodness of various trees is not a contribution or specific focus of our paper. HRPM provides a framework that allows an easy management of state for multicast. In fact, many different tree construction techniques could be employed in HRPM. Different trees could also be used for the $Src \rightarrow AP$ and $AP \rightarrow Member$ trees. Evaluating the impact of trees specifically is part of our future work. For example, the work in [26] and [27] show that a Steiner tree problem does not provide minimum bandwidth consumption in wireless networks, since it does not account for wireless multicast advantages. The inclusion of trees built with heuristics for minimal data overhead trees in [26] and [27] in HRPM is a promising avenue for future work.

5 PERFORMANCE STUDY

In this section, we first describe the methodology of our study. We then present the performance results.

5.1 Simulation Methodology and Metrics

We implemented HRPM in the Glomosim [28] simulator. Glomosim has a comprehensive radio model and has been widely used for simulation studies of MANETs. We use an IEEE 802.11 radio with a bit rate of 2 megabits per second (Mbps) and a transmission range of 250 m. The mobility scenarios were generated by using the modified random waypoint mobility model [29]. For all simulations, the nodes move with a speed distributed uniformly at random between 1 and 20 m/s, and a pause time of 0 s is chosen. The simulation duration is 500 s, and the node density is 20 nodes/radio range. For multicast traffic, a source generates 512-byte packets at a constant rate of two packets/s. HRPM uses geographic forwarding with a beacon period of 4 s. Nodes send a LOCATION UPDATE after every movement of 100 m.

Since HRPM is the first location-based multicast proposed for large groups, we compare it to ODMRP [4], a nonlocation-based mesh multicast protocol that is well suited to operate in large groups and is widely used in multicast protocol studies. We used the Glomosim implementation of ODMRP, with the parameters set to the values specified in [30]. We also implemented flooding-based multicast (FLOOD) [31] in Glomosim for comparison. We also compare HRPM with a nonhierarchical version of HRPM (RPM) as a representative of the previously proposed location-based multicast protocols that are not hierarchical. In all the sections, unless otherwise specified, HRPM adjusts the decomposition index d to the group size based on the equations in Section 4 by using $\omega = 20\ \text{percent} \cdot PacketSize$.

The multicast protocols are evaluated using the following metrics:

1. *Multicast delivery ratio* (*MDR*): fraction of multicast data packets originated by the source that are received by the receivers.
2. *FC*: average number of data packet transmissions per delivered data packet to a receiver.
3. *Control overhead*: number of control packets transmitted by the multicast protocol (for HRPM this includes beaconing packets for geographic routing and multicast control packets such as JOIN, LOCATION UPDATE, and HANDOFF, and for ODMRP, this includes JOIN REQUEST and JOIN TABLE packets).
4. *Byte overhead*: total bytes of control data transmitted by the multicast protocol (these are the bytes of the control packets in metric 3 plus the encoding overhead in the data packets in the case of HRPM).
5. *Normalized Encoding Overhead* (*NEO*): ratio of the total number of encoding bytes[3] transmitted *at every hop* (including in the data packets finally not

3. The bytes used for the encoding of destinations and/or locations.

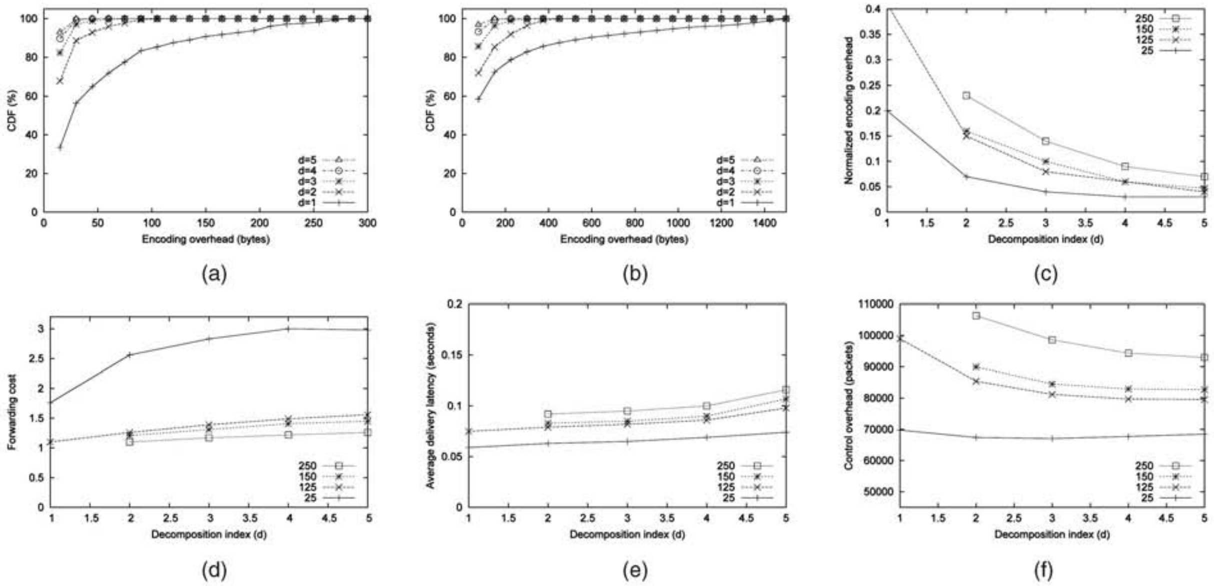

Fig. 3. Impact of the decomposition index d in HRPM. (a) Group size is 25. (b) Group size is 125. (c) NEO. (d) FC. (e) Average delivery latency (in seconds). (f) Control overhead (packets).

received) to the total number of data bytes received at the final destinations.

6. *Average Delivery Latency* (*Delay*): packet delivery latency averaged over all of the multicast packets delivered to all receivers.

5.2 Impact of Decomposition Index d

In this section, we study the impact of the decomposition index d. We use a network of 500 nodes in a terrain of $2300\ \mathrm{m} \times 2300\ \mathrm{m}$ with one multicast group and one source. The HRPM hierarchy has two levels, as there is no need for more levels, as discussed in Section 4. To evaluate the impact of different values of d on a given group size, HRPM's dynamic adjustment of d and limit on the minimum cell size is disabled. Instead, the decomposition index d is progressively assigned with values of 1, 2, 3, 4, and 5, which divide the network into 1, 4, 9, 16, and 25 cells, respectively. We first evaluate the savings in the encoding overhead for a small group (25 members) and a large group (125 members). We then evaluate the impact of d on multicast performance, with the group size ranging from 25 to 250 members.

Figs. 3a and 3b depict the CDF of the encoding overhead for all data packets transmitted. HRPM with $d = 1$ (no hierarchy) is equivalent to a small-group location-based multicast protocol (RPM) and suffers the long-tailed distribution of the encoding overhead. Additionally, these large packets are near the source, making the source a hot spot of congestion.

As d increases, as shown in Figs. 3a and 3b, the number of packets that have large encoding overhead decreases sharply. This occurs due to the reduction in encoding when a hierarchy is introduced. Since the largest value of d used is 5, the $Src \rightarrow AP$ tree has low encoding overhead, that is, less than 4 bytes. The maximum encoding overhead inside each AP cell is d^2 times smaller than that at the RP in the nonhierarchical case. This explains the short-tailed distribution of encoding overhead for larger values of d.

Fig. 3c depicts the NEO as d is increased. Note that HRPM with $d = 1$ (RPM) cannot support more than 125 members, and hence, we do not show any data points for larger groups. This occurs because the packet size grows beyond the 802.11 MAC-layer threshold in Glomosim (2,346 bytes), beyond which MAC fragmentation is required. This fragmentation feature is not supported in Glomosim. As predicted in the analysis, for large groups, the NEO is reduced significantly as d is increased. For a group size of 125 members, the NEO is reduced from 41 percent to 4 percent as d is varied from 1 to 5, a saving of 37 percent. More significantly, these savings are achieved at the cost of minimal increase in FC (Fig. 3d) and no reduction in MDR. The MDR is not depicted, since it remains close to 100 percent, with varying d's for all group sizes.

Figs. 3d and 3e show that as d is increased, both the FC and the delay of HRPM increase very slowly compared to the nonhierarchical version for large groups (with 125 members or more). This is because for large groups, the detours to the APs are not as costly, since a packet needs to travel many hops within the cell to reach multiple nodes anyway. For the small group (25 members), the increase in FC is more significant as d is increased. This is because the group members are sparser, but HRPM always first sends packets to APs, which then forward the packets to the few group members in their corresponding cells. A similar effect is expected, even in large groups, when d is increased to the extent that the group members in each cell are sparse. Thus, the choice of d trades off NEO with FC and delay.

Fig. 3f shows that the control overhead decreases as d is increased. This overhead is dominated by a constant number of beacon packets required for geographic forwarding. The remaining overhead are from HRPM's control packets, including JOIN, LOCATION UPDATE, and HANDOFF, with LOCATION UPDATE packets dominating the others. In nonhierarchical multicast, the updates travel to

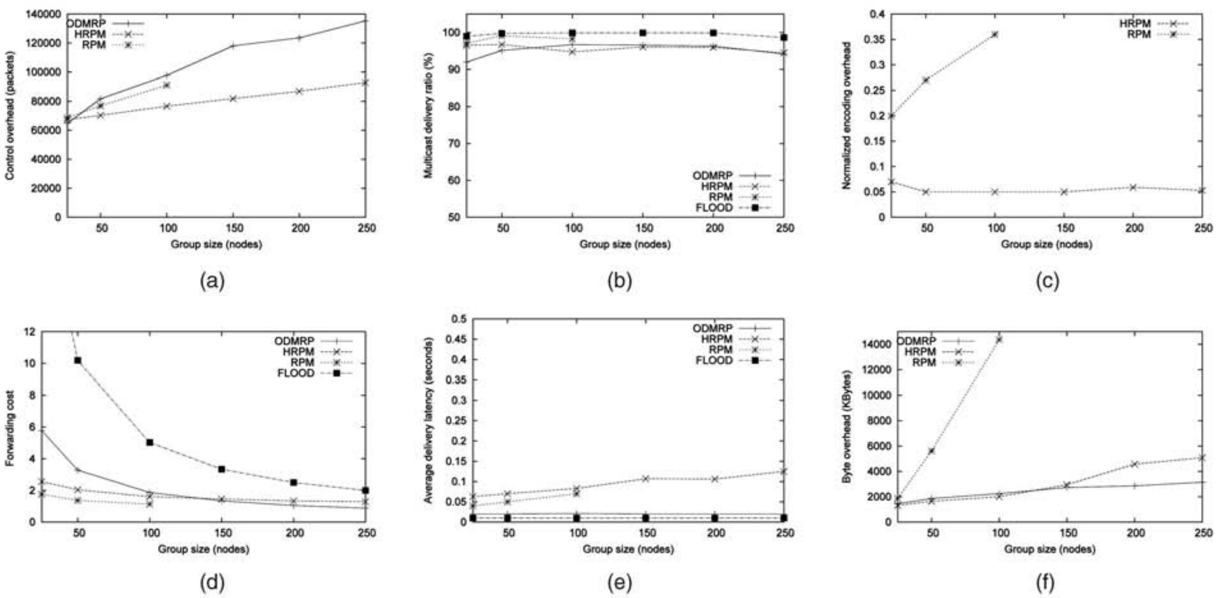

Fig. 4. Impact of the group size. RPM cannot support more than 125 group members due to its large incurred encoding overhead. (a) Control overhead (packets). (b) MDR (in percent). (c) NEO. (d) FC. (e) Average delivery latency (in seconds). (f) Byte overhead (in kilobytes).

the RP, whereas in a hierarchy, the more frequent member-to-AP updates travel shorter distance to the nearest APs. Further, the aggregated updates from AP to RP, which travel longer distance, are less frequent (as discussed in Section 3.1.3). Thus, the overhead of HRPM is lower than that of RPM. Also, as d increases, the member-to-AP updates travel shorter distances to the APs, and the control overhead reduces further. Although HANDOFF packets happen at all APs in a hierarchy, in addition to at the single RP, this overhead is overshadowed by updates.

5.3 Impact of Group Size

In this section, we study the impact of the group size on the protocol performance. We vary the group size from 50 to 250 members in a 500-node network with one multicast group and one source. The source sends data packets at 2 packets/s.

Fig. 4a shows that the control overhead of HRPM is lower than those of ODMRP and RPM across all group sizes, with the gap widening as the group size increases. FLOOD does not have any control overhead, as the protocol directly floods the data packets. ODMRP requires the source to periodically flood JOIN REQUEST messages. Each member node sends a JOIN TABLE packet in response to form a forwarding group (mesh) for the delivery of data packets. Thus, as the number of group members increases, the JOIN TABLES increase, thereby increasing the overhead of ODMRP. Similarly, the overhead of HRPM increases with group members due to the increase in the updates. However, HRPM builds a virtual hierarchy and performs group management without incurring any flooding cost due to the use of geographic hashing and thus has a lower overhead than ODMRP. In fact, the overhead of HRPM/RPM is dominated by the *beaconing* required for geographic forwarding. Beaconing incurs a constant overhead of 62,000 packets for all the group sizes depicted in the graph, and the actual protocol overhead of HRPM/RPM is a smaller fraction of the total overhead (8 percent at 25 nodes and 33 percent at 250 nodes). As explained earlier, the aggregation of LOCATION UPDATE at the APs in HRPM reduces its overhead compared to RPM.

Fig. 4b shows that FLOOD achieves the highest MDR of all the protocols. HRPM, ODMRP, and RPM also achieve a close to 100 percent MDR for all the group sizes. Note that the MDR for ODMRP for small group sizes is slightly lower due to a sparse forwarding mesh.

The encoding overhead (Fig. 4c) of HRPM remains steady as the group size increases, since it adjusts the d value to the varying group sizes. Note that although ω (encoding overhead constraint) is chosen to be 20 percent of the packet size, the average encoding overhead of HRPM is always below 7 percent (with an average of 5.5 percent). The encoding overhead of RPM significantly increases as the group size increases. Note that ODMRP and FLOOD do not encode destinations/locations in the data packet.

The next two performance metrics, FC (Fig. 4d) and delay (Fig. 4e), are affected by the tree construction algorithm used. Since HRPM constructs bandwidth-minimizing trees, it has the desirable property of a much lower FC than FLOOD and ODMRP for small groups, in which a large number of nonmember nodes are part of the mesh, resulting in higher FC. However, as the group size increases, much more member nodes become part of the forwarding mesh, which lowers the FC of FLOOD and ODMRP. Note that as the group size becomes large and group members become dense, ODMRP achieves an FC that is lower than 1 due to the multicast advantage. In spite of using a virtual hierarchy, the FC of HRPM is very close to that of RPM as the group size increases, which shows that the HRPM hierarchy does not cause significant detours in routing data packets.

The delay of FLOOD is the lowest, as expected (shown in Fig. 4e). Furthermore, due to the detour in the hierarchy approach, HRPM has a slightly higher delay than RPM. In contrast, ODMRP has a lower delay than both RPM and

TABLE 1
Impact of Increasing the Number of Sources to 5

Group Size	25		150	
	HRPM	ODMRP	HRPM	ODMRP
Control overhead (packets)	69,103	340,477	87,728	580,970
Byte overhead (KB)	3,052	8,671	14,443	15,219
MDR	98.66	98.46	97.78	97.80
FC	2.47	11.7	1.45	2.7
Delay	0.07	0.02	0.09	0.03

HRPM for three reasons. First, ODMRP can deliver multiple packets with a single transmission (wireless multicast advantage). Second, ODMRP uses unreliable broadcast, thus avoiding the cost of RTS/CTS channel access. However, this mechanism becomes increasingly unreliable with the increased network size, number of groups, number of sources, etc. RPM and HRPM use reliable unicast delivery, which incurs higher delay. Third, RPM and HRPM use bandwidth-minimizing overlay minimum spanning trees for data delivery. However, since any tree construction algorithms can be used under the HRPM/RPM framework without affecting the overhead, delay-minimizing trees such as LGK trees [12] could potentially be used to reduce the delay. Since FC and delay are affected by the tree construction algorithm and other factors mentioned above, and such metrics are less meaningful when one protocol achieves low delivery ratio, the rest of the performance comparison focuses on the control overhead and the delivery ratio.

Finally, Fig. 4f shows that compared to ODMRP, HRPM has similar byte overhead for smaller group sizes and but higher byte overhead for larger group sizes. This is simply because the number of data transmissions and the average encoding overhead for each data transmission in HRPM increase with the group size, whereas ODMRP has no encoding overhead bytes. However, HRPM scales significantly better than RPM, since the byte overhead of RPM rises quickly with the group size due to the increased encoding overhead.

In summary, HRPM significantly outperforms nonhierarchical location-based multicast with increasing group size and can also support significantly large group sizes competitively with ODMRP, which does not have disadvantages with the increasing group size.

Multiple sources. We also performed simulations where we increased the number of sources in the group to 5 for a small group (25 members) and a large group (150 members). The results are summarized in Table 1. ODMRP requires each source to periodically refresh the forwarding state in the network to deal with mobility and build the data delivery mesh. Thus, its overhead significantly grows with the number of sources. HRPM allows each source to build a virtual tree with almost no extra cost: it just needs to hash the active APs based on the group vector retrieved from the RP. Thus, the overhead of HRPM grows very slowly as the number of sources increases. Compared to Fig. 4a, the overhead of ODMRP increases by 425 percent for the group of size 25 and by 392 percent for the group of size 150 when the number of sources is increased from 1 to 5, whereas the overhead for HRPM only increased by 2.5 percent and 6.3 percent, respectively. Even the byte overhead of HRPM is lower than ODMRP. This is because the low-cost multicast group management in HRPM does not require each source to maintain its own multicast forwarding group. Note that HRPM achieves significant overhead reductions while delivering comparable numbers of packets as ODMRP for both group sizes. Finally, the delay of HRPM is slightly higher than ODMRP, since it uses unicasts instead of broadcasts and does not use the wireless multicast advantage. However, we argue that this is acceptable, given the large amount of overhead reduction. As the parameters scale, ODMRP delay is likely to increase due to an increase in contention from its large overhead.

In summary, HRPM scales well with the group size as compared to nonhierarchical protocols (for example, RPM), which cannot function beyond a certain group size (for example, 125 members). Due to the adaptive hierarchy construction, it maintains the encoding overhead below 7 percent as the group size scales. It delivers comparable percentages of packets as ODMRP across a range of group sizes while incurring lower overhead. For a fixed group size, as the number of sources increases, HRPM's overhead only increases slightly, whereas ODMRP suffers a large increase in overhead.

5.4 Impact of Number of Groups

In this section, we study the impact of the number of groups on protocol performance. We consider a 500-node network in an area of $2,300\ \mathrm{m} \times 2,300\ \mathrm{m}$. As the number of groups is increased, the group size is adjusted to keep the total number of receivers constant. We consider several scenarios by varying $L \times G$, where L is the number of groups, and G is the group size. The configurations are 2×90, 5×36, 10×18, 15×12, 20×9, 30×6, 36×5, and 45×4, with each scenario having 180 receivers, and each group having one source.

Fig. 5a shows that the overhead of HRPM grows very slowly as the number of groups is increased. This is because the update overhead of HRPM does not increase with the number of groups in the network. In contrast, ODMRP's overhead increases significantly as the numbers of groups increases. Increasing the number of groups results in the sources of different groups competing to broadcast JOIN REQUEST messages. This causes congestion and results in a drop in MDR to below 60 percent for ODMRP (Fig. 5b). In contrast, HRPM consistently delivers 95 percent or more data packets. Finally, Fig. 5c shows that the byte overhead of HRPM is also significantly lower than ODMRP. As the number of groups increases, the number of receivers per group is lower, and hence, the average encoding overhead decreases.

5.5 Impact of Network Size

In this section, we evaluate how the network size affects the multicast protocol performance. We vary the network size from 100 to 1,000 nodes while keeping the density constant at 20 nodes/radio range as before. For each network size, we consider group sizes of 5 percent and 30 percent of nodes in the network, respectively. To isolate the effect of the network size from the effect of multiple sources or multiple groups, for each network size, we consider only one group with one source.

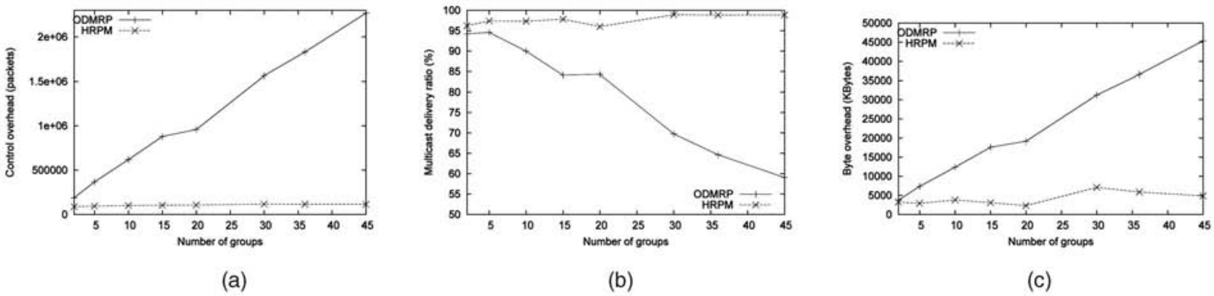

Fig. 5. Impact of the number of groups. (a) Control overhead (packets). (b) MDR (in percent). (c) Byte overhead (in kilobytes).

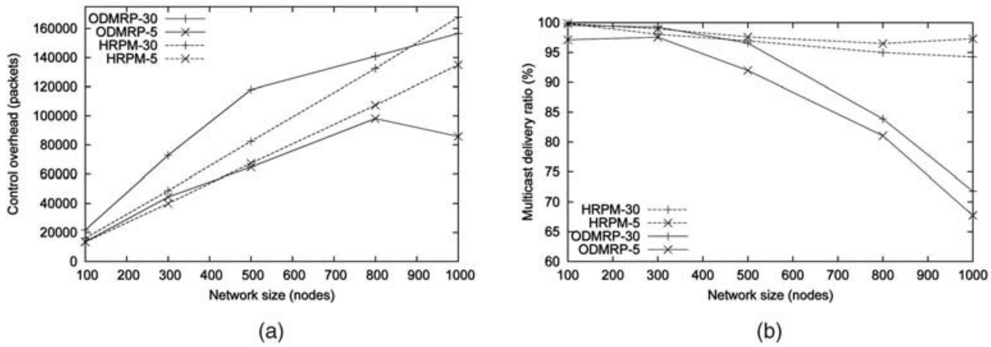

Fig. 6. Impact of the network size. (a) Control overhead (packets). (b) MDR (in percent).

Fig. 6b shows that the MDR of ODMRP drops significantly as the network size increases for both small and large groups. As the network size increases, the flooding-based mesh construction of ODMRP becomes increasingly costlier and unreliable. Additionally, the group members are spread farther apart, leading to higher probabilities of failures. In contrast, HRPM delivers more than 95 percent of the packets for the smaller groups and close to 95 percent of the packets for the larger groups across a wide range of network sizes. Additionally, as shown in Fig. 6a, HRPM incurs comparable overhead as ODMRP for small groups and lower overhead than ODMRP for large groups, except for a 1,000-node network. At such a large network size, the JOIN REQUEST for ODMRP does not reach all the members, thereby reducing the number of JOIN TABLES sent. This causes a reduction in the overhead and MDR. In summary, HRPM is more scalable than ODMRP for both small and large groups as the network size increases. Note that ODMRP is expected to further degrade in comparison to HRPM if the number of sources or groups are increased in large networks, as suggested by results in the previous sections.

5.6 Impact of Nonuniform Node Distribution

In all previous scenarios, nodes were randomly uniformly distributed in the entire area. We now stress HRPM by introducing nonuniformity in node distribution by taking 150 group members in a 500-node network in an area of $2300\ \mathrm{m} \times 2300\ \mathrm{m}$ and confining the movement of these 100 out of the 150 group members to the bottom left part of the area of size $1000\ \mathrm{m} \times 1000\ \mathrm{m}$. This introduces a large density of group members in the cells in that area and causes the HRPM APs in these affected congested cells to switch to localized ODMRP-based data delivery, since the number of group members remains too large to satisfy the w constraint.

The evaluation showed that HRPM deals well with nonuniform member distributions if they arise. Specifically, compared to the results in Fig. 4 for 150 group members distributed uniformly, the same scenario with nonuniform member distribution resulted in HRPM delivering a comparable fraction of packets (95.3 percent) with comparable control overhead. The overall byte overhead was reduced by 7 percent due to no encoding overhead in the congested cells, whereas the FC also reduced from 1.46 to 1.28 due to packets being delivered in congested cells with wireless multicast advantage. Finally, the delay was also reduced by 11 percent due to the use of flooding to deliver packets in congested cells.

We note that the fact that HRPM handles nonuniform member distribution by using the localized ODMRP in hotspot cells should not be generalized as ODMRP outperforms HRPM, as the target scenario of HRPM, like all other stateless protocols, is multicast groups with sparsely distributed members. In such scenarios, high concentration of members happens as a rare event; that is, they only appear in a few cells. When this happens, HRPM resorts to the flooding-style protocol to be more efficient only in the troublesome cells.

6 Related Work

HRPM is closely related to previous location-based multicast protocols. In addition, it is related to stateless nonlocation-based multicast protocols, hierarchical nonlocation-based multicast protocols, and other uses of geographic hashing.

6.1 Location-Based Multicast Protocols

Previous location-based protocols [11], [12], [13] were proposed for small groups due to the constraint of encoding either the entire tree or the destinations in the data packet headers. In DSM [11], each node floods its location in the network. DSM constructs a physical Steiner tree by using the TM heuristic [24] at the source, optimally encodes the physical multicast tree into each packet, and delivers the packet by using source routing. LGT [12] requires each group member to know every other group member's location. LGT proposes two overlay multicast trees: a bandwidth-minimizing LGS tree and a delay-minimizing LGK tree. PBM [13] does not explicitly construct trees but rather relies on a multicast geographic forwarding strategy similar to the hop-by-hop forwarding proposed in SGM [32] and DDM [8].

Other new proposals for geographic multicast routing have also been proposed either in a general setting or for sensor networking (see [27], [33], [34], and [35]). Although these works target different ways of constructing multicast trees in conjunction with geographic routing, HRPM's focus is to provide a framework for scalable and efficient group management. In fact, a promising future research direction is to combine localized tree construction as in [34] with the scalable group management of HRPM.

The SPBM protocol proposed in [36] is closely related to HRPM, as the two share the essence of improving the scalability of location-based multicast by using hierarchical group management. However, HRPM is different from SPBM in several fundamental aspects: 1) SPBM uses *flooding* in hierarchical group management. In contrast, HRPM uses *mobile geographic hashing* (convergence to the RP) in hierarchical group management, which does not incur any flooding cost. 2) SPBM defines a static hierarchy by dividing the network into a quad tree with a predetermined maximum aggregation level L. In contrast, HRPM uses a per-group hierarchy that dynamically adjusts to the group membership. 3) The need to propagate the group member's *location information* up the hierarchy while exploiting the distance effect (so that flooding to remote squares in the grid is less frequent) makes SPBM more susceptible to node mobility due to the delayed propagation of information when a node's position in the hierarchy changes. For example, when a group member enters a large square that did not previously contain a group member, it will not receive any packets until the membership in the square has been spread to reach the source. In contrast, in HRPM, the locations of APs are "virtual" and, hence, fixed, and the low-overhead rendezvous-based group management allows an AP to update the RP as soon as its *membership* changes. Within each leaf cell, the AP keeps track of each group member's last known location, which allows packets to be forwarded to group members despite that they are moving into a new cell, since geographic forwarding is resilient to slight variations in destination locations. Experimentally, the evaluation of SPBM [36] shows that SPBM degrades in PDR and becomes similar to ODMRP as the maximum node speed increases to 15 m/s, with a pause time of 10 s. In contrast, all the simulations in our study were carried out at a higher maximum node speed of 20 m/s, with a pause time of 0 s, and HRPM consistently outperforms ODMRP.

Note that our evaluation of HRPM uses a similar node density as in the SPBM evaluation [36] although with a packet size that is eight times larger and twice the traffic volume. Moreover, HRPM has been evaluated by scaling a much wider range of network parameters such as the group size, number of groups, number of senders, and network size. The magnitude of the parameters studied are also different. Although SPBM has been shown to work well for 25 group members, HRPM has been evaluated for up to 250 group members. Similarly, SPBM has been evaluated for 196 nodes, whereas HRPM has been evaluated for up to 1,000 nodes.

6.2 Stateless Multicast Protocols

Stateless multicast protocols have been proposed to reduce state at forwarding nodes by encoding multicast destinations in headers and are typically used for small groups. The work in SGM [32] proposed this technique for the Internet. REUNITE [37] requires only branch-point routers to keep state for IP multicast. DDM [8] uses similar principles to provide stateless multicast in MANETs.

6.3 Hierarchical Multicast Protocols

Several hierarchical nonlocation-based protocols have been proposed, which can be overlay or nonoverlay based. Protocols such as AMRIS [38] and PAST-DM [6] propose an overlay-based approach in which the overlays are a form of hierarchies.

An example of a nonoverlay hierarchical MANET multicast protocol is HDDM [9], which extends DDM to include a hierarchical structure. Similar to HDDM, HRPM also leverages the well-known technique of introducing a hierarchical structure to reduce the overhead. Despite this similarity, HDDM is a topology-aware approach, whereas HRPM is a location-aware approach. Thus, the design challenges and issues in both protocols are very different. HRPM needs to provide location management and routes by using locations rather than topology. The focus of our paper was to improve the scalability of location-based multicast, and so, a comparison with HDDM is out of the scope of this work. Note that our evaluation of HRPM scales parameters (network size, group size, and number of groups) to larger values than previous work on scalable multicast.

The work in [39] proposed the use of *cores* to reduce control traffic for creating multicast delivery structures. They propose that group members form a multicast group by sending JOIN REQUESTS to a set of cores. RPs are similar in concept to core nodes. However, RPs/APs in HRPM can be located without any overhead by using geographic hashing and can be more resilient to mobility due to not being tied to a particular node whose movement needs to be tracked.

6.4 Location Management for Unicast

Several location management protocols have been proposed for *unicast* services. The work in [40] provides a comparison of some well-known unicast location services. However, location management for unicast is fundamentally different from that for multicast such as in HRPM, since it does not need to provide locations of an entire group to the source node.

6.5 Other Uses of Geographic Hashing

HRPM shares the concept of geographic hashing with GHT [19] for data-centric storage systems and consistent hashing in distributed indexing such as DIM [41] and DIFS [42] for supporting range queries in sensor networks. These protocols were proposed for use in static sensor networks and do not have to deal with mobility. Geographic hashing has also been used in location services (for example, [43] and [40]) in which each node's identifier is hashed to a home region that consists of one or more nodes, which serve as that node's location servers.

6.6 Geographic Routing in Wireless Networks

HRPM uses geographic forwarding to forward data and control packets. The first proposal for geographic forwarding was laid out in [44]. Subsequently, detailed algorithms have been designed for the application of geographic forwarding in wireless networks (see [45], [16], and [17]). Many optimizations and modifications have been proposed for these basic algorithms to deal with real network topology and provide robustness. Work has also been done on real testbeds to identify and remove pathologies that arise in geographic routing [22]. The work in [46] shows that guaranteed delivery can be done in arbitrary planar graphs. Geographic forwarding in HRPM is not restricted to one single proposal and can potentially take advantage of new and more efficient techniques developed for geographic routing.

Another body of work proposes the use of geographic routing without actual node location information [47], typically achieved by using localization algorithms to assign nodes' virtual coordinates and then routing geographically by using these virtual coordinates. Such schemes can be used in HRPM.

7 Conclusions

In this paper, we propose the HRPM protocol, which leverages two techniques: distributed mobile geographic hashing and hierarchical decomposition of large multicast groups to improve the scalability of location-based multicast. Together, the two techniques enable lightweight hierarchical membership management, which reduces the per-packet encoding overhead without incurring the high cost associated with maintaining a distributed state at any particular mobile nodes.

Our simulation results show that HRPM significantly improves the scalability of location-based multicast in terms of the group size. Coupled with its leverage of stateless geographic forwarding, HRPM scales well in terms of the group size, the number of groups, and the size of the network. In particular, HRPM maintains a close to 95 percent MDR while incurring, on the average, a 5.5 percent per-packet tree-encoding overhead for up to 250 group members in a 500-node network. Furthermore, it achieves a steady 95 percent delivery ratio while incurring nearly constant overhead as the number of groups increases from 2 to 45, while keeping the total number of receivers constant at 180, in a 500-node network. Last, it steadily achieves above 90 percent of delivery ratio as the network scales up to 1,000 nodes with up to 30 percent of group members.

For our future work, we plan to study the impact of new tree construction algorithms, the use of broadcast that does not require RTS/CTS, and exploiting the wireless multicast advantage on different data delivery performance metrics such as delay, FC, and delivery ratio. We are also interested in evaluating manycast and anycast services using HRPM.

Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) Grant ANI-0338856.

References

[1] C. de Morais Cordeiro, H. Gossain, and D.P. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions," *IEEE Network,* vol. 17, no. 1, Jan.-Feb. 2003.

[2] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad Hoc On-Demand Distance Vector Routing Protocol," *Proc. ACM MobiCom '99,* Aug. 1999.

[3] J.G. Jetcheva and D.B. Johnson, "Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks," *Proc. ACM MobiHoc '01,* Oct. 2001.

[4] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '99),* Sept. 1999.

[5] J. Xie, R.R. Talpade, A. Mcauley, and M. Liu, "AMRoute: Ad Hoc Multicast Routing Protocol," *Mobile Networks and Applications,* vol. 7, no. 6, pp. 429-439, 2002.

[6] C. Gui and P. Mohapatra, "Efficient Overlay Multicast for Mobile Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '03),* Mar. 2003.

[7] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad Hoc Routing," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '99),* Sept. 1999.

[8] L. Ji and S. Corson, "Differential Destination Multicast—A MANET Multicast Routing Protocol for Small Groups," *Proc. IEEE INFOCOM '01,* Apr. 2001.

[9] C. Gui and P. Mohapatra, "Scalable Multicasting for Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[10] J. Luo, P.T. Eugster, and J.-P. Hubaux, "Route-Driven Gossip: Probabilistic Reliable Multicast in Ad Hoc Networks," *Proc. IEEE INFOCOM '03,* Mar. 2003.

[11] S. Basagni, I. Chlamtac, and V. Syrotiuk, "Location-Aware Dependable Multicast for Mobile Ad Hoc Networks," *Computer Networks,* vol. 36, pp. 659-670, Aug. 2001.

[12] K. Chen and K. Nahrstedt, "Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET," *Proc. IEEE INFOCOM '02,* June 2002.

[13] M. Mauve, H. Füßler, J. Widmer, and T. Lang, "Position-Based Multicast Routing for Mobile Ad Hoc Networks," Technical Report CS TR-03-004, Univ. of Mannheim, 2003.

[14] USCG Navigation Center GPS Page, http://www.navcen.uscg.nil/gps/default.html, Jan. 2000.

[15] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. Third ACM Int'l Workshop Discrete Algorithms and Methods for Mobile Computing (DIALM '99),* Aug. 1999.

[16] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM MobiCom 2000,* Aug. 2000.

[17] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad Hoc Routing," *Proc. ACM MobiHoc '03,* June 2003.

[18] L. Kleinrock and F. Kamoun, "Hierarchical Routing for Large Networks: Performance Evaluation and Optimization," *Computer Networks,* vol. 1, pp. 155-174, 1977.

[19] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage in Sensornets," *Proc. First ACM Workshop Wireless Sensor Networks and Applications (WSNA '02),* Sept. 2002.

[20] J. Li, J. Jannotti, D.S.J.D. Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. ACM MobiCom '00,* Aug. 2000.

[21] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms.* MIT Press, 1990.

[22] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," *Proc. Second Usenix Symp. Networked Systems Design and Implementation (NSDI '05),* May 2005.

[23] C. Carter, S. Yi, P. Ratanchandani, and R. Kravets, "Manycast: Exploring the Space between Anycast and Multicast in Ad Hoc Networks," *Proc. ACM MobiCom,* 2003.

[24] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Math. Japonica,* vol. 24, pp. 573-577, 1980.

[25] E.N. Gilbert and H.O. Pollak, "Steiner Minimal Trees," *SIAM J. Applied Math.,* vol. 16, pp. 1-20, 1968.

[26] P.M. Ruiz and A.F.G. Skarmeta, "Approximation Algorithms for Minimum Bandwidth Multicast Trees in Wireless Mesh Networks," *Lecture Notes in Computer Science,* 2006.

[27] P.M. Ruiz and I. Stojmenovic, "Cost-Efficient Multicast Routing in Ad Hoc and Sensor Networks," *Handbook on Approximation Algorithms and Metaheuristics,* T. Gonzalez, ed., Chapman and Hall/CRC, 2006.
[28] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A Library for Parallel Simulation of Large-Scale Wireless Networks," *Proc. 12th Workshop Parallel and Distributed Simulation (PADS '98),* May 1998.
[29] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," *Proc. IEEE INFOCOM '03,* Apr. 2003.
[30] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," *Proc. IEEE INFOCOM '00,* Mar. 2000.
[31] K. Obraczka, G. Tsudik, and K. Viswanath, "Pushing the Limits of Multicast in Ad Hoc Networks," *Proc. 21st Int'l Conf. Distributed Computing Systems (ICDCS '01),* Apr. 2001.
[32] R. Boivie, N. Feldman, and C. Metz, "Small Group Multicast: A New Solution for Multicasting on the Internet," *IEEE Internet Computing,* vol. 4, no. 3, pp. 75-79, 2000.
[33] I. Stojmenovic, "Localized Network Layer Protocols in Sensor Networks Based on Optimizing Cost over Progress Ratio," *IEEE Network,* vol. 20, no. 1, Jan.-Feb. 2006.
[34] J. Sanchez, P. Ruiz, X. Liu, and I. Stojmenovic, "GMR: Geographic Multicast Routing for Wireless Sensor Networks," *Proc. Third Ann. IEEE Conf. Sensor, Mesh, and Ad Hoc Comm. and Networks (SECON '06),* 2006.
[35] S. Wu and K.S. Candan, "GMP: Distributed Geographic Multicast Routing in Wireless Sensor Networks," *Proc. 26th IEEE Intl' Conf. Distributed Computing Systems (ICDCS '06),* 2006.
[36] M. Transier, H. Fuler, J. Widmer, M. Mauve, and W. Effelsberg, "Scalable Position-Based Multicast for Mobile Ad-hoc Networks," *Proc. First Int'l Workshop Broadband Wireless Multimedia: Algorithms, Architectures and Applications (BroadWim '04),* Oct. 2004.
[37] I. Stoica, T.S.E. Ng, and H. Zhang, "REUNITE: A Recursive Unicast Approach to Multicast," *Proc. IEEE INFOCOM '00,* Mar. 2000.
[38] C. Wu and Y. Tay, "AMRIS: A Multicast Protocol for Ad Hoc Wireless Networks," *Proc. Military Comm. Conf. (MILCOM '99),* Nov. 1999.
[39] E. Madruga and J. Garcia-Luna-Aceves, "Scalable Multicasting: The Core Assisted Mesh Protocol," *ACM/Baltzer Mobile Networks and Applications,* special issue on management of mobility in distributed systems, vol. 6, no. 1, 2001.
[40] S.M. Das, H. Pucha, and Y.C. Hu, "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing," *Proc. IEEE INFOCOM '05,* Mar. 2005.
[41] X. Li, Y.J. Kim, R. Govindan, and W. Hong et al., "Multi-Dimensional Range Queries in Sensor Networks," *Proc. First ACM Conf. Embedded Networked Sensor Systems (SenSys '03),* Nov. 2003.
[42] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker, "DIFS: A Distributed Index for Features in Sensor Networks," *Proc. First IEEE Int'l Workshop Sensor Network Protocols and Applications (SNPA '03),* May 2003.
[43] I. Stojmenovic, "Home Region Based Location Updates and Destination Search Schemes in Ad Hoc Wireless Networks," Technical Report TR-99-10, School of Information Technology and Eng., Univ. of Ottawa, Sept. 1999.
[44] G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," USC/ISI Technical Report ISI/RR-87-180, Mar. 1987.
[45] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Wireless Networks,* vol. 7, no. 6, 2001.
[46] H. Frey and I. Stojmenovic, "On Delivery Guarantees of Face and Combined Greedy-Face Routing in Ad Hoc and Sensor Networks," *Proc. ACM MobiCom,* 2006.
[47] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing without Location Information," *Proc. ACM MobiCom '03,* Sept. 2003.

Saumitra M. Das received the BEngg degree from the University of Bombay, India, and the MS degree from Carnegie Mellon University. He is currently working toward the PhD degree in the School of Electrical and Computer Engineering, Purdue University. His research interests include cross-layer system designs for multihop wireless networks, scalable routing strategies in wireless ad hoc networks, and mobile robotics. He is a member of the IEEE.

Himabindu Pucha received the BEngg degree from the University of Bombay, India, and the MSEE degree from Purdue University. She is currently working toward the PhD degree in the School of Electrical and Computer Engineering, Purdue University. Her research interests include Internet routing and overlay networks, peer-to-peer systems and applications, and mobile computing. He is a member of the IEEE.

Y. Charlie Hu received the MS and MPhil degrees from Yale University in 1992 and the PhD degree in computer science from Harvard University in 1997. From 1997 to 2001, he was a research scientist at Rice University. He is currently an associate professor of electrical and computer engineering and computer science at Purdue University. He is a cofounder and technical program committee (TPC) cochair of the International Workshop on Mobile Peer-to-Peer Computing (MP2P) and the TPC vice chair of the 33rd International Conference on Parallel Processing (ICPP 2004). His research interests include operating systems, distributed systems, networking, and parallel computing. He has published more than 100 papers in these areas. He is a senior member of the IEEE, Usenix, and ACM. He received a US National Science Foundation Faculty Early Career Development (CAREER) Award in 2003.