ADHOC 131
11 July 2005; Disk Used
ARTICLE IN PRESS
No. of Pages 21, DTD = 5.0.1

Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Ad Hoc Networks xxx (2005) xxx–xxx
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Hierarchical multicast techniques and scalability in mobile Ad Hoc networks

Chao Gui *, Prasant Mohapatra

Department of Computer Science, University of California at Davis, One Shields Avenue, Davis, CA 95616, USA

Received 4 August 2004; received in revised form 28 March 2005; accepted 14 June 2005

Abstract

Many potential applications of Mobile Ad Hoc Networks (MANETs) involve group communications among the nodes. Multicasting is an useful operation that facilitates group communications. Efficient and scalable multicast routing in MANETs is a difficult issue. In addition to the conventional multicast routing algorithms, recent protocols have adopted the following new approaches: overlays, backbone-based, and stateless. In this paper, we study these approaches from the protocol state management point of view, and compare their scalability behaviors.

To enhance performance and enable scalability, we have proposed a framework for hierarchical multicasting in MANET environments. Two classes of hierarchical multicasting approaches, termed as domain-based and overlay-driven, are proposed. We have considered a variety of approaches that are suitable for different scenarios such as multicast group sizes and number of groups. Results obtained through simulations demonstrate enhanced performance and scalability of the proposed techniques.

Keywords: Hierarchical multicasting; Mobile Ad hoc networks; Domain-based multicasting; Overlay multicasting; Stateless multicasting; Scalability

1. Introduction

The use of mobile and wireless devices are becoming ubiquitous. Thus the need for efficient intercommunication among these devices is becoming critical. In addition to the infrastructure-based cellular wireless network, the study and developments of infrastructureless wireless networks have been very popular in recent years. Mobile Ad hoc NETworks (MANETs) belong to the class of infrastructureless networks, which do not require the support of wired access points

---

* Corresponding author. Tel.: +1 530 754 8380; fax: +1 530 754 4767.
E-mail addresses: guic@cs.ucdavis.edu (C. Gui), prasant@cs.ucdavis.edu (P. Mohapatra).

1570-8705/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.adhoc.2005.06.003

for intercommunication. It is a dynamically reconfigurable wireless network where the nodes are mobile resulting in variable network topology. Due to the limited radio propagation range, nodes of a MANET communicate either through single hop or multihop transmissions. The nodes act as both hosts as well as routers. Applications of MANETs include battlefield communication, disaster recovery, coordinated task scheduling (such as earth moving or construction), vehicular communication for traffic management, data and information sharing in difficult terrain, and extension of the infrastructure-based wireless networks.

Most applications of MANETs listed earlier operate in a group-based collaborative manner. So they need support for group communication protocols. A recent survey of multicast routing protocols in MANETs was reported in [1], and the performance comparison of some of these protocols are discussed in [2]. Protocol state reduction techniques have been proposed through is represented by hierarchical multicast [3–5] and overlay multicast [6,7] in the Internet, and recent works in MANET multicasting [9–13,15]. Among these MANET multicast protocols, AMRoute (Ad hoc Multicast Routing Protocol) [10] and PAST-DM (Progressively Adapted Sub-Tree algorithm on Dynamic Mesh) [11], are overlay multicast protocols, which limit the protocol state maintenance within the group members. Backbone-based protocols, such as MCEDAR [9], and the protocols reported in [16,17], use another state constraining method. Only a selected subset of nodes which form the virtual backbone of the network get involved in routing. Thus protocol states are confined within the virtual backbone. The stateless multicasting protocols do not maintain any protocol state at the forwarding nodes. Examples of these protocols include DDM (Differential Destination Multicast) [12], LGT (Location Guided Tree construction algorithms) [13] and RDG (Route Driven Gossip) [15].

In this paper, we study the relationship of the protocol state management techniques and the performance of multicast operations. For performance, we focus on protocol control overhead and protocol robustness. We further address interested in the following two questions:

(1) Will the state constraining methods successfully reduce the protocol control overhead?
(2) When the multicast service scales up vertically (in terms of the group size) and horizontally (in terms of the number of groups), how will the scalability impact the protocol performance?

In order to better address these questions, we present two hierarchical multicast routing solutions for MANETs. The first solution, termed as domain-based hierarchical routing, divides a large multicast group into sub-groups, each with a node assigned as a sub-root. Only the sub-roots maintain the protocol states, and are selected on the basis of topological optimality. Thus, we can have a more flexible control on the protocol state distribution. The second solution, termed as overlay-driven hierarchical routing, has a different way of building multicast hierarchy. Overlay multicasting is used as the upper layer protocol, and stateless small group multicasts are used as lower layer multicast protocols. This hierarchical multicast solution achieves protocol robustness, as well as provides efficient data delivery. These features make overlay multicast approach more suitable for the MANET environment.

We study the protocol performance using simulations of large networks (400 mobile nodes). We simulate protocol scalability behaviors with group size of up to 200 members and number of groups up to 12. The results show robust and scalable performance for both hierarchical multicast schemes proposed in this paper.

The rest of the paper is organised as follows. In Section 2, we study the state management methods of the current MANET multicast protocols, and their scalability issues. In Section 3, we briefly study the traditional multicast methods in the Internet, using hierarchical methods, and discuss how hierarchical multicasting is different in MANETs. In Section 4, we present two hierarchical multicast schemes for MANETs. Results of performance studies are presented in Section 5.

In Section 6, we discuss the related works, followed by the conclusions in Section 7.

2. Multicasting in MANETs: State management and scalability

State management of multicast protocols involves timely updatings of the multicast routing tables at the involved nodes to maintain the correctness of the multicast routing structure, tree or mesh, according to the current network topology. Even under moderate node mobility and multicast member size, state management incurs considerable amount of control traffic. When the group size grows, and/or number of groups increase, traditional tree or mesh based methods [18–21] become inefficient. To address the scalability issues, we need to reduce the protocol states and constrain their distribution, or even use methods that do not need to have protocol state. A number of research efforts have adopted this method, which can be classified into the following categories: overlay multicasting, backbone-based multicasting and stateless multicasting. We study these different approaches for constraining protocol states, and their scalability issues.

2.1. Overlay multicast protocols

In overlay multicast, a virtual infrastructure is built to form an overlay network on top of the physical network. Each link in the virtual infrastructure is a unicast tunnel in the physical network. IP layer implements minimal functionality—a best-effort unicast datagram service, while the overlay network implements multicast functionalities such as dynamic membership maintenance, packet duplication and multicast routing. Overlay multicast was proposed to deploy multicast functionality to an all unicast IP network such as the Internet [6,7]. Different overlay mulitcast methods are surveyed and compared in [8] AMRoute [10] is an ad hoc multicast protocol that uses the overlay multicast approach. The virtual topology can remain static even though the underlying physical topology is changing. Moreover, it needs no support from the non-member nodes, i.e., all multicast functionality and protocol states are kept within the group member nodes. The protocol does not need to track the network mobility since it is totally handled by the underlying unicast protocol.

The advantages of overlay multicasting come at the cost of low efficiency of packet delivery and long delay. When constructing the virtual infrastructure, it is very hard to prevent different unicast tunnels from sharing physical links, which results in redundant traffic on the physical links. Besides, the problem of low delivery efficiency is discussed in Section 4.2.

2.2. Backbone-based multicast protocols

For a backbone-based approach, a distributed election process is conducted among all nodes in the network, so that a subset of nodes are selected as CORE nodes. The topology induced by the CORE nodes and paths connecting them form the virtual backbone, which can be shared by both unicast and multicast routing. In MCEDAR [9], a distributed *minimum dominating set* (*MDS*) algorithm[1] is applied for this purpose, and the resulting backbone has the property that all nodes are within one hop away from a CORE node. A CORE node and its dominated nodes form a cluster. Protocols in [16,17] use different techniques for selecting backbone nodes.

Once a virtual backbone is formed, the multicast operation is divided into two levels. The lower level multicast, which is within a cluster, is trivial. For the upper level multicast, the protocol in [16] uses a pure flooding approach within the backbone. MCEDAR builds a routing mesh, named as *mgraph*, within the virtual backbone, to connect all CORE nodes.

The backbone topology is much more simple and stable than the whole network topology. If backbones are built upon slow-moving nodes, more topology stability is expected even with high host mobility. However, backbone-based method makes each CORE node a "hot-spot" of network

---

[1] Due to the NP-completeness of MDS problem, the distributed algorithm provides approximate solutions. However, a near optimal solution will be enough.

traffic, which poses limits on horizontal scalability. Backbone-based protocols are limited for supporting horizontal scalability. Since data traffic of all the multicast groups should pass the same set of CORE nodes, the number of multicast groups that can be supported by the network is limited by the channel bandwidth at each CORE node.

2.3. Stateless multicast protocols

A recent shift toward stateless multicasting is represented by DDM [12], LGT [13] and RDG [15]. All these protocols do not require maintenance of any routing structure at the forwarding nodes. These protocols use different techniques to achieve stateless multicasting. LGT builds an overlay packet delivery tree on top of the underlying unicast routing protocol, and multicast packets are encapsulated in a unicast envelop and unicasted between the group members. RDG uses a probabilistically controlled flooding technique, termed as gossiping, to deliver packets to all the group members.

In DDM, a source encapsulates a list of destination addresses in the header of each data packet it sends out. When an intermediate node receives the packet, its DDM agent queries the unicast routing protocol about which next-hop node to forward the packet toward each destination in the packet header.

DDM is intended for small groups, therefore, it intrinsically excels only in horizontal scalability. When group size is large, placing the addresses of all members into the packet headers will not be efficient. The protocol has a caching mode, so that only the difference from the previous states is actually placed in the headers. However, as the forwarding set at the on-route nodes inevitably grow large, each intermediate node needs to keep routes for a large set of destinations. This poses a heavy burden on the supporting unicast protocol even under moderate mobility. Further, in order to answer the "next-hop" queries for a large number of destinations, on-demand routing protocols, which are commonly proposed for MANETs, need to flood the entire network very frequently with route discovery packets.

3. Hierarchical multicast

Hierarchical decomposition is an efficient approach to enhance scalability while minimizing overheads of the routing techniques. The basic approach of hierarchical routing has been used to decompose the flat routing structure into non-overlapping logical partitions. Each of these partitions can be further decomposed to form additional levels of hierarchy. Each partition or group within any hierarchical level use a local routing algorithm and the same or a different algorithm can be adopted for inter-level routing. The control overheads are thus reduced significantly, compared to a single flat routing scheme. This basic principle can also be used for hierarchical routing for multicast operations.

3.1. Hierarchical multicast in the Internet

Several flat as well as hierarchical routing protocols have been proposed for supporting multicasting in the Internet [22–25,3–5]. Hierarchical Distance Vector Multicast Routing Protocol (HDVMRP) [3] divides the flat routing region into several non-overlapping domains. Each domain runs its own internal multicast routing protocol, which is DVMRP for the proposal. Inter-domain multicast traffic are routed by another routing protocol at the higher level. Constructing the hierarchical multicast tree in such manner allows heterogeneity among the protocols at different domains and among protocols at different levels. Another hierarchical multicast routing protocol called HIP [4] builds a hierarchical multicast tree by introducing the concept of "virtual router". All border routers of a domain are organized to appear as a single router in the higher level tree. A different way of hierarchical tree building can be named as a "tree of trees," which is used by CBT[5]. In this approach, the leaf nodes of a higher level multicast tree can each be functioning as the root of a lower-level tree.

These protocols for hierarchical multicasting are well-suited for the Internet environment, where characteristics are different from that of MANET environments. These approaches can be aggregated and named as domain-based hierarchical multicasting technique. This technique can be adopted for a variety of networks. After partitioning the network topology into domains, a local multicast protocol is employed within each domain. Local routing protocols operate dedicatedly for its own domain. Any topology change which takes place outside the domain can be ignored. For routing between domains, the same or a different routing protocol is adopted at the higher level of hierarchy.

3.2. Why the same methods cannot be adopted for MANETs?

The hierarchical multicast routing techniques proposed for the Internet cannot be directly adopted for the MANETs. Several issues differentiate the MANET structure which poses problem while implementing the hierarchical Internet multicast routing protocols. As shown in Fig. 1, the Internet is organized as a set of domains. The inter-domain connectivity is provided by having the border routers within each domain linked to the border routers of other domains. According to the HDVMRP protocol, the source node first multicasts to all border routers in its domain. The Level-2 multicast routing is running only on all the border routers, which directs packets to the domains with intended group members. The border routers of the intended domain receives the packets first, and further multicasts them using Level-1 protocol.

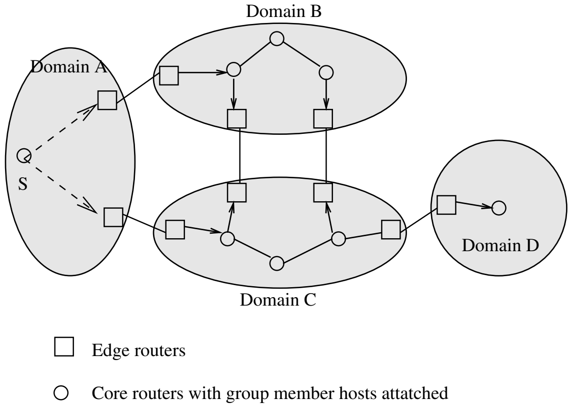


Fig. 1. Internet hierarchical multicast protocol.

Protocols such as HDVMRP are not suited for MANETs. The links in MANETs form in ad hoc manner, and data is transmitted through radio broadcast. Thus, if the network is partitioned into domains, the connection between two domains will be the intersection region of the coverage regions of the two domains. Furthermore, the partitioned domain will neither have the same edge or core nodes at all the times. Adopting hierarchical protocols like HDVMRP requires the fixed designation of edge nodes. In MANETs, the role of edge nodes will be played by different nodes because of the mobility and variable topology. It is thus desirable to explore the feasibility, design issues, trade-offs, and the performance of hierarchical multicasting techniques in MANETs.

4. Framework for hierarchical multicast schemes for MANET

In this section, we present two hierarchical multicast solutions, both of which have the goal of achieving lower multicast overhead and robustness for large-scale multicasting. We refrain from developing a new multicast routing protocol, but present a framework for hierarchical multicasting in MANETs. Based on the framework, a variety of techniques can bo adopted for effective multicasting in MANETs.

A critical component of hierarchical multicasting in MANETs involves the way the multicast tree or mesh are constructed. For the proposed framework, we have formed a generic classification of various possible configurations of hierarchical multicasting in MANETs. This classification is depicted in Fig. 2. The approaches differ in the relationship between two adjacent levels of multicast trees, i.e., how the lower level multicast trees are organized to serve the upper level. In this section, we describe the methodologies of these multicasting techniques.

4.1. Domain-based hierarchical multicast

4.1.1. General approach

A multicast group of large size can be partitioned into certain number of subgroups, so that

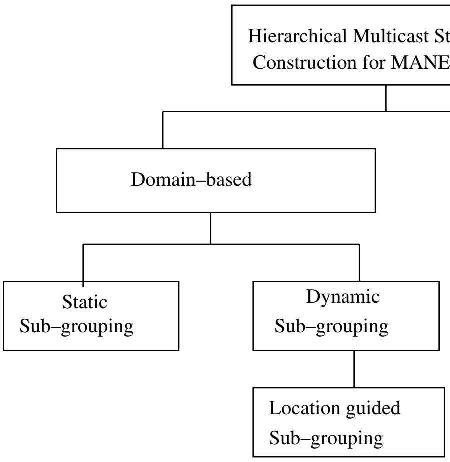


Fig. 2. Different manners of constructing hierarchical multicast trees.

each sub-group is of tractable size. Within each sub-group, a special node is chosen to serve as a sub-root. All source nodes of the group, together with all the sub-roots, form a special sub-group for the purpose of upper level multicast. The source node will first use the upper level multicast tree to deliver packets to all the sub-roots. Then, each sub-root uses the lower level multicast protocol to build its own lower level multicast tree and further delivers packets to its sub-group members.

For all cases, it is safe to partition the multicast group according to relative vicinity. Fig. 3 shows an ideal case of partitioning according to geographical regions. In this example, the shaded

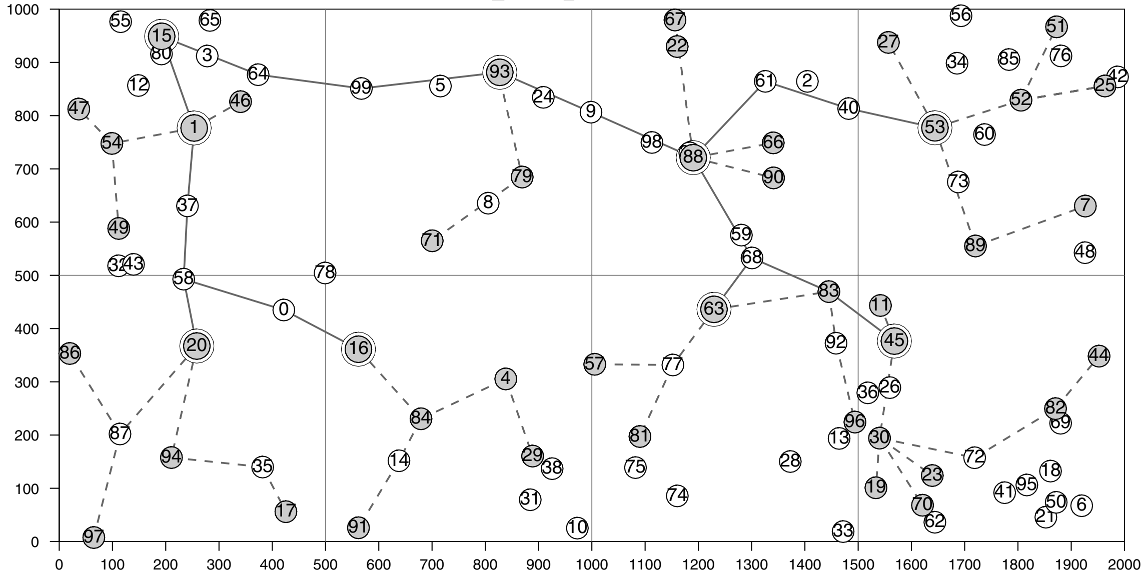


Fig. 3. Hierarchical multicast trees. Shaded nodes are group members. Double circled nodes are selected sub-roots for the domains. The solid lines form the upper-level multicast tree, with node 15 as the root. Dotted lines are the branches of the lower-level multicast trees.

nodes form the multicast group. Node 15 is a source node, and the upper level multicast tree is shown in solid lines, which spans over all sub-roots marked in the figure with double circles. The lower level multicast trees are shown with dotted lines.

Heterogeneity is allowed among the multicast protocols employed at different sub-groups and at the higher level groups. The partitioning approach can be applied recursively to form multiple levels of hierarchical multicast, so that it is possible to support arbitrary large size groups with bounded amount of states maintained at each node. However, for the ease of explanation, we have restricted our discussions to two levels.

4.1.2. *An example: Hierarchical DDM*

In the previous section, the scalability problems of DDM protocol are analyzed. In this section, we propose a hierarchical DDM scheme. The geographical region-based partitioning needs a location service for the network. We do not assume its availability, thus, a topology-aware approach is adopted in our protocol.

The key issue in hierarchical DDM is the hierarchy maintenance, which involves how to optimally partition the multicast group into the sub-groups. In the worst case when distant members are put into one sub-group, the performance will degrade. Specifically, we need to answer the following three questions:

(1) How to build the multicast hierarchy? Specifically, how to partition the multicast group so that adjacent cluster of members can form a subgroup? Also, which node among the nodes in a sub-group is selected as a sub-root?
(2) When a new member joins the group, which sub-group is it assigned to?
(3) An optimal partitioning conducted long ago may not represent the current network topology. How to dynamically adjust the partitioning?

The answers to these questions are proposed as follows.

Group partitioning and sub-root selection. Before partitioning, the source node, denoted as S, only has a flat list of current group members. In order to build the multicast hierarchy according to the current network topology, node S generates a HIER_REQ message. The message contains a small piece of information about the format of the partition. The most important information is the expected size of each sub-group, which is arbitrated by node S. This message is delivered to all group members using the original DDM protocol. Since this is not a network wide broadcast, the cost of the message delivery is mainly proportional to the group size. To further reduce the cost, it can be piggy-backed onto the first data packet.

When a member node, denoted as I, receives the packet carrying this HIER_REQ message, the DDM header of the packet contains a list of members, to which node I is responsible for forwarding the packet. We can view it as the subtree in the multicast tree rooted at node I. Further, this member list is the result of the forwarding process from S to I, representing the most current topology information. If the cardinality of this list matches the intended sub-group size indicated in the HIER_REQ message, node I becomes a candidate for sub-root.

To become a sub-root, node I unicasts back to node S a HIER_REP message. It contains the node I's sub-group member list. Node S need to wait for a period to collect the HIER_REP messages from the member nodes that request to be sub-root candidates. S then partitions the whole member list based on the collected HIER_REPs.

The partition calculation transforms the group member list GL into the form $\{SGL_1, SGL_2, \ldots, SGL_k\}$, in which SGL_i represents the ith subgroup. We denote the root of SGL_i as SR_i. For all the newly selected sub-roots, S need to unicast to SR_i an SR_CONFIRM message, carrying the sub-group member list SGL_i. Upon receiving this message, SR_i recognizes that it succeeds as a sub-root, and record SGL_i as its sub-group member list.

Hierarchy maintenance. If a sub-root dies, the whole sub-group can no longer receive data packets from the source. We thus need a hierarchy maintenance procedure. Periodically, the source node will piggy-back a HELLO message onto a data packet at the upper layer multicast. Upon receiving this message, each sub-root needs to reply with a HELLO_ACK message. Thus, the source node can check each sub-root if the HELLO_ACK has arrived within a threshold of latency. When a sub-root is identified as not functioning, the source needs to assign another node in the same sub-group as the sub-root.

Join and leave operations. According to the original DDM protocol, a new member joins the multicast group by unicasting a join request message to the source node. However, in order to optimally assign a sub-group for a new member to join, hierarchical DDM needs to extend this join process. When node I needs to join the group, it first unicasts a JOIN_REQ to the source node S. According to the status of a group partition process, node S will respond a JOIN_REQ differently. If the partitioning process has finished, S will reply node I a JOIN_SUB message to tell it to start finding a sub-root for itself. Otherwise, if the partitioning has not finished yet, and S still has a flat member list, S will refrain from responding. In this case, node I may try sending JOIN_REQ to S several times as if the packet is lost. When partitioning is done, node I will get a JOIN_SUB respond. When node I receives JOIN_SUB reply, it starts finding its sub-group by broadcasting a SUB_REQ message with a limiting time-to-live (TTL) field value l. The message is flooded in the local space around node I, with a scope up to l hops away. Node I can start with a small TTL value and gradually increase it using the expanding ring search technique adopted in [19]. A sub-root SR_i receiving this SUB_REQ message will not forward the message, but reply a SUB_REP message to I. When node I receives the SUB_REP, it can infer its hop distance from the sending sub-root by checking the unicast routing information. Node I needs to wait for a period collecting SUB_REP messages. Finally, node I can select the nearest responding sub-root, and join its sub-group by replying a SUB_JACK message.

For a normal group member, the leave operation can just follow the same procedure in the original DDM protocol. For a sub-root, when its LEAVE message reaches the source node, the source need to re-assign the sub-root role to another node in the same sub-group. This is the same procedure mentioned in the "Hierarchy Maintenance" part.

Dynamic partition. With node mobility, an optimally calculated group partition will eventually mismatch the current network topology. Some members of a sub-group may move far away and close to the members of another sub-group. Every node in the network is running a DDM agent, forwarding packet for its sub-group, or other sub-groups. A group member node, I, of sub-group SG1 could be forwarding packets for another sub-group SG2. Node I can utilize this chance to decide if it is better to switch sub-group. Whenever node I receives or forwards a data packet, it can query from the unicast routing information to infer its current hop distance to the sub-root sending the packet. Let $h_{i,1}$ and $h_{i,2}$ denote node I's hop distances to the sub-root of SG1 and SG2, respectively. If $h_{i,1} > h_{i,2}$, and their difference exceeds a threshold value, node I will decide that it is better to switch to SG2. In order to switch, node I needs to unicast SUB_REQ message to SR2, sub-root of SG2. When it receives the confirming SUB_REP message from SR2, node I can further unicast SUB_LEAVE message to SR1. Both SR1 and SR2 will need to update its sub-group member list accordingly during this switch process. Note that once the partitioning is finished, the source node only takes care of the upper layer multicast. As long as the member list and the sub-rooting do not change, the source node does not need to know this switching procedure.

Partition sharing among different sources. When there are multiple sources for the same group, the sources should be able to share the group partitioning, thus share the cost as well. For this purpose, one source can serve as the "Core" for the group. Just as other core-based multicast protocols, we assume availability of the service which maps a multicast address to the address of its core. Before sending out data packets, a source node queries the core for the group member list and the current list of sub-roots. The core does not forward data traffic for other sources. There is no single point of failure problem in this design. A member list is the only state needed to function as a core. When a core dies, any source node can take up the role of core.

Discussion on hierarchical DDM. Hierarchical DDM is not purely stateless. The protocol states are the subgroup member lists at the sub-roots. Since the sub-roots are selected by the source node, the distribution of protocol states are flexibly tunable, which is a key advantage compared to the static uncontrollable distribution manner in the backbone-based protocols.

Hierarchical DDM scheme solves the scalability problem of basic DDM. The packet headers are significantly shortened. The load placed on the supporting unicast protocol is also reduced. A forwarding node will only need to serve one or a small number of sub-groups, which is a small fraction of the whole group. This reduced load on the unicast protocols will reduce the unicast overheads significantly when the unicast routing uses on-demand type of protocols.

Algorithm 1. Overlay-driven hierarchical multicast protocol (For all member nodes)

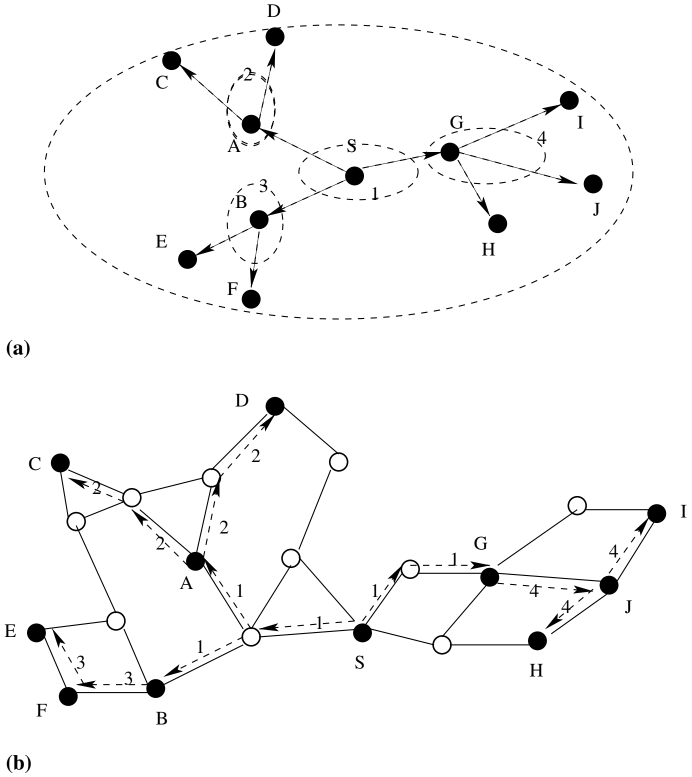
```
Upon this node, P, receiving a data packet from an on-tree neighbor, Q:
    1. Call the overlay routing protocol to update the "Overlay on-tree neighbor list" ($OTN\_LIST_P$);
    2. Generate small group list ($SG\_LIST_P^Q = OTN\_LIST_P - \{Q\}$);
    3. Organize a lower level multicast group for $SG\_LIST_P^Q$;
    4. Pass the data packet to lower level small-group multicast protocol for delivery;
End
```

4.2. Overlay-driven hierarchical multicast

Another method for constructing hierarchical multicasting trees involves the application layer support at the higher levels of multicasting. In this method, an overlay multicast protocol is used to construct the virtual multicast tree. Currently, several such protocols have been proposed specifically for MANET, and the examples are AMRoute [10], LGT [13], PAST_DM [11] and PMA [14]. In this paper, we refrain from proposing another overlay multicast method. Instead, we will focus on how a new hierarchical multicasting method, named as the *overlay-driven hierarchical multicast*, can be derived based on overlay multicast trees. In contrast to domain-based hierarchical multicast, in which the upper level multicast only involves a subset of the group member nodes, the overlay-driven method requires the upper level multicast tree to logically span all the group members.

After the overlay multicast tree is built, the forwarding of data packets are still driven by the virtual tree. Each non-leaf node is responsible of delivering data packets to its children on the virtual tree. With the normal overlay multicast, each node uses several unicasts to deliver the packet to all the children nodes. However, in overlay multicast tree, each non-leaf node uses a small-group multicast session to deliver the packet to all its children nodes simultaneously. Algorithm 1 illustrates the overlay-driven hierarchical algorithm. The procedure should be running at each member node. Fig. 4 illustrates the overlay-driven tree construction method through an example. Fig. 4(a) shows the overlay multicast tree of a session. The root of this tree is at node S. In the example shown in Fig. 4(a), there are four non-leaf nodes (aka. forking points) in the overlay multicast tree, which take place around node S, A, B and G, respectively. With respect to this multicast session, with node S as the source node, each forking point is assigned a unique identification number, named as FORK_ID. The lower level multicasts take place at every forking point. A sub-group at a given forking point is composed of the forking node and its on-tree neighbors. Fig. 4(b) shows all the four lower level multicast trees, with dashed line showing the on-tree edges. Each edge is attached with the FORK_ID of its sub-group. Each tree is rooted at a forking node in the overlay multicast tree. Due to node capacity constraints, the node degrees at the overlay multicast tree are bounded. Thus, the size of each sub-group is always bounded by a small number. A *small group multicast* protocol such as DDM will be ideal at this level.

In contrast to the explicit sub-grouping method employed by domain-based hierarchical multicast, the sub-grouping in overlay-driven hierarchical multicast is conducted in an implicit manner.



(a)

(b)

Fig. 4. Hierarchical multicast trees. (a) Overlay multicast tree. (b) Overlay—driven hierarchical multicast tree.

The difference between the two tree construction methods is the relationship between adjacent levels of multicast trees. Because of design constraints, overlay-driven method can only have two levels of hierarchical multicast, in which the upper level multicast always uses an overlay multicast protocol.

Overlay-driven hierarchical multicast improves data delivery efficiency of overlay multicast. The metric "*stress*" of a physical link is defined in [7] as the number of identical packets it carries. In native multicast routing, it has the optimal value as 1. However, in overlay multicast, a physical link often needs to forward the same packet multiple times. One cause of this phenomenon is the mismatch of the overlay topology and the physical topology. Another cause is that overlay multicast requires each forking node unicast the data packet multiple times to its children nodes. Overlay-driven hierarchical multicast replaces these multiple unicasts into one multicast operation. In the ideal case, which is shown in Fig. 4, all the physical links achieve the optimal stress value.

When an overlay multicast protocol is selected for the upper level multicast, we need to consider if it is using a static or a dynamic virtual mesh. Protocols using static virtual mesh, such as AMRoute, achieve the protocol simplicity and do not have mesh maintenance overhead. The drawback is that as nodes continuously move farther away from its original place, the increasing mismatch between virtual and physical topology will decrease the data delivery efficiency. The physical links cannot achieve optimal stress value even when the proposed hierarchical method is applied. A dynamic virtual mesh is proposed in PAST-DM protocol [11]. With controlled overhead, the virtual mesh topology gradually adapts to the changes of underlying physical topology. If there is no serious mismatch between overlay multicast tree and the physical topology, as shown in Fig. 4, the lower level multicasts can be geographically local and the tree branches will have small hop length. The overlay-driven hierarchical multicast tree will achieve near optimal average stress value.

5. Performance comparison study

In this section, we use a simulation-based study to compare the relative pros-and-cons of the proposed schemes. We use GloMoSim [26] simulator for the following evaluations. At the physical layer, GloMoSim uses a comprehensive radio model that accounts for noise power, signal propagation and reception.

5.1. Simulation setups and performance metrics

In the following simulations, the network field size is 2500 m × 2500 m, containing 400 mobile nodes. All the nodes follow the *random waypoint* mobility model [28] with speed range of 1–20 m/s. We vary the mobility with different pause times as 0, 60, 120, ..., 420, 600, and 900 s. To avoid the initial unstable phenomenon in *random waypoint* model [27,28], we let the nodes move for 3600 s before starting any network traffic [29], which lasts for 900 simulation seconds in each simulation run. For the multicast traffic, the source of multicast session generates packets at a constant rate of 2 packets per second. Each packet is 512 bytes. We are particularly interested in the scalability of the protocols.

The following metrics are used for comparing protocol performances.

1. *Data Delivery Rate*: Percentage of data packets delivered to the receivers.
2. *Data Forwarding Efficiency*: Number of data packet transmissions per delivered packet.
3. *Relative Control Bit Overhead*: Number of control overhead in bits per delivered bit. The transmitted control bits includes the control packets and the bytes in each packet header. For DDM, the involved unicast control bit overhead is also included.
4. *Average Delivery Latency*: Packet delivery latency averaged over all packets delivered to all receivers.

Our simulation includes two parts as follows. In the first part, presented in Section 5.2, we choose to implement the DDM protocol, based on which two hierarchical multicast schemes are also implemented. One is the hierarchical DDM multicast presented in Section 4.1.2, which is named as HDDM. The other is HDDM without dynamic partition, which is named as HDDM-Static. For fairness of comparison, AODV is used as the underlying unicast protocol for both hierarchical DDM protocols. In both HDDM protocols, the minimum and maximum allowable size of each sub-group are 9 and 20, respectively. For performance references, we also run simulation with a mesh based protocol, ODMRP [18].

In the second part of the performance study, presented in Section 5.3, we compare the performance of overlay multicasting using only unicasts versus the proposed overlay-driven hierarchical multicasting. We choose DDM as the lower layer multicast protocol. In order to demonstrate the difference of using DDM multicasts rather than using individual unicasts, we force the two methods to use the same overlay multicast tree. To achieve this goal, a topology of overlay multicast tree is hard-coded into each member node, which remains static through out a simulation run. The same static overlay tree is used for both methods. For the underlying unicast protocol in both cases, AODV is used.

5.2. Performance of hierarchical DDM protocols

In this part of simulation, we change the networks nodal mobility with the *pause time* of the *random waypoint* model. In order to study both vertical scalability and horizontal scalability, we change the group size from 20 to 200 in one group and change the number of groups from 2 to 12.

5.2.1. Performance versus mobility

Fig. 5 presents the performance metrics as functions of pause time. The group size in the simulations is 150. As shown in Fig. 5(a), ODMRP and

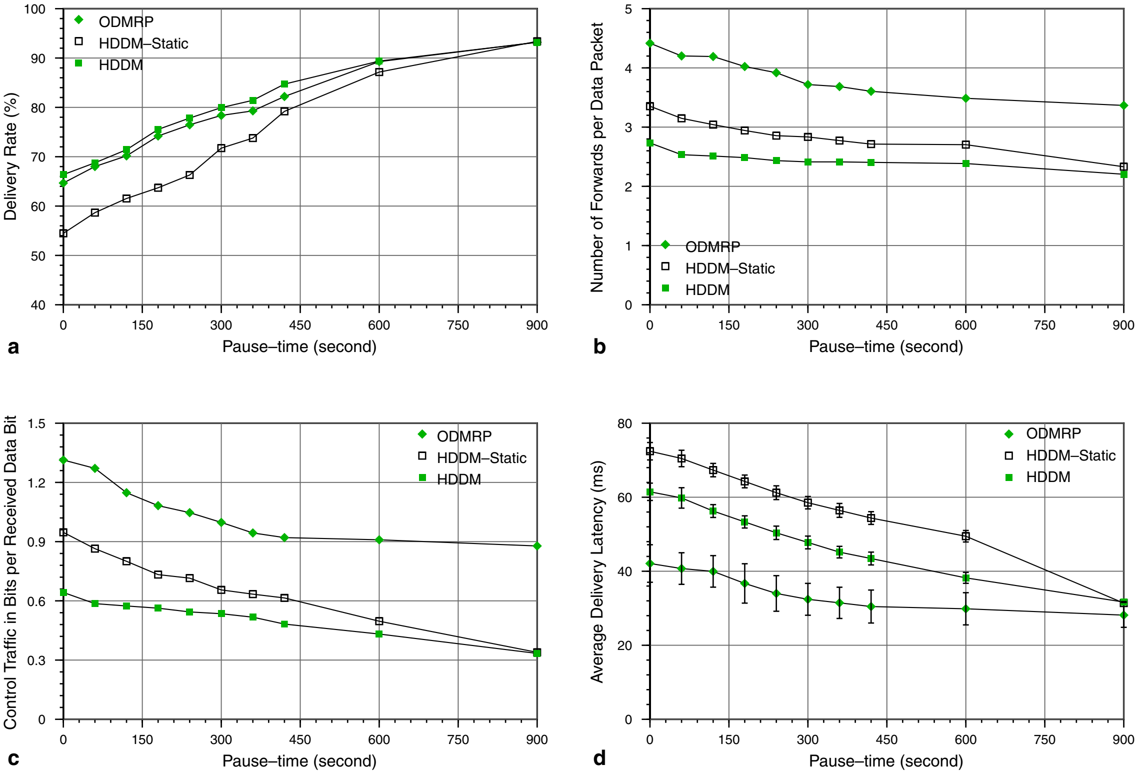


Fig. 5. Performance versus mobility. (Group size is 150, 1 group, 1 source per group.) (a) Packet delivery ratio, (b) forwarding efficiency, (c) normalized bit overhead and (d) average delivery latency.

HDDM achieve similar packet delivery ratio for all pause time setups. HDDM-Static delivers nearly the same amount of data packets in the static scenario (pause time equals 900s). As mobility increases with less pause time, the delivery ratio of HDDM-Static drops faster than the other two protocols. When pause time is low, more nodes will move far away from other nodes in the same sub-group. If nodes can switch to other sub-groups, a sub-root can attract nearby group members to join its sub-group, which reduces the forwarding hops at the lower layer multicast.

Fig. 5(b) and (c) show the results of performance metrics of data delivery efficiency and control overhead. Compared to ODMRP, HDDM achieves slightly better data delivery ratio with much less control traffic and lower network load. ODMRP makes the source node periodically flood the network with JOIN_QUERY messages. The nodes on the shortest path from the source to the receivers form the forwarding group, which relay every data packet they receive. The forwarding group forms a mesh which includes all the source-to-member paths. The mesh's size is fairly large compared to the group size in the simulation settings. Thus, more data packet transmissions are incurred in ODMRP. The control traffic in ODMRP are JOIN_QUERY and JOIN_REPLY packets, while in both HDDM protocols, major part of control traffic is piggy-backed in the packet headers. The high cost of media access in MANET environment favors the in-band signaling styl of control traffic in HDDM. The multicast hierarchy significantly reduces the length of DDM headers. For a group of size 150 members, the average number of destinations in the headers is only 16 for 60 s pause time, which accounts for the much reduced control traffic.

The average delivery latency is shown in Fig. 5(d). The packet delivery latency is averaged for all the delivered packets at each receiver. For each protocol, the averaged value and the variance of the latencies at all receivers are shown by the curve points and the error bars. ODMRP has lower latency than the both HDDM protocols because ODMRP always tries to include the shortest path within the forwarding group. The two-phase delivery paths (from source to sub-roots then to receivers) in HDDM are often longer than the optimal paths. However, we observe that the variance of delay among the receivers in HDDM is much lower than that of ODMRP. The reason is that the lengths of delivery paths for the receivers are unified by the multicast hierarchy. We also observe a gap between the two HDDM protocols. This gap is the effect of dynamic partition, which tries to shorten the delivery path at the lower level multicasts.

5.2.2. Vertical scalability issues

In these simulates, we have one multicast group of size varying from 20 to 200. Fig. 6 shows the performance metrics as functions of group size. In this example scenario, the pause time is set as 60 s. Results for different pause time scenarios show similar trends.

Fig. 6(a) shows the result for packet delivery ratio. As group sizes increase, ODMRP delivers more fraction of packets. The reason is that the forwarding mesh becomes more reliable with more redundant paths as it increases its size. Both HDDM protocols show a stable delivery ratio, with a slight decreasing trend. Irrespective of the group size, the forwarding structure of both HDDM protocols is always a hierarchical tree, which becomes less reliable for a larger group.

Data forwarding efficiency is shown in Fig. 6(b). HDDM is much more efficient in delivering data packets than ODMRP. Though most packets delivered to the receivers do not follow the shortest path, the forwarding load from source to a sub-root is shared among all the members in the sub-group. Thus, hierarchical delivery reduces the data traffic load successfully. The forwarding mesh formed by ODMRP is of relative big size when group size is small, resulting in very inefficient data forwarding process. As group size grow larger, this problem is alleviated.

Fig. 6(c) shows the result of control overhead. The curve for ODMRP first decreases with the increased group size. Though the amount of control packets increases, the number of delivered packets increases faster with more receivers. However, the curve increases again when group size is large than 120. The reason is that the JOIN_REPLY packets sent by the receivers collide more frequently, and the number of retransmissions of JOIN_REPLY increases drastically. Both HDDM protocols show better scalability trend than ODMRP. The control traffic does not increase as fast as the group size. Most control cost by the HDDM protocols are piggy-backed onto the packet headers. If one packet transmission can reach multiple receivers from a forwarding node, the delivered data bits are counted as multiple data packets, while the bit overhead of control traffic is still counted as the bits of one packet header. This in-band signaling feature becomes advantageous when the traffic load is high.

Fig. 6(d) shows the averaged delivery latency and variance among the receivers. Compared to ODMRP, HDDM and HDDM-Static both have higher delay but lower variance. This is the effect of multicast hierarchy mentioned in the previous section. The curve for ODMRP has a greater increasing trend than the other two. The network under ODMRP has much higher traffic load than the hierarchical protocols. Though the packets are using the shortest path in ODMRP, the delay at each link is long when traffic load is high.

We derive the following inferences. As the group size increases, ODMRP has better performance in terms of delivery rate and forwarding efficiency, however, control overhead and delivery latency increases faster than the group size. Both HDDM protocols provide stable performance for all metrics. The scaling trend in control overhead shows HDDM will be efficient for large groups.

5.2.3. Horizontal scalability issues

In this section, we study the performance behaviors with respect to the horizontal scalability. We consider the following 6 scenarios: 72 by 2, 48 by 3, 36 by 4, 24 by 6, 18 by 8 and 12 by 12. Here,

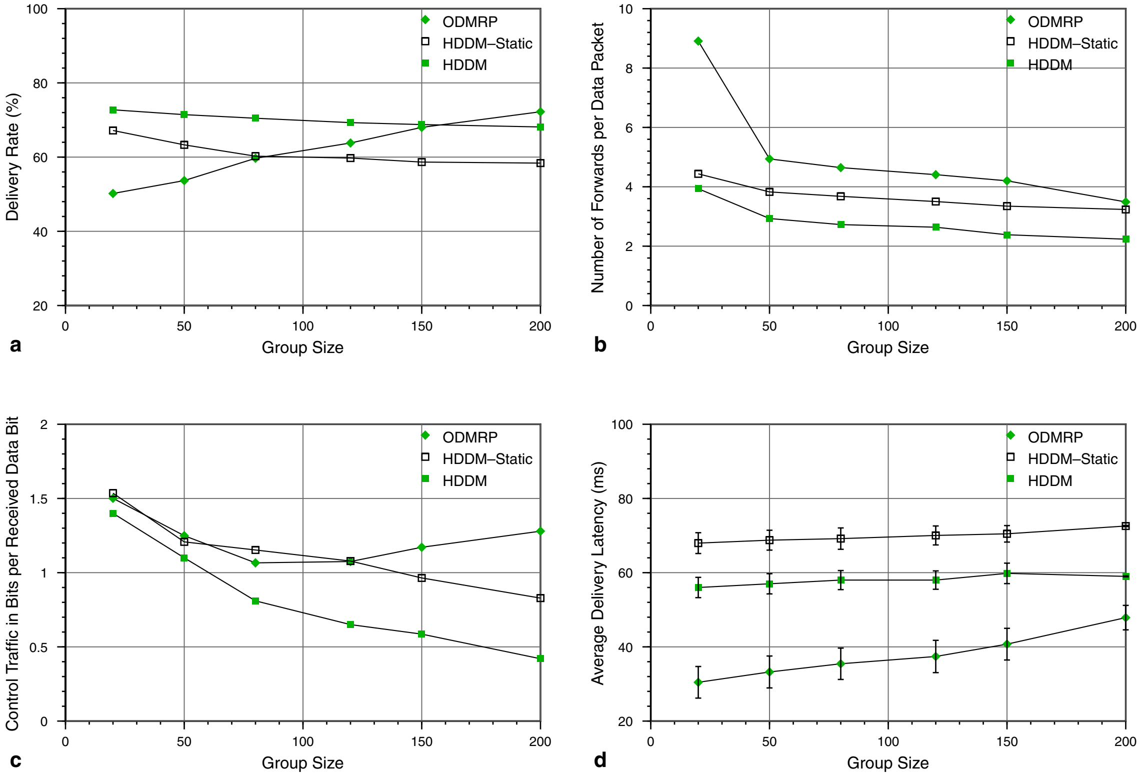


Fig. 6. Performance versus group size. (Pause time is 60 s, 1 group, 1 source per group.) (a) Packet delivery ratio, (b) forwarding efficiency, (c) normalized bit overhead and (d) average delivery latency.

"72 by 2" means 2 multicast groups, and 72 members per group. Thus, in all scenarios, the total number of receivers is fixed to 144. There is one source for each group. The traffic demand remains equal in all scenarios. For the results shown in Fig. 7, the points along the curves show the average value and the error bars show the variances among all the groups in one simulation.[2]

Fig. 7(a) shows the results of packet delivery ratio. As the number of groups increases, performance of ODMRP shows quick drop to less than 10% for 12 groups. With more number of groups, there are more forwarding meshes competing for radio channel. The size of meshes do not decrease proportional to the group sizes. This causes severe traffic jam and packet collisions. Both HDDM and HDDM-Static do not have this problem. As the number of groups increases, the total number of sub-groups and the size of each sub-group remain almost the same. The curve for HDDM finally converges to HDDM-Static when the group number increases to 12. As the group size decreases, the number of sub-groups decreases due to the lower bound on the size of each sub-group. Thus there is less chance for members to switch sub-groups. When group size reduces to 12 in the 12 group scenario, both HDDM protocols reduce to flat DDM.

The results for forwarding efficiency is shown in Fig. 7(b). With more groups of smaller size ODMRP uses much more forwarding transmissions to deliver a data packet. The same trend is found in the previous section, when the group sizes becomes smaller. Both HDDM protocols present more stable curves. With smaller group, the chance

---

[2] In sub-figures (b) and (c) the variance values are too small to be represented in the figures. Thus they are omitted.

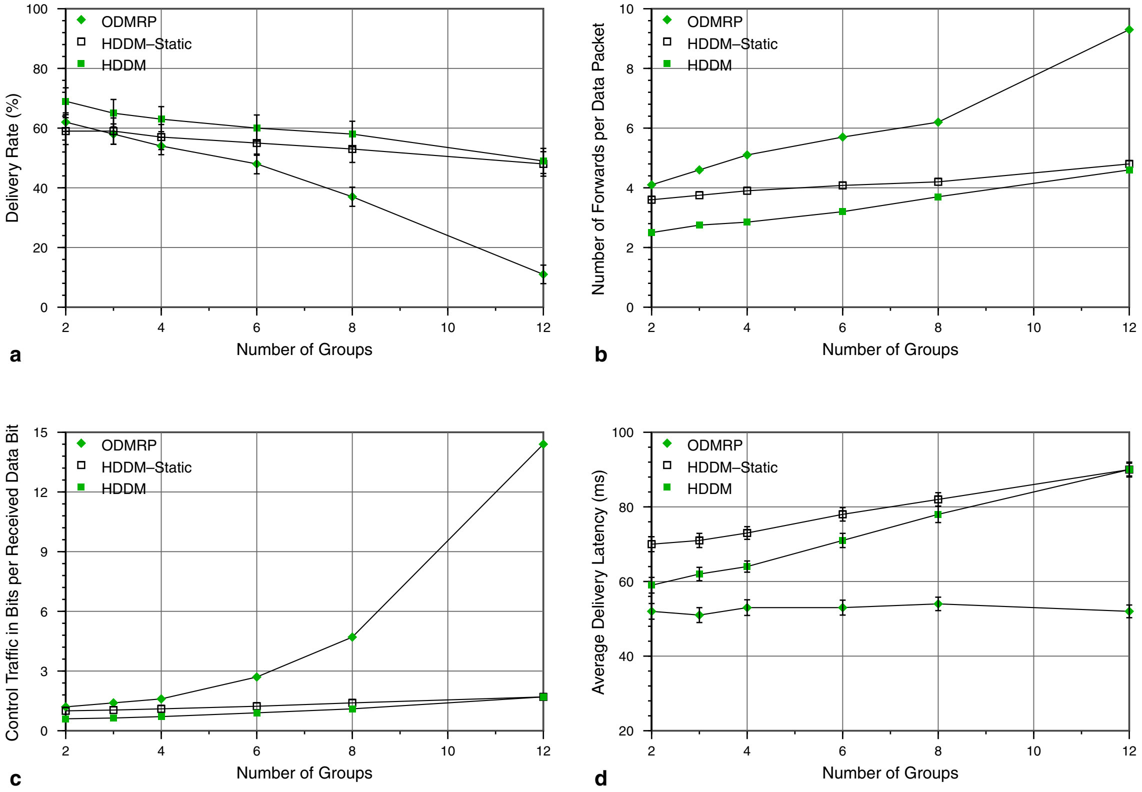


Fig. 7. Performance versus number of groups. (Pause time is 60 s, 1 source per group.) (a) Packet delivery ratio, (b) forwarding efficiency, (c) normalized bit overhead and (d) average delivery latency.

for one broadcast transmission to reach multiple members decreases, thus their curves ascends when the number of groups increases.

Fig. 7(c) shows the results for relative control bit overhead. The control traffic incurred by ODMRP increases dramatically with the increase in the number of groups. In ODMRP, after the source floods the JOIN_QUERY message, all members should reply with JOIN_REPLY packet. These reply packets will cause implosion problem when the group size is large. This problem is solved by aggregating the JOIN_REPLY packets. When two JOIN_REPLY packets reach one node, only one aggregated reply is needed to be forwarded further. However, with many groups of small size, the number of JOIN_REPLY packets is huge and they have less chance to be aggregated. Thus, the control traffic increases significantly. The delivered packets are reduced, and this makes the value of relative control overhead increase even further. Both of the HDDM protocols do not have this problem. The control overhead remains stable with respect to horizontal scalability. The reason for the stability is that for the sub-group multicast level, the number of sub-groups does not change much with different scenarios.

Fig. 7(d) shows the results for average delivery latency and the variance among the groups in the network. This metric favors the case when the delivery ratio is low. In this case, the major part of the delivered packets are those that travel a short hop distance, thus have small delivery latency. Both HDDM protocols have increased delivery latency when number of groups increases. In the case of small number of large groups, the topology-aware partition method tend to make each sub-group only contain adjacent member nodes. In the case of more number of smaller groups, the members of a sub-group become more widely spread in the network. This results in more hops for the packet delivery at lower level multicast groups. Thus the delivery latency becomes larger.

We ran derive the following conclusions. When there are more multicast groups in the network, ODMRP's performance degrades rapidly. Both of the HDDM protocols present very stable behavior in terms of horizontal scalability. When there are more groups, dynamic partitioning becomes less effective.

5.2.4. Multiple source performance

An additional scalability issue to study is how the protocols perform for multiple sources. In Fig. 8, we show how different performance metrics change with regard to increasing number of senders. In the example setup shown in the figure, the group size is 100, and the number of senders are varied from 1 to 8. Fig. 8(a) shows the performance of packet delivery rate. With more senders, data packets from different senders will collide very frequently, reducing the delivery rate. We can observe that the delivery rate of ODMRP drops faster than the two HDDM protocols, because the increased amount of control packets for ODMRP causes more collisions with data packets. In contrast, the control overhead of the HDDM protocols are included in the header of the data packets. The performances of forwarding efficiency are shown in Fig. 8(b). The results show that the number of forwarding hops averaged over all the delivered packets does not change much with different number of senders. This is true for all three protocols. The multicast routing structure (the forwarding group for ODMRP, and the subgroup partitioning for both HDDM protocols) are shared among the different senders. The performance of control overhead are shown in

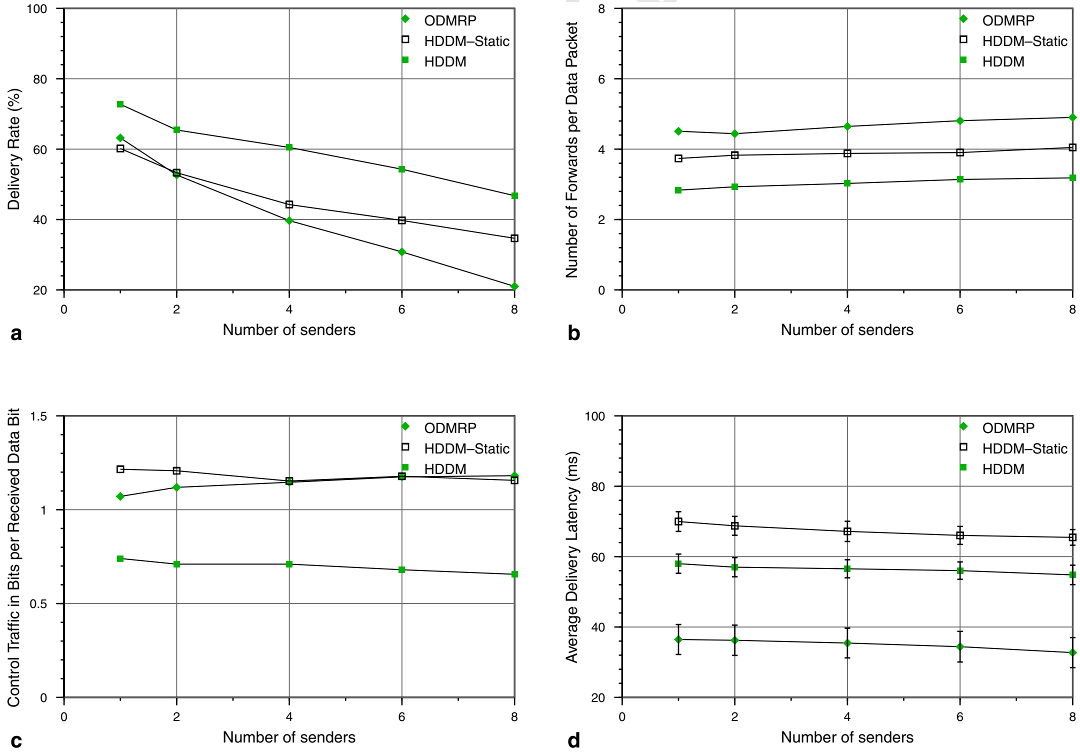


Fig. 8. Performance versus number of senders. (a) Packet delivery ratio, (b) forwarding efficiency, (c) normalized bit overhead and (d) average delivery latency.

Fig. 8(c). We can observe that the normalized bit overhead has slightly decreasing trend for both HDDM protocols, while displaying an increasing trend for ODMRP. We analyzed the reasons, which are the following. The control overhead of HDDM protocols are from two sources: the cost of longer headers at each data packet, and the control cost of the underlying unicast protocol for maintaining routs for HDDM protocol. The second part of the cost are shared by all senders. Since the HDDM-static protocol do not optimize the sub-group partitioning, it has more unicast overhead than the other HDDM protocol. For the ODMRP protocol, since each source needs to periodically flood the network in order to maintain the forwarding group, the control overhead increases with more sources in the group. Due to more collisions, the delivered packets do not increase in proportion to the increased control overhead. Finally, the performance of packet delay is shown in Fig. 8(d). ODMRP still has smaller average delay, but the variance of the delay among the packets is higher than the two HDDM protocols. Note that the packet delay averaged over only delivered packets is more favorable over the case when the delivery rate is low, because of the reasons discussed in the previous section for Fig. 7(d). With more senders, the delivery rate drops significantly for all three protocols. Thus the average packet delay does not change noticibly for multiple senders.

5.3. Overlay-driven multicast protocols

In this part of simulation study, we focus on the performance of both multicast methods with regard to different virtual tree topologies. We vary the nodal fan-out degree from 5 to 10 to achieve both a "thin" tree and a "fat" tree. Fig. 9 shows the topologies of overlay trees for a group of 80 members, both for low fan-out and high fan-out degrees. For both topologies, we first emulate an overlay by making packet deliveries, following the overlay tree, only using unicasting. We then simulate an overlay driven hierarchical multicast by using DDM at each forking point of the overlay tree. We vary the group size from 20 to 200 and the simulation results are shown in Fig. 10. The curves labeled "Unicast-d-5" and "Unicast-d-10" are for performances of overlay method using the "thin" tree and the "fat" tree, respectively. Similarly, "DDM-d-5" and "DDM-d-10" are for overlay-driven methods under both topologies. Notations "d-5" and "d-10" represent fan-out degrees of 5 and 10, respectively.

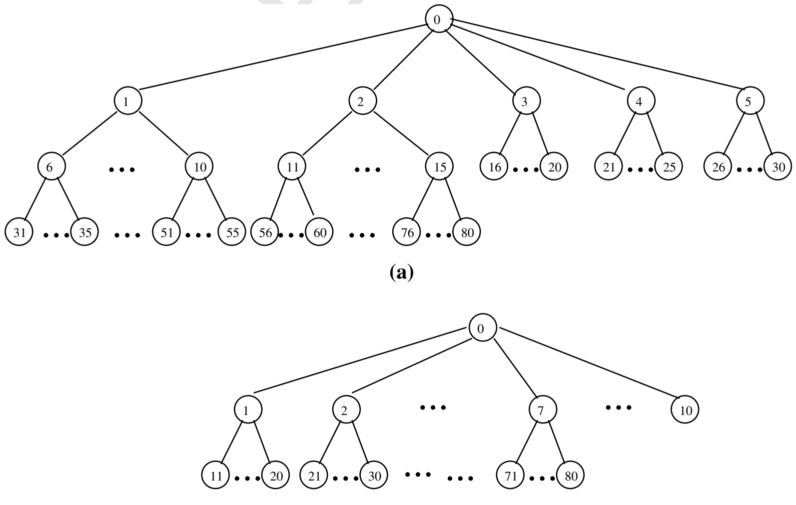


Fig. 9. Topology of overlay tree with low and high fan-out degrees. (a) "Thin" overlay tree and (b) "fat" overlay tree.

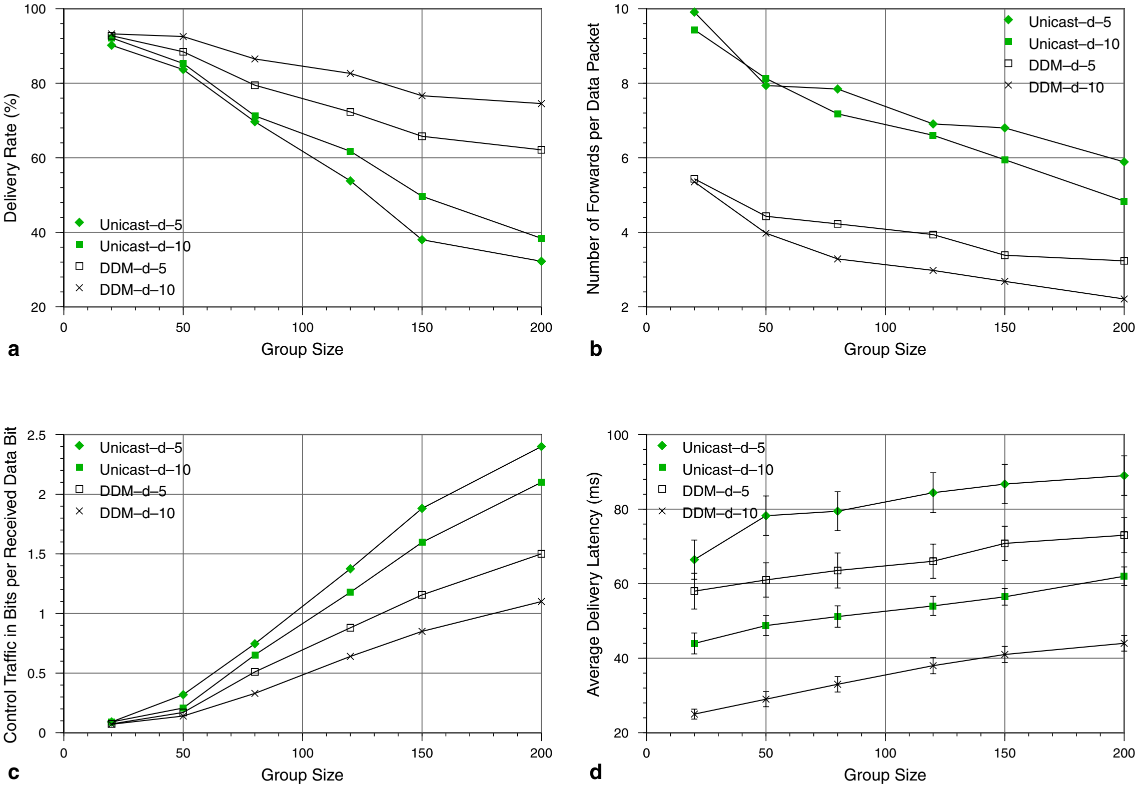


Fig. 10. Performance of overlay driven methods versus group size. (Pause time is 60 s, 1 source per group). (a) Packet delivery ratio, (b) forwarding efficiency, (c) normalized bit overhead and (d) average delivery latency.

The performance of packet delivery ratio is shown in Fig. 10(a). As the group size increases, the delivery load increases from very moderate load to very high. This is depicted by the drop from 90% at 20 members to 32% at 200 members in the curve for “Unicast-d-5”. However, the delivery’ ratio drop for the “Unicast-d-10” curve is less significant. With a higher node degree, the height of the virtual tree is reduced, especially when the group is larger. This accounts for its better performance than the “thin” but “tall” virtual tree. When the group size increases, the performance of both overlay-driven methods are much better than the overlay methods. As the group member become denser, up to 50% when group size is 200, it is more likely that one packet. transmission can reach multiple members. This opportunity is exploited by the overlay-driven methods. This is the main reason for its better scalability. Among the two virtual tree topologies, the “fat” tree provides better performance, which can be explained by the same reason of the reduced tree height.

In Fig. 10(b), which shows the forwarding efficiency with regard to group size, the gap between simple overlay method and overlay-driven method are more significant. By exploiting the chance of delivering to multiple member nodes by one transmission, overlay-driven method reduces the number of transmissions by more than 50%. When group size is only 20, the height of a “fat” tree is the same as that of a “thin” tree. The performance difference cannot be observed. However, from 50 and up, the difference becomes more with the increase in group size.

Fig. 10(c) shows the performance of control overhead. With increased group size, the underlying unicast protocol has to maintain routes for more destinations. The significant increase in the unicast overhead is displayed in the figure with increased normalized bit overhead, even though the number of delivered packets has increased. With more group members, the normalized bit overhead for overlay-driven methods are much less than the simple overlay methods. The major reason for this trend is that the overlay-driven methods deliver much more data packets than the simple overlay methods.

The performance of packet delay and the variance are shown in Fig. 10(d). Among the four curves in the figure, the upper two curves are for "Unicast-d-5" and "DDM-d-5", respectively. The height of overlay tree is much larger when the fan-out degrees in the tree is lower, which means the physical paths taken from the source to the receivers on the tree leaves are far longer than the optimized path. This aspect further results in higher packet delays, and also traffic good-put. We can also observe that the delay variance are much lower for the packets delivered with "fatter" overlay trees for both "Unicast-d-10" and "DDM-d-10". Thus, we can conclude that a fatter overlay tree is more suitable for overlay multicast.

6. Related work

For multicast in the Internet, the issue of forwarding state management and state scalability has been recognized and studied by several researchers. Hierarchical multicast methods [3–5] are one approach for the reduction of forwarding states. In [34], Gerla et al. have proposed a framework for reducing multicast protocol state by "aggregated multicast". It forces aggregated multicast multiple groups to share one distribution tree. Core routers need to keep states only per aggregated tree instead of per group. This can significantly reduce the total number of trees in the network and thus reduce forwarding states. In [33], Thaler and Handley have studied the aggregatibility of multicast forwarding state at the routers. Their analytical and simulation studies show that certain amount of state aggregation is achievable, even under totally random multicast address allocation and random group memberships. They also presented an interface-centric data structure model which allows aggregation of ranges of multicast addresses in the forwarding table. The protocols for hierarchical multicasting are well-suited for the Internet environment, where characteristics are different from that of MANET environments.

In MANET, a few schemes [30,31] have proposed to build a virtual hierarchy in a wireless multi-hop network. This hierarchy is built by various clustering methods, and can be used for better support of a number of network-wide operations, such as multimedia transport and QoS provisioning. PHAM (Physical Hierarchy-driven Ad Hoc Multicast) [32] is a specially tailored multicast algorithm for the MANETs with physical hierarchy. It is assumed that the network is organized in physical groups. Each physical group has a super node which has more capabilities, such as transmission power and computation power. Our hierarchical multicast algorithms, however, assumes a flat network structure.

7. Conclusion

In this paper, we apply the hierarchical routing principle to MANET multicast routing. We categorize the current multicast routing protocols by the amount and distribution of the protocol states. We also study the scalability issues of each category. We propose two different approaches for hierarchical multicast tree construction: domain-based method and overlay-driven method. The domain-based method uses the topological vicinity of nodes to form different levels of hierarchy. At each level, the same or different multicasting protocol can be adopted. By keeping the group size small at each of the levels, efficient small group multicasting protocol could be adopted. The overlay-driven approach uses two levels of hierarchy; the higher level is an overlay topology and the lower level is formed around the nodes of the overlay topology. For the purpose of evaluation, we have used the DDM multicasting scheme that has been shown to be very efficient for small groups.

We presented a detailed performance evaluation of the proposed hierarchical multicasting techniques. The simulation results have demonstrated the performance benefits, enhanced scalability, and low overheads associated with the proposed techniques. A comparative study of variations of our techniques is also presented and the relative merits of these techniques for different mobility and size of MANETs are analyzed.

For the future work, we identify the need to develop a light-weight but reliable multicast protocol for small groups. It can be applied to the upper level multicast in the routing hierarchy to achieve better reliability in packet delivery.

Acknowledgement

This research was supported in part by the National Science Foundation under the grants CCR-0296070 and ANI-0296034.

References

[1] C. de Morais Cordeiro, H. Gossain, D.P. Agrawal, Multicast over wireless mobile ad hoc networks: present and future directions, IEEE Network 17 (1) (2003).
[2] S.J. Lee, W. Su, J. Hsu, M. Gerla, R. Bagrodia, A performance comparison study of ad hoc wireless multicast protocols, in: Proc. IEEE Infocom'00, Tel-Aviv, Israel, March 2000.
[3] A.S. Thyagarajan S.E. Deering, Hierarchical distance-vector multicast routing for the MBone, in: Proc. ACM SIGCOMM'95, Cambrige, Massachusetts, Sep. 1995.
[4] C. Shields, J.J. Garcia-Lunar-Aceves, The HIP protocol for hierarchical multicast routing, in: Proc. 17th Annual ACM Symposium on Principles of Distributed Computing, Puerto Vallarta, Mexico, June 1998.
[5] C. Shields, J.J. Garcia-Luna-Aceves, The ordered core based tree protocol, in: Proc. IEEE Infocom'97, Kobe, Japan, April 1997.
[6] H. Eriksson, MBone: The multicast backbone, Commun. ACM 37 (1994) 54–60.
[7] Y. Chu, S. Rao, H. Zhang, A case for end system multicast, in: Proc. ACM SIGMETRICS'00, Santa Clara, CA, June 2000.
[8] C. Abad, W. Yurcik, R.H. Campbell, A survey and comparison of end-system overlay multicast solutions suitable for network-centric warfare, in: SPIE Defense and Security Symposium/BattleSpace Digitization and Network-Centric Systems IV, 2004.
[9] P. Sinha, R. Sivakumar, V. Bharghavan, MCEDAR: Multicast Core-Extraction Distributed Ad Hoc Routing, in: Proc. IEEE WCNC'99, September 1999.
[10] J. Xie, R.R. Talpade, A. Mccauley, M. Liu, AMRoute: ad hoc multicast routing protocol, ACM Mobile Networks Appl. 7 (6) (2002).
[11] C. Gui, P. Mohapatra, Efficient overlay multicast for mobile ad hoc networks, in: Proc. IEEE WCNC'03, New Orleans, LA, March 2003.
[12] L. Ji, M.S. Corson, Differential destination multicast—A MANET multicast routing protocol for small groups, in: Proc. IEEE Infocom'01, Anchorage, Alaska, April 2001.
[13] K. Chen, K. Nahrstedt, Effective location-guided tree construction algorithms for small group multicast in MANET, in: Proc. IEEE Infocom'02, New York, NY, June 2002.
[14] L. Xiao, A. Patil, Y. Liu, L.M. Ni, A.-H. Esfahanian, Prioritized overlay multicast in mobile Ad Hoc environments, IEEE Comput. 37 (2) (2004).
[15] J. Luo, P.T. Eugster, J.-P. Hubaux, Route driven gossip: probabilistic reliable multicast in ad hoc networks, in: IEEE Infocom'03, San Francisco, CA, March–April 2003.
[16] C. Jaikaeo, C-C. Shen, Adaptive backbone-based multicast for ad hoc networks, in: Proc. IEEE ICC'02, New York, NY, April–May 2002.
[17] M. Gerla, C.-C. Chiang, L. Zhang, Tree multicast strategies in mobile, multihop wireless networks, ACM Mobile Networks Appl. 4 (1999) 193–207.
[18] S.J. Lee, M. Gerla, C.-C. Chiang, On demand multicast routing protocol, in: Proc. IEEE WCNC'99, September 1999, pp. 1298–1302.
[19] E.M. Royer, C.E. Perkings, Multicast operations of the ad-hoc on-demand distance vector routing protocol, in: Proc. ACM MOBICOM'99, Seattle, WA, August 1999.
[20] J.J. Garcia-Luna-Aceves, E.L. Madruga, The core-assisted mesh protocol, IEEE J. Select. Areas Commun. 17 (8) (1999) 1380–1394.
[21] S.K. Das, B.S. Manoj, C.S.R. Murthy, A dynamic core based multicast routing protocol for ad hoc wireless networks, in: Proc. ACM MOBIHOC'02, Lausanne, Switzerland, June 2002.
[22] T. Pusateri, Distance Vector Multicast Routing Protocol, Internet-Draft, March 1998. Work in progress.
[23] A.J. Ballardie, Core Based Trees (CBT version 2) Multicast Routing Protocol Specification, Internet-Draft, July 1997. Work in progress.
[24] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, L. Wei, Protocol Independent Multicast Version 2, Dense Mode Specification, Internet-Draft, May 1997. Work in progress.
[25] D. Estrin-, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, Internet-Draft, September 1997. Work in progress.
[26] GloMoSim, Available from: <http://pcl.cs.ucla.edu/projects/glomosim/>.
[27] T. Camp, J. Boleng, V. Davies, A survey of mobility models for Ad Hoc network research, Wireless Commun. Mobile Comput. (WCMC) 2 (5) (2002) 483–502.
[28] J. Yoon, M. Liu, B. Noble, Random Waypoint Considered Harmful, in: Proc. IEEE Infocom '03 San Francisco, CA, March–April 2003.

[29] BoinnMotion Project, Available from: <http://web.informatik.urii-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/>.

[30] C.R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE J. Select. Areas Commun. 15 (7) (1997) 1265–1275.

[31] R. Ramanathan, M. Steenstrup, Hierarchically-organized, multihop mobile wireless networks for quality-of-service support, ACM/Baltzer Mobile Networks Appl. 3 (1) (1998) 101–119.

[32] Y.-B. Ko, S.-J. Lee, K.-Y. Lee, A multicast protocol for physically hierarchical ad hoc networks, in: Proc. IEEE VTC Jeju, Korea, April 2003.

[33] D. Thaler, M. Handley, On the aggregatability of multicast forwarding state, Proc. IEEE Infocom'00 (Mar) (2000).

[34] A. Fei, J. Cui, M. Gerla, M. Faloutsos, Aggregated Multicast: an Approach to Reduce Multicast State, in: Proceedings of Sixth Global Internet Symposium in conjunction with Globecom 2001, November 2001.

Chao Gui is a Ph.D. candidate in the Department of Computer Science at the University of California, Davis. His research interests include wireless networking and mobile computing. He received an MS in computer science from the University of Central Florida.

Prasant Mohapatra is a professor in the Department of Computer Science at the University of California, Davis. His research interests include wireless networks, sensor networks, and quality of service issues in Internet and Web servers. He received a Ph.D. in Computer Engineering from Pennsylvania State University.