International Journal of Soft Computing 2 (3): 426-439, 2007

Agent Based Multicast Routing in Mobile Ad Hoc Networks

[1]E. Baburaj and [2]V. Vasudevan
[1]Catherine Cottage, Alanchy, Palapallam Via Kanyakumari District, Tamil Nadu, South India
[2]Arulmigu Kalasalingam College of Engineering, India

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. These protocols are implemented within RETSINA agents. In this study, 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-based, are proposed. We have considered a variety of approaches that are suitable for different mobility patterns and multicast group sizes. Results obtained through simulations demonstrate enhanced performance and scalability of the proposed techniques.

Key words: Hierarchical multicasting, Mobile Ad Hoc networks, domain-based multicasting, overlay multicasting, stateless, multicasting, scalability

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 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 infrastructurebased wireless networks. There has been a plethora of work reported on point-to-point communications in MANETs using unicast routing techniques (Perkins and Bhagwat, 1994; Perkins and Royer, 1999; Johnson and Maltz, 1996). However, most potential 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 (Morias *et al*., 2003) and the performance comparison of some of these protocols are discussed in Lee *et al*. (2002). In multicast routing, a shift towards protocol state reduction and constraining in multicast provisioning is represented by hierarchical multicast (Thyagargian and Deering, 1995; Shields and Garcia, 1998, 1997) and overlay multicast (Eriksson, 1994; Chu *et al*., 2000; Kwon and Fahmy, 2002) in the Internet and recent works in MANET multicast (Sinha *et al*., 1999; Xie *et al*., 2002; Gui and Mohapatra, 2003; Ji and Corson, 2001; Chen and Nahrstedt, 2002; Luo *et al*., 2003). Among the MANET multicast protocols, AMRoute (Adhoc Multicast Routing Protocol) (Xie *et al*., 2002) and PAST-DM (Progressively Adapted Sub-Tree algorithm on Dynamic Mesh) (Gui and Mohapatra, 2003), are overlay multicast protocols, which constrain the protocol state within the group members. Backbone-based protocols, such as MCEDAR (Sinha *et al*., 1999) and protocols reported in (Jaikaeo and Shen, 2002; Gerla *et al*., 1999) 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

Corresponding Author: E. Baburaj, Anna University,Chennai, India

DDM (Differential Destination Multicast) (Ji and Corson, 2001), LGT(Location Guided Tree construction algorithms) (Chen and Nehrstadt, 2002) and RDG (Route Driven Gossip) (Luo *et al*., 2003). We study the relationship of the protocol state management techniques and the performance of multicast provisioning. For performance, we focus on protocol control overhead and protocol robustness. We are further interested in the following 2 questions.

- Will the state constraining methods successfully reduce the protocol control overhead?
- When the multicast service scales up vertically (in terms of the group size) and horizontally (in terms of the number of groups), how the scalability will 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. Using overlay multicast as the upper layer multicast protocol and stateless small group multicasts as lower layer multicast protocol, this hierarchical multicast solution achieves protocol robustness, as well as efficient data delivery. These features make overlay multicast approach more suitable for the MANET environment. We study the protocol performance using simulations of large network size (400 moving nodes). We simulate protocol scalability behaviors with group size up to 200 members and number of groups up to 12. The results show robust scalable performance of the domain-based hierarchical multicast scheme proposed in this study.

MULTICASTING IN MANETS: STATE MANAGEMENT AND SCALABILITY

State management of multicast protocols involves timely updating 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 (Lee *et al*., 1999; Royer and Perkins, 1999; Garcia and Madruga, 1999; Das *et al*., 2002) 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.

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. AMRoute (Xi *et al*., 2002) 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 multicast 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 this study.

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 (Sinha *et al*., 1999) a distributed Minimum Dominating Set (MDS) algorithm1 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. The proposed protocol in (Jaikaeo and Shen, 2002; Gerla *et al*., 1999) 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 (Jaikaeo and Shen, 2002) 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 backbone are built upon slowmoving nodes, more topology stability is expected even with high host mobility. However, backbone-based method makes each core node a "hot-spot" of network 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.

Stateless multicast protocols: A recent shift towards stateless multicasting is represented by DDM (Ti and Corson, 2001), LGT (Chen and Nahrstedt, 2002) and RDG (Luo *et al*., 2003). 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. When an on-tree node receives a data packet from its parent node, it gets the identities of its children from the infomation included in the header of the packet. For RDG, a probabilistically controlled flooding technique, termed as gossiping, is used 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 towards 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.

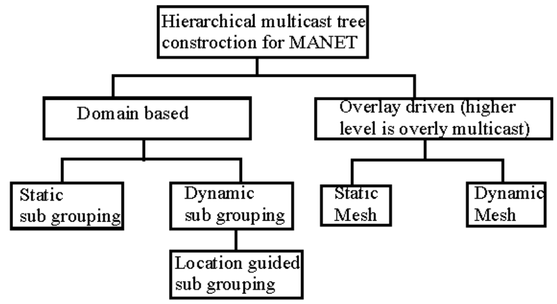

Fig. 1: Different manners of constructing hierarchical multicast trees

HIERARCHICAL MULTICASTING

Hierarchical routing (Kleinrock and Kamoun, 1977) approach can be used to significantly reduce the protocol states in a large scale network. In this study, 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 be 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. 1.

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.

Domain-based hierarchical multicast

General approach: A multicast group of large size can be partitioned into certain number of sub-groups, so that 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

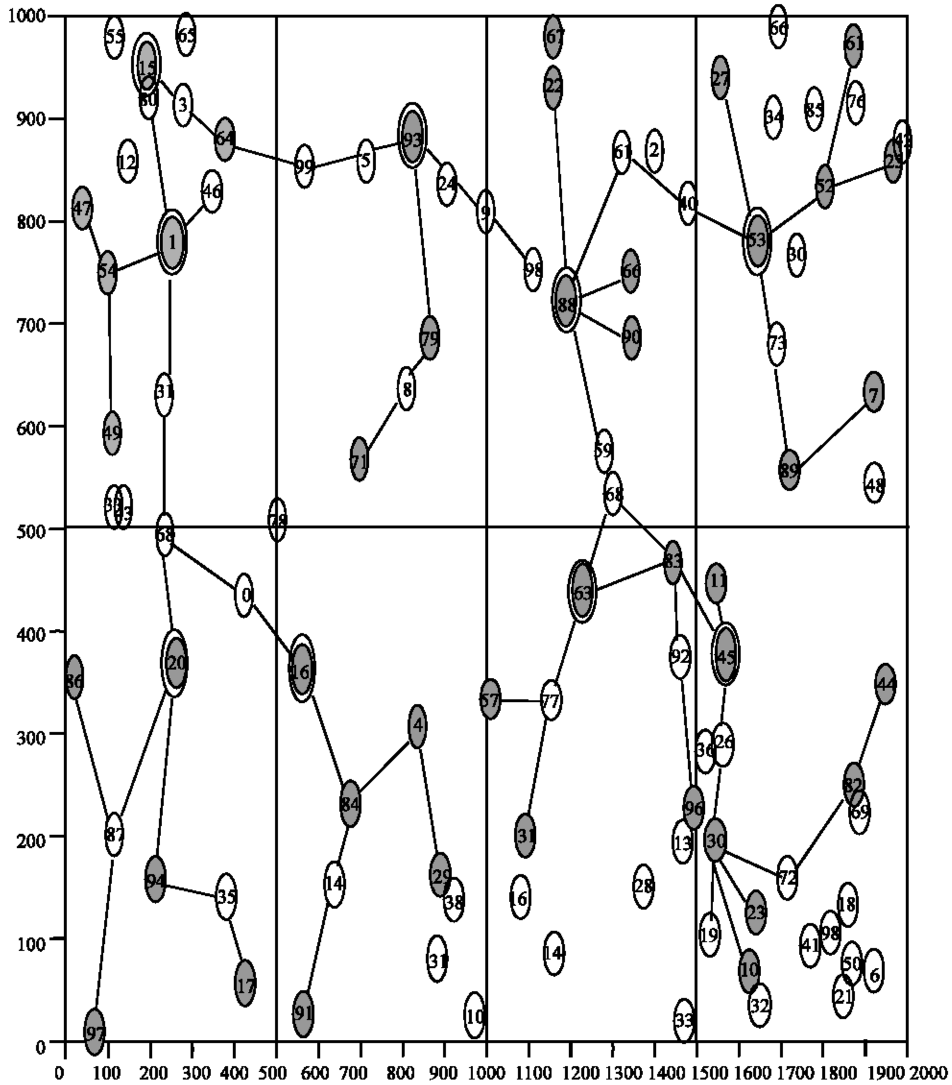

Fig. 2: Hierarchical multicast trees. Shaded nodes are group members. Double circled nodes are selected sub-roots for the domains. The solid lines form the upper-level multicast tree, with node 15 as the root. Dotted lines are the branches of the lower-level multicast trees

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. Figure 2 shows an ideal case of partitioning according to geographical regions.

In this example, the shaded 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.

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 (Grossglauser and Vetterli, 2003) of 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 the distant members are put into one sub-group, the performance will degrade. Specifically, we need to answer the following three questions:

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

The answers to all 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 on the format of the partition. The most important information is the expected size of each subgroup, 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 {SGL1, SGL2,•••, SGLk}, in which SGLi represents the *i*-th sub-group. We denote the root of SGLi as SRi. For all the newly selected sub-roots, S need to unicast to SRi an sr confirm message, carrying the subgroup member list SGLi. Upon receiving this message, Sri recognizes that it succeeds as a sub-root and record SGLi 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 piggyback a Hello message onto a data packet at the upper layer multicast. Upon receiving this message, each sub-root needs to reply with an 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 subgroup subgroup 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 Perkins and Royer (1999). A sub-root *SRi* 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 from 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 mis-match the current network topology. Some members of a sub-group may move far away and close to the embers 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 *hi,*1 and *hi,*2 denote node I's hop distances to the sub-root of SG1 and SG2, respectively. If *hi,*1 > *hi,*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 subrep message rom 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. 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. 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 reside at the sub-roots as the sub-group member lists. 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

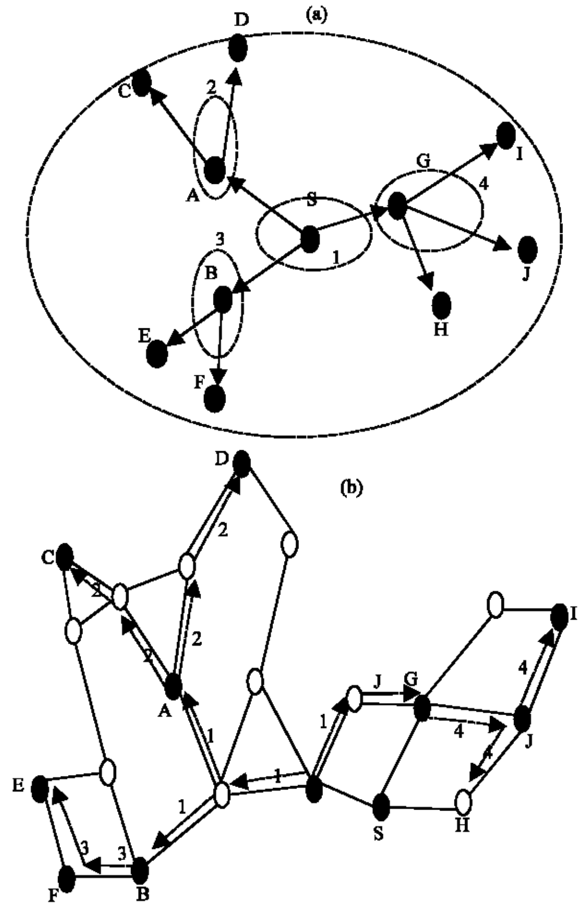

Fig. 3: Hierarchical multicast trees, a): Overlay multicast tree, b): Overlay-driven hierarchical multicast tree

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.

Overlay-driven hierarchical multicast: Another method for constructing hierarchical multicasting trees can be achieved through the application layer support at the higher levels of multicasting. 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. Another 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. Figure 3 illustrates the overlay-driven tree construction method through an example. Figure 3(a) shows the overlay multicast tree of a session. The root of this tree is at node S. In the domainbased method, the upper level multicast only involves a subset of the group member nodes. However, overlay-driven method requires the upper level multicast tree to logically span all the group members. After the overlay multicast tree is built, the sub-grouping for the lower level multicast is implicitly done. In the example shown in Fig. 3a, there are 4 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 FOLK 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. Figure 3(b) shows all the four lower level multicast trees, with dashed line showing the on-tree edges. Each edge is attached with the FOLK 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. Algorithm 1 illustrates the overlay-driven hierarchical algorithm. The procedure should be running at each member node.

Algorithm 1: Overlay-driven hierarchical multicast protocol(For all member nodes)
Upon this node, P, receiving a data packet from an on-treeneighbor, Q*:*

- Call the overlay routing protocol to update the "Overlay on-tree neighbor list"(OTN LISTP);
- Generate small group list (SG LISTQ)
  P = OTN LISTP - {Q});
- Organize a lower level multicast group for SG LISTQP;
- Pass the data packet to lower level small-group multicast protocol for delivery;

End: Overlay-driven hierarchical multicast improves data delivery efficiency of overlay multicast. The metric stress of a physical link is defined in Chu and Zhang (2000) 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 mis-match 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. 3, 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 (Gui and Mohapatra, 2003).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. 3, 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.

PERFORMANCE COMPARISON STUDY

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

Simulation setup and performance metrics: In the following simulations, the network field size is 2500×2500 m, containing 400 mobile nodes. All the nodes follow the random waypoint mobility model (Yoon *et al.*, 2003) with speed range of 1 to 20 m s^{-1}. We vary the mobility with different pause times as 0, 60, 120, ..., 420, 600 and 900 sec. To avoid the initial unstable phenomenon in random waypoint model (Camp *et al.*, 2002; Yoon *et al.*, 2003) we let the nodes move for 3600 sec before starting any network traffic (BMP), 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

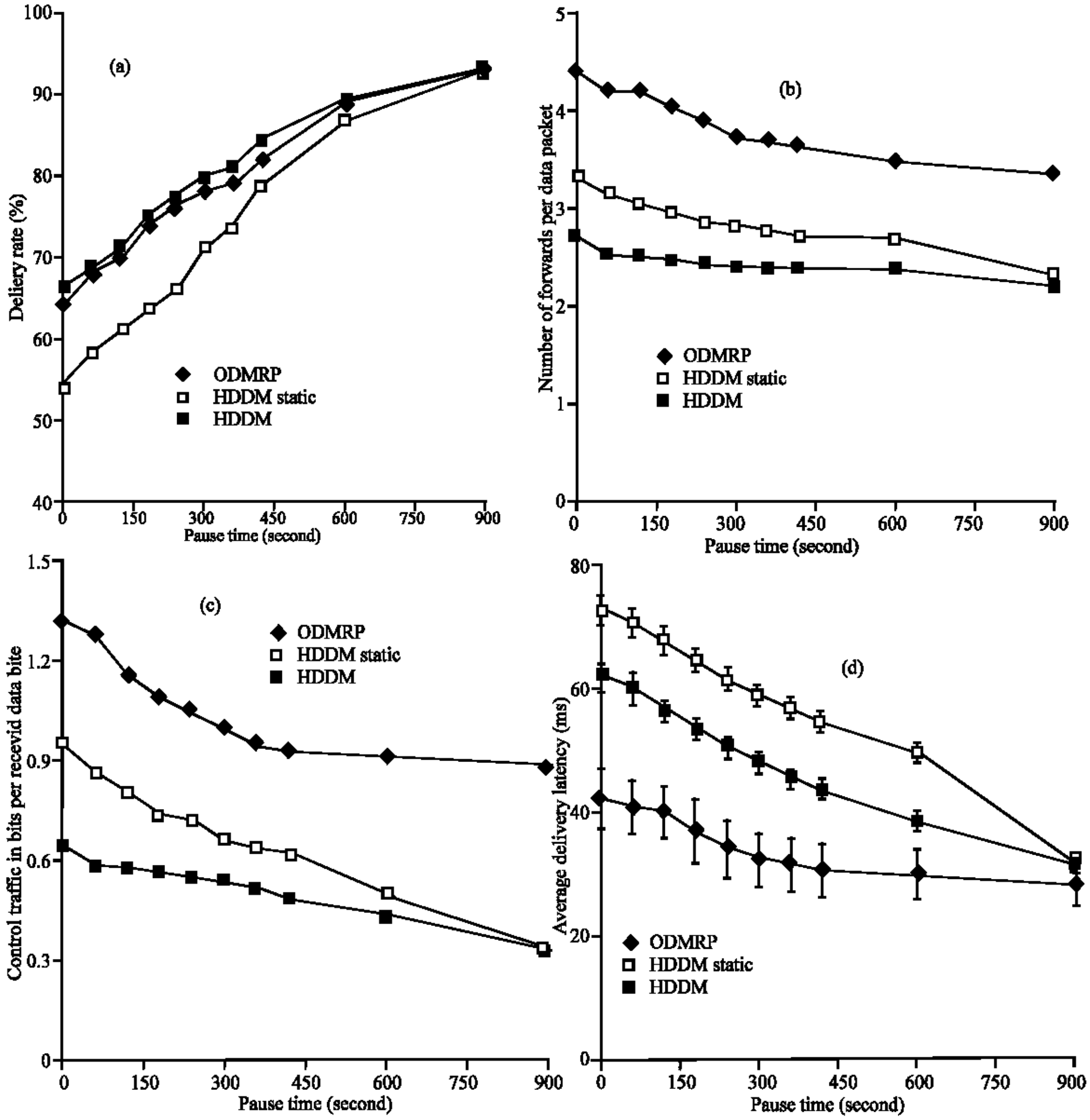

Fig. 4: Simulation setup and performance metrics(a-d), a): Packet delivery ratio, b): Forwarding efficiency, c): Normalized Bit Overhead, d): Average delivery latency

interested in the scalability of the protocols. The following metrics are used for comparing protocol performances.

- Data delivery rate: Percentage of data packets delivered to the receivers.
- Data forwarding efficiency: Number of data packet transmissions per delivered packet.
- 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.
- Average delivery latency: Packet delivery latency averaged over all packets delivered to all receivers. In this simulation, we choose to implement the DDM protocol, based on which two hierarchical multicast schemes are also implemented. One is the hierarchical DDM multicast named as HDDM. The other is HDDM without dynamic partition, which is named as HDDM-Static. For fairness of comparison, AODV (Perkins and Royer, 1999) is used as the underlying unicast protocol for both hierarchical DDM protocols. In both HDDM protocols, the minimum and maximum allowed size of each sub-group are 9 and 20,

respectively. For performance references, we also run simulation with a mesh based protocol, ODMRP (Lee *et al.*, 1999).

Performance versus mobility: In this study, Fig. 4 presents the performance metrics as functions of pause time. The group size in the simulations is 150. As shown in Fig. 4a, ODMRP and HDDM achieve similar packet delivery ratio for all pause time setups. HDDMStatic 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 down faster than the other two protocols. When pause time is low, more amount of 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. This reduces the forwarding hops at the lower layer multicast. Figure 4b 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, much 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 piggybacked in the packet headers. The high cost of media access in MANET environment favors the in-band signaling style 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 60s pause time. This accounts for the much reduced control traffic. The average delay latency is shown in Fig. 4d. 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-phased 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 is the effect of dynamic partition, which tries to shorten the delivery path at the lower level multicasts.

Vertical scalability issues: In this study, Fig. 5 shows the performance metrics as functions of group size. With fixed pause time as 60 sec, we have one multicast group of size from 20 to 200. Figure 5a shows the result for packet delivery ratio. As group sizes increases, 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. 5b. 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 subroot 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. Figure 5c 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 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. Figure 5d shows the averaged delivery latency and variance variance among the receivers. Compared to ODMRP, HDDM and HDDM-

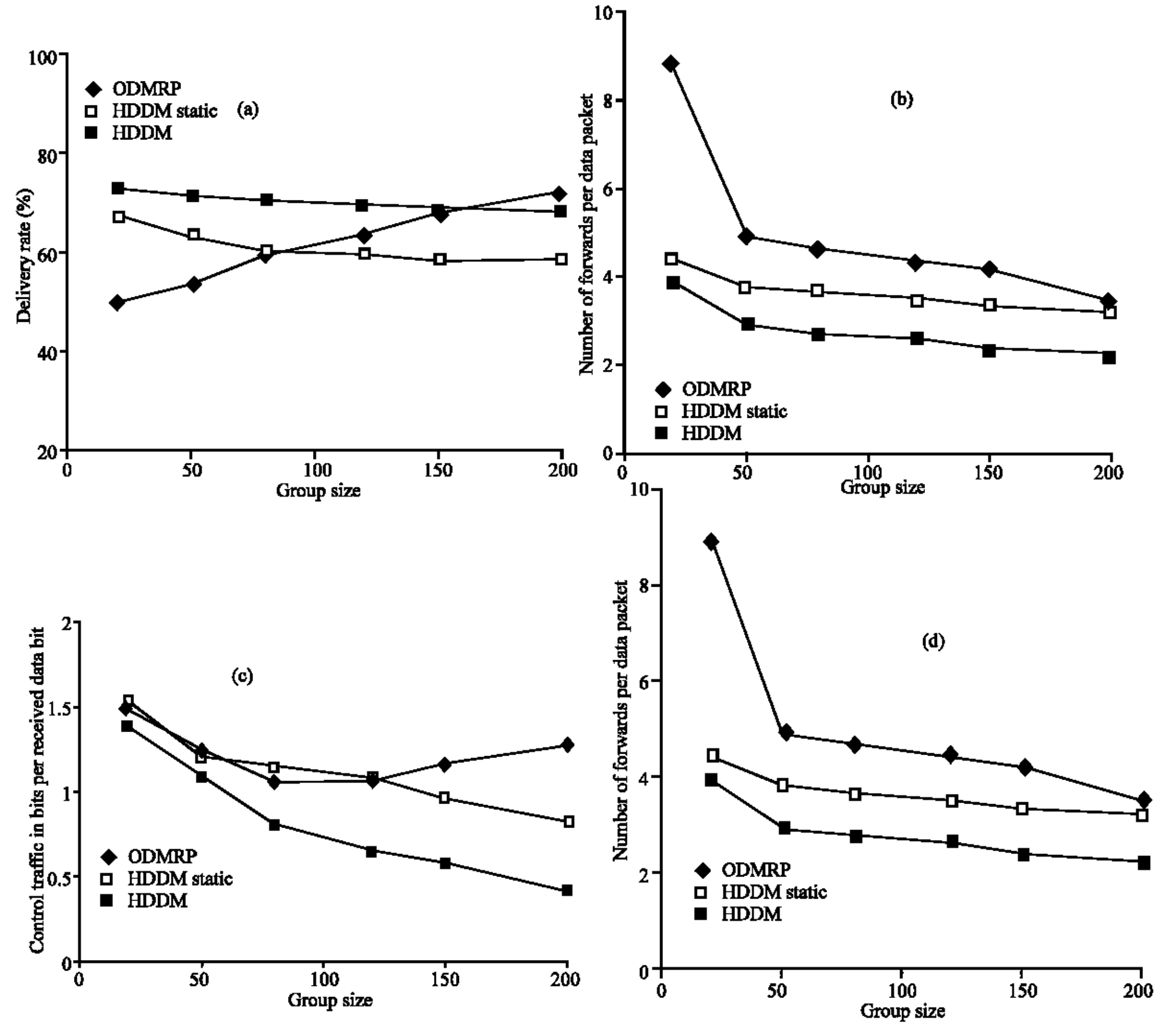

Fig. 5: Performance versus group size. (Pause time is 60s, 1 group, 1 source per group), a): Packet delivery ratio, b): Forwarding efficiency, c): Normalized Bit Overhead, d): Average delivery latency

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.

Horizontal scalability issues: 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, "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.

Figure 6(a) shows the packet delivery ratio and the variance among the groups in the network. As the number of groups increases, performance of ODMRP shows quick drop to less than 10% for 12 groups. With more 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

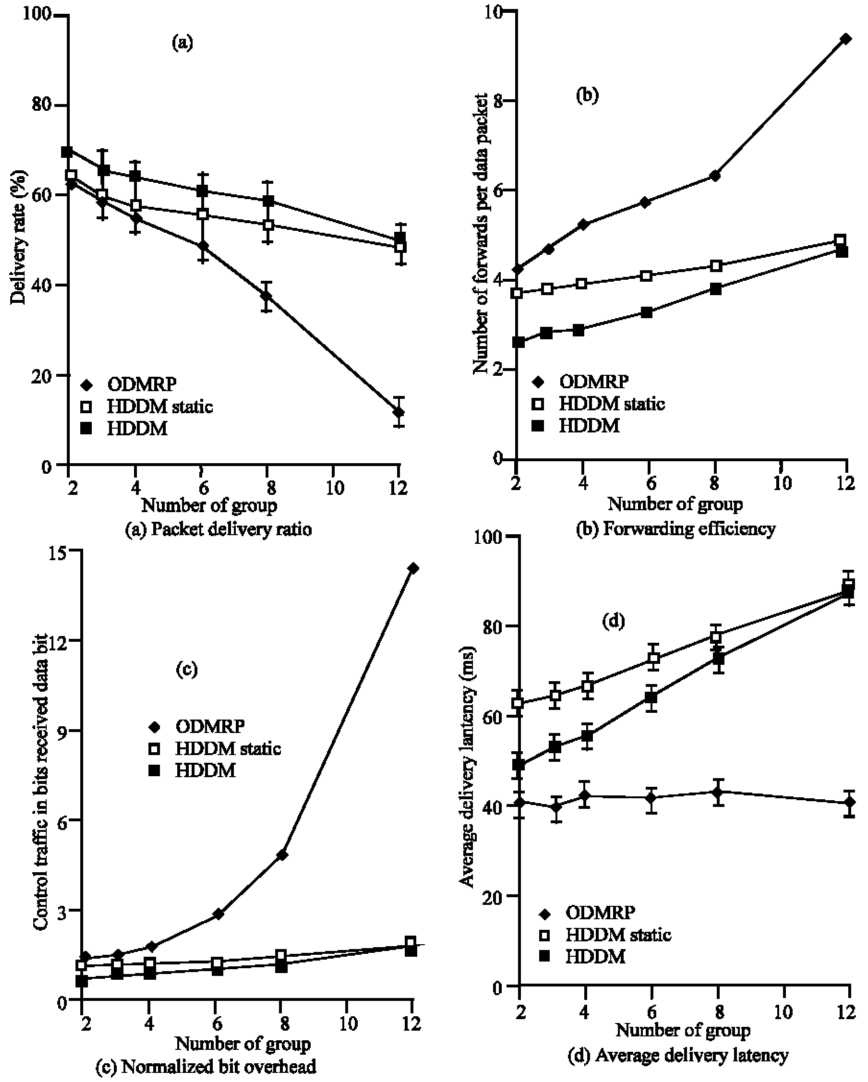

Fig. 6: Performance versus number of groups (Pause time is 60s, 1 source per group)

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. 6(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 study, when the group sizes becomes smaller. Both HDDM protocols present more stable curves.With smaller group, the chance for one broadcast transmission to reach multiple members decreases, thus their curves ascends when the number of groups increases. Figure 6(c) shows the results for relative control bit overhead. 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 is that for the sub-group multicast level, the number of sub-groups does not change much with the different scenarios. Figure-6(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 subgroup 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 can derive the following conclusions. When there are more multicast groups in the network, ODMRP has quick drop in all performance metrics. Both of the HDDM protocols present very stable behavior in terms of horizontal scalability. When there are more groups, dynamic partitioning becomes less effective.

Related work: A few schemes (Lin and Gerla, 1997; Ramanathan and Steenstrup, 1998) 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) (Ko *et al*., 2003) is a specially tailored multicast algorithm for the kind of 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. The multicast forwarding state at the Internet routers is studied in (Thaler and Handley, 2000). Several hierarchical routing protocols have been proposed for supporting multicasting in the Internet (Thayajaragian and Deering, 1995; Shields and Garcia, 1998, 1997). HDVMRP (Hierarchical Distance Vector Multicast Routing Protocol) (Thayagarajan and Deering, 1995) 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 (Shields and Garcia, 1998) 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 OCBT (Shields and Garcia, 1998). In this approach, the leaf nodes of a higher level multicast tree can each be functioning as the root of a lower-level tree. The 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. In MANET, the links are formed in ad hoc manner and data is transmitted through radio broadcast. 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.

THE RETSINA MULTI-AGENT INFRASTRUCTURE

RETSINA (Thyagarajan and Deering, 1995) (Reusable Task-based System of Intelligent Networked Agents) is a multi-agent infrastructure that was developed for information gathering and integration from web-based sources and decision support tasks. Each agent in RETSINA specializes in a specific class of tasks. When the agents execute tasks or plan for task execution, they organize themselves to avoid processing bottlenecks and form teams to deal with dynamic changes in information, tasks, number of agents and their capabilities.

In RETSINA, the agents are distributed and execute on different machines. The RETSINA architecture is shown in Fig. 7.

Based on models of users, agents and tasks, the agents decide how to decompose tasks and whether to pass them to others, what information is needed at each decision point and when to cooperate with other agents. The agents communicate with each other to delegate tasks, request or provide information, find information sources, filter or integrate information and negotiate to resolve inconsistencies in information and task models. The system consists of three major classes of agents: 3 interface agents, task agents and information agents (Shields and Garcia, 1997).

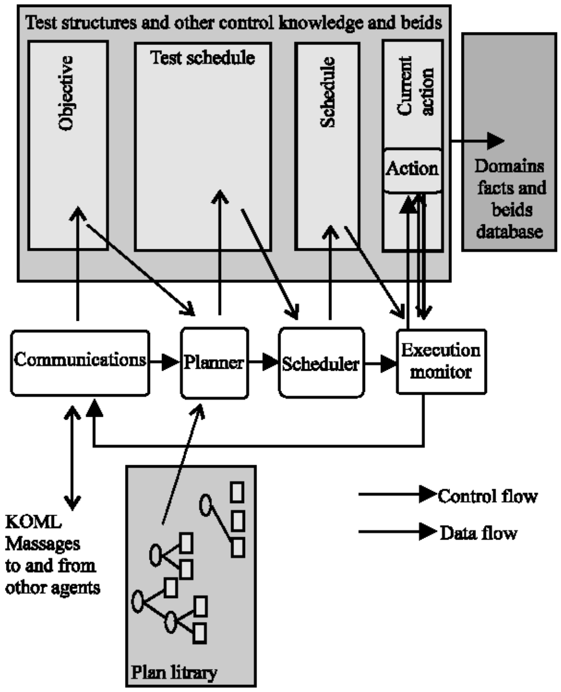

Fig. 7: RETSINA architecture

Interface agents interact with users receiving their specifications and delivering results. They acquire, model and utilize user preferences. The main functions of an interface agent include:

- Collecting relevant information from the user to initiate a task,
- Presenting relevant intermediate and final results,
- Requesting additional information during task execution.

The interface agents hide the underlying structural complexity of the agent system.

For instance, there may be a hybrid of 2 types, such as interface + task agent.

Task agents (Perkins and Royer, 1990) formulate plans and carry them out. They have knowledge of the task domain and which other task agents or information agents are relevant to performing various parts of the task. In addition, task agents have strategies for resolving conflicts and fusing information retrieved by information agents. A task agent

- Receives user delegated task specifications from an interface agent.
- Interprets the specifications and extracts problem solving goals.
- Forms plans to satisfy these goals.
- Identifies information seeking sub goals that are present in its plans.
- Decomposes plans and cooperates with appropriate task agents or information agents for plan execution, monitoring and results composition.

Information agents provide intelligent access to a heterogeneous collection of information sources. They have models of the information resources and strategies for source selection, information access, conflict resolution and information fusion. Information agents can actively monitor information sources.This multiagent infrastructure has the protocols.

CONCLUSION

In this study, 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 overlaydriven 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-weighted 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.

REFERENCES

C. de Morais Cordeiro, H. Gossain and D.P. Agrawal, 2003. Multicast over wireless mobile ad hoc networks: Present and future directions, IEEE Network, Vol. 17.

Camp, T., J. Boleng and V. Davies, 2002. A Survey of Mobility Models for Ad Hoc Network Research, Wireless Communication and Mobile Computing (WCMC)., 2: 483-502.

Chen, K., and K. Nahrstedt, 2002. Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET, Proc. IEEE Infocom, New York.

Chu, Y., S. Rao and H. Zhang, 2000. A Case for End System Multicast, Proc. ACM SIGMETRICS, Santa Clara, CA.

Das, S.K., B.S. Manoj and C.S.R. Murthy, 2002. A Dynamic Core Based Multicast Routing Protocol for Ad hoc Wireless Networks, Proc. ACM MOBIHOC, Lausanne, Switzerland, GloMoSim, http://pcl.cs. ucla.edu/ projects/glomosim/.

Eriksson, H., 1994. MBone: The Multicast Backbone, Commun. ACM., 37: 54-60.

Garcia-Luna-Aceves, J.J. and E.L. Madruga, 1999. The Core-Assisted Mesh Protocol, IEEE J. Select. Areas Commun.,17: 1380-94.

Gerla, M., C.C. Chiang and L. Zhang, 1999. Tree multicast strategies in mobile, multihop wireless networks, ACM Mobile Networks and Applications, 4: 193-207.

Grossglauser, M. and M. Vetterli, 2003. Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks, Proc. IEEE Infocom, San Francisco, CA.

Gui, C. and P. Mohapatra, 2003. Efficient Overlay Multicast for Mobile Ad Hoc Networks, Proc. IEEE WCNC, New Orleans, LA,.

Jaikaeo, C. and C.C. Shen, 2002. Adaptive Backbone-Based Multicast for Ad hoc Networks, Proc. IEEE ICC, New York.

Ji, L. and M.S. Corson, 2001. Differential Destination Multicast -A MANET Multicast Routing Protocol for Small Groups, Proc. IEEE Infocom, Anchorage, Alaska.

Johnson D.B. and D.A. Maltz, 1996. Dynamic Source Routing in Ad-Hoc Wireless Networks, Mobile Computing, T. Imielinski and H. Korth (Eds.), Kluwer, pp:153-81.

Kleinrock, L. and F. Kamoun, 1977. Hierarchical Routing for large networks; performance evaluation and optimization, Computer Networks, 1: 155-174.

Ko, Y.B., S.J. Lee and K.Y. Lee, 2003. A multicast protocol for physically hierarchical ad hoc networks, Proc. IEEE VTC, Jeju, Korea.

Kwon, M. and S. Fahmy, 2000. Topology-Aware Overlay Networks for Group Communication, Proc. ACM NOSSDAV2, Miami, FL.

Lee, S.J. W. Su, J. Hsu, M. Gerla and R. Bagrodia, 2000. A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols, Proc. IEEE Infocom, Tel-Aviv, Israel.

Lee, S.J., M. Gerla and C.C. Chiang, 1999. On Demand Multicast Routing Protocol, Proc. IEEE. WCNC', pp: 1298-1302.

Lin C.R. and M. Gerla, 1997. Adaptive clustering for mobile wireless networks, IEEE. J. Select. Areas Commun.,15: 1265-1275.

Luo, J., P.T. Eugster and J.P. Hubaux, 2003. Route Driven Gossip: Probabilistic Reliable Multicast in Ad Hoc Networks, IEEE Infocom, San Francisco, CA.

Perkins C.E. and E.M. Royer, 1999. Ad-hoc On-Demand Distance Vector Routing, Proc. IEEE. Workshop on Mobile Comput. Sys. Applications, pp: 90-100.

Perkins C.E. and P. Bhagwat, 1994. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, Comp. Commun. Rev., pp: 234-44.

Ramanathan, R. and M. Steenstrup, 1998. Hierarchically-organized, multihop mobile wireless networks for quality-of-service support, ACM/Baltzer Mobile Networks and Applications, 3: 101-119.

Royer, E.M. and C.E. Perkings, 1999. Multicast Operations of the Adhoc On-Demand Distance Vector Routing Protocol, Proc. ACM MOBICOM, Seattle, WA.

Shields, C. and J.J. Garcia-Luna-Aceves, 1997. The Ordered Core Based Tree Protocol, Proc. IEEE Infocom, Kobe, Japan.

Shields, C. and J.J. Garcia-Luna-Aceves, 1998. The HIP Protocol for Hierarchical Multicast Routing, Proc. 17th Annual ACM Symposium on Principles of Distributed Computing, Puerto Vallarta, Mexico.

Sinha, P., R. Sivakumar and V. Bharghavan, 1999. MCEDAR: Multicast Core- Extraction Distributed Ad Hoc Routing, Proc. IEEE WCNC.

Thaler, D. and M. Handley, 2000. On the aggregatability of multicast forwarding state, Proc. IEEE Infocom.

Thyagarajan, A.S. and S.E. Deering, 1995. Hierarchical distance-vector multicast routing for the MBone, Proc. ACM SIGCOMM, Cambrige, Massachusetts.

Xie, J., R.R. Talpade, A. Mccauley and M. Liu, 2002. AMRoute: ad hoc multicast routing protocol, ACM Mobile Networks and Applications, Vol. 7.

Yoon, J., M. Liu and B. Noble, 2003. Random Waypoint Considered Harmful, Proc. IEEE Infocom, San Francisco, CA, BonnMotion Project, http://web. informatik. unibonn. de/ IV/ Mitarbeiter/ dewaal/ BonnMotion/.