Efficient Overlay Multicast for Mobile Ad Hoc Networks

Chao Gui and Prasant Mohapatra
Computer Science Department
University of California, Davis
Davis, CA 95616
Email: {guic,prasant}@cs.ucdavis.edu

Abstract— **Overlay multicast protocol builds a virtual mesh spanning all member nodes of a multicast group. It employs standard unicast routing and forwarding to fulfill multicast functionality. The advantages of this approach are robustness and low overhead. However, efficiency is an issue since the generated multicast trees are normally not optimized in terms of total link cost and data delivery delay. In this paper, we propose an efficient overlay multicast protocol to tackle this problem in MANET environment. The virtual topology gradually adapts to the changes in underlying network topology in a fully distributed manner. A novel *Source-Based Steiner* tree algorithm is proposed for constructing the multicast tree. The multicast tree is progressively adjusted according to the latest local topology information. Simulations are conducted to evaluate the tree quality. The results show that our approach solves the efficiency problem effectively.**

Index Terms—**MANET, Overlay Multicast, Stateless Multicast, Virtual Topology, *Source-Based Steiner* Tree Algorithm**

I. INTRODUCTION AND BACKGROUND

MOBILE Ad Hoc Network (MANET) [1] refer to a form of infrastructureless network connecting mobile devices with wireless communication capability. Each node behaves as a router as well as an end host, so that the connection between any two nodes is a multi-hop path supported by other nodes. For typical applications, MANET is used to support close collaboration among team members. Thus, multicast support is critical and a desirable feature of ad hoc networks.

Multicasting in MANET faces many challenges due to the continuous changes in network topology and limited channel bandwidth. Thus conventional multicast schemes designed for wire-line networks cannot directly apply. Many multicast routing protocols have been proposed for MANET [2], [3], [4], [5], [6], [7]. For these protocols, robustness and high overhead are key problems. They maintain state information at all involved nodes, both member nodes and non-member nodes that act as routers for supporting the multicast session. This widespread maintenance of state information lowers the robustness due to node mobility. If the routing topology involves some fast moving nodes, even though they are not member nodes, the multicast session is hampered. Further, the state information in the involved nodes should be updated when members join or leave the group. This is another burden on all the involved nodes.

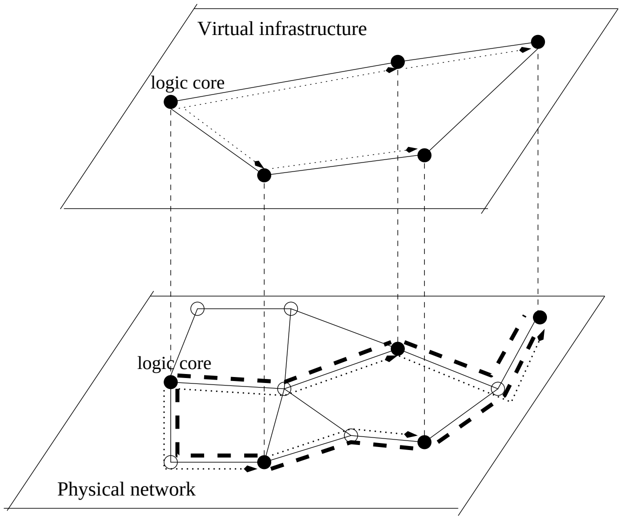


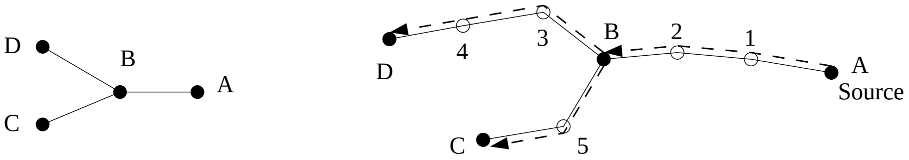
Fig. 1. Concept of virtual topology for overlay multicast.

Overlay multicast [8], [9], [10] is proposed as an alternative approach for providing multicast services in the Internet. A virtual infrastructure is built to form an overlay network on top of the physical Internet. Each link in the virtual infrastructure is a unicast tunnel in the physical network. IP layer implements a 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[11] is an ad hoc multicast protocol that uses the overlay multicast approach. Bidirectional unicast tunnels are used to connect the multicast group members into a virtual mesh. After the mesh creation phase, a shared tree for data delivery purpose is created and maintained within the mesh. One member node is designated as the logical core, which is responsible for initiating the tree creation process periodically. Figure 1 illustrates the concept of virtual mesh and the shared tree built within the mesh.

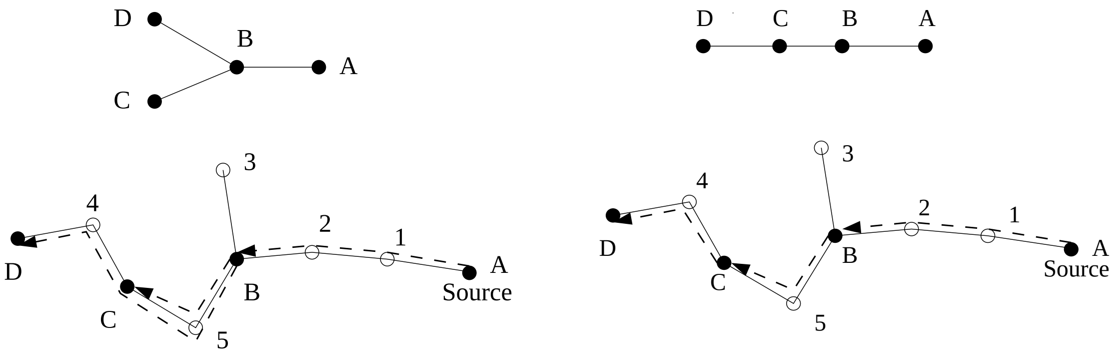
The virtual topology can remain static even though the underlying physical topology is changing. Moreover, it needs no support from the non-member nodes, i.e. all multicast functionality and state information are kept within the group member nodes. This complies to the "stateless" architectural principle for network protocols. Other advantages are simplicity and flexibility. The protocol does not need to track the network mobility since

---

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



a) Initial virtual topology and physical topology



b) Changed physical topology and initial virtual topology

c) Changed physical topology and adapted virtual topology

Fig. 2. Efficiency of overlay multicast.

it is totally handled by the underlying unicast protocols. Unlike some other multicast schemes, AMRoute protocol has no special requirements imposed on the unicast protocol. Thus, it can operate seamlessly on multiple domains that use different unicast routing protocols.

The rest of the paper is organized as follows. In Section II, we describe the efficiency problem of overlay multicast. In Section III, we present our multicast protocol in detail, aiming at solving this efficiency problem. In Section IV, we present the numerical results of simulation studies. The related work are discussed in Section V. Finally, the concluding remarks are presented in Section VI.

II. EFFICIENCY PROBLEM OF OVERLAY MULTICAST

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. Figure 2 illustrates an example case where the static virtual topology could cause excessive redundancy. Figure 2(a) depicts an initial setup of both virtual topology and physical topology. Node A is the source node. The virtual topology can serve as a multicast routing tree for data delivery. The dashed lines in the physical topology are unicast tunnels corresponding to the virtual topology. As the nodes move, unicast routing protocol will form different tunnels for the same virtual topology, which is shown in Figure 2(b). As we can see, 9 physical links are used for the virtual topology. Physical links B-5 and 5-C are redundantly included. Figure 2(c) shows the adapted virtual topology and its corresponding tunnels, which only requires 7 physical links. Initially, nodes C and D have higher hop distance and the virtual link C-D is not included in the virtual topology. However, with the movement of network nodes, the hop distance between C and D reduces significantly. With static virtual topology, the virtual link C-D can never be utilized, resulting in high redundancy. AMRoute[11] periodically rebuilds the shared routing tree in order to account for the node mobility. However, it still gets into the inefficient situation show in Figure 2(b) because the shared tree is always built using the static virtual mesh. To tackle this problem, we propose an overlay multicast scheme that constantly optimizes the quality of generated multicast trees. It eliminates redundant physical links, so that the overall bandwidth consumption of the multicast session is reduced.

III. PROTOCOL DESCRIPTION

Our proposed protocol for overlay multicast is called Progressively Adapted Sub-Tree in Dynamic Mesh(PAST-DM). The virtual mesh topology gradually adapts to the changes of underlying network topology in a fully distributed manner with minimum control cost. The multicast tree for packet delivery is also progressively adjusted according to the current topology. Exploiting the advantages of overlay multicast approach, the join and leave operations can be simple and robust.

A. Dynamic Virtual Mesh by Link State Exchange

A multicast session begins with the construction of a virtual mesh connecting all group members. Each member node starts a neighbor discovery process using the expanded ring search (ERS) technique [12]. Group_REQ message is used for this purpose. The maximum radius of the ring should be limited to a very small value. This is justified by the simulation results. For a group of 20 nodes randomly chosen from a network of 100 nodes, the average hop length of the virtual links on the multicast tree is 3.8. When node I receives a Group_REQ message from node J, it records node J as its neighbor in the virtual mesh, along with the hop distance to reach node J. Node I then sends back a Group_REP message to J, so that node J will record the same. The maximum degree of the virtual topology is controlled. When the number of virtual neighbors of a node reaches the upper limit, the node will stop the neighbor discovery process. For a singular remote member node, it is possible that the radius of the search ring reaches the upper limit while no neighbor has been found yet. In this case, a special flooding technique is needed.

Each member node keeps track of other members in its vicinity. This can be done by an query to its route table maintained by unicast protocol, or by a periodic neighbor discovery operation. Each node records its virtual neighbors as its *virtual link state*. PAST-DM makes each member node maintain the topology map of the virtual mesh. This is done by the link state exchange technique, which is used in the Fisheye State Routing protocol[13] for ad hoc network. At each node, the topology map is represented as a link state table. The entries are the link state information of all group nodes obtained from virtual neighbors. Every node periodically exchanges this link state table with its neighbor nodes only(no flooding). Each entry in a link state table carries a sequence number, and the entry with a higher sequence number will always replace the one with lower sequence number. The link state of a node will eventually be carried to the faraway nodes after several exchanges. Through the link state tables, each node has a local view of the whole virtual topology. To avoid a storm of link state exchanges, each node can make the interval between two consecutive exchanges as the period plus a small random offset. This will give a more stable overall network performance.

B. Data Delivery Tree

Compared to shared tree method, source-based tree approach is more efficient for data delivery. In PAST-DM protocol, each source constructs its own data delivery tree based on its local link state table. No extra overhead of control message is needed. This is the key difference between PAST-DM method and other source-based tree protocols. We developed a novel *Source-Based Steiner* tree algorithm for the tree computation. To minimize the total cost of multicast tree, the source needs to construct a *Steiner* tree for the virtual mesh. As the local view of the virtual mesh at the source node is based on its link state table, the topology information close to the source is more up-to-date and accurate. It is progressively less accurate as hop distance increases. Thus, if there is a tie between two virtual links with the same cost during the tree construction, the one that is closer to the source node is favored. Specifically, the virtual links adjacent to the source should always be included in the tree since they are from the most up-to-date link state information. To address this property, we propose a *Source-Based Steiner tree* algorithm as discussed next.

Let $ds(n)$ denote the hop distance from source node s to node n (regardless of the costs on the links). For a virtual link (n1,n2), its hop distance to source node is defined as follows.

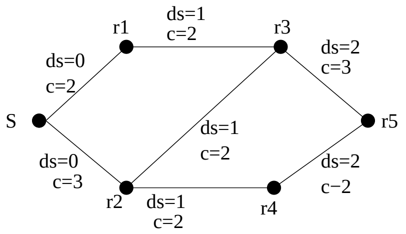
$$ds(n_1, n_2) = min[ds(n_1), ds(n_2)].$$

Let $c(n_1, n_2)$ be the cost of the virtual link (n_1, n_2). We define the *"adapted cost"* of the link as its cost multiplied by its distance to the source.

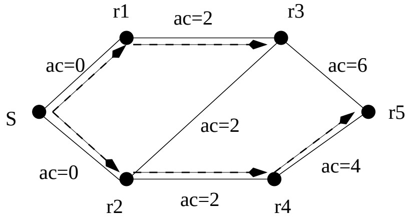
$$ac(n_1, n_2) = ds(n_1, n_2) \cdot c(n_1, n_2).$$

Thus, for any link that is adjacent to the source node, its distance value and *adapted cost* should both be 0. With each link using its *adapted cost* value, we apply the following heuristic [14] for computing the *Steiner* tree. With the zero-cost links, the initial tree can have the source node as root and all neighbors as its first-level children. The partially constructed tree increments toward the *Steiner* tree by including the nearest receiver to it, together with the shortest path connecting them.

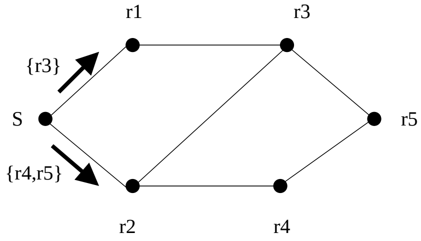
By applying the *Source-Based Steiner tree* algorithm, the source makes all its neighbors as its children in the multicast tree and divides the remaining nodes into subgroups. Each subgroup forms a subtree rooted at one of the first-level children. The source node does not need to compute the whole multicast tree. It puts each subgroup into a packet header, combines the header with a copy of the data packet, and unicasts the packet to the corresponding child. Each child is then responsible of further delivering the data packet to all nodes in its subgroup. It does so by repeating the *Source-Based Steiner tree* algorithm. Eventually, the subgroup will become empty, then the process stops. Or, if the subgroup has only one node, the data packet is directly unicast to the final receiver.



(a) Distance and cost of each link



(b)Adapted cost of each link and Source−Based Steiner tree



(c) Receiver lists in the header of the packets sent from S to its children

Fig. 3. Example of Tree Construction.

Figure 3 shows an example. At the source node S, its receiver list is $\{r1 \cdots r5\}$. Figure 3(a) shows its local view of virtual topology. The *Source-Based Steiner* tree using adapted costs is shown in Figure 3(b). S generates two smaller lists: $\{$r3$\}$ and $\{$r4, r5$\}$. They are included into the header of the packets sent to r1 and r2 as shown in Figure 3(c).

To reduce the computational complexity, the source node can cache the recent tree construction result for the incoming data packets. This cached result can be kept valid until the local link state table is updated during the next incoming link state exchange.

C. Join and Leave

Contrary to the complicated join and leave process with conventional "stateful" multicast, overlay multicast supports dynamic membership in a simple and robust manner. When a node intends to join the multicast group, it starts with a normal neighbor discovery process described in Section III-A. As multiple groups may co-exist in the network, it needs a group address in its Group_REQ process. As the member nodes of the intended group respond with Group_REP messages, it can collect its own virtual neighbors and set up its own link state. As the responding group nodes also include the newcomer as their neighbor, they will start to exchange link state tables with the new member. In this way, the join of this new member is eventually recognized by far away nodes. The new member will gradually build its own view of the virtual mesh as well.

A new member can start receiving data packets right after it is recognized by its virtual neighbors, which is well before the faraway source node recognizes this new member. This is achieved by an additional data forward function on each child node on the multicast tree. When a child node has forwarded the data packet to all nodes in its subgroup, it checks if all its neighboring receivers are included in the subgroup. An additional unicast is needed for each missing neighboring receiver. As a new member may have multiple virtual neighbors, it will receive multiple copies of the same data packet. It should discard the duplicate ones. However, once the source recognizes the new member and puts it into the delivery list, it will no longer receive any more duplicate packets.

To leave the group, a member node needs to unicast a Group_LV message to its current virtual neighbors. The link state exchange between it and its neighbors will stop. The neighbors will not deliver the data packet to it even though it may still appear in the subgroup node list for a while.

IV. PERFORMANCE EVALUATION

A. Simulation Configuration

Our simulation network has 100 mobile nodes randomly roaming within a 2000m × 750m free space. The radio transmission range of each node is 250 meters. In this geometric setup, the probability of partition in network topology is relatively small. The average hop distance is 4.4. The longest hop distance is 11. Each simulation run lasts for 500 simulation seconds. The movement of each node follows the random waypoint model [15]. Each node remain static at the initial point for 10 seconds, then selects a destination location randomly within the roaming area and moves straight toward the destination with a constant speed, which is uniformly distributed over [0, 20] m/s. After arrival, the node pauses at the location for 10 seconds then moves to another destination, and so on.

For the multicast sessions, we choose the group size to be 5, 10, 20, 30 and 40. With a group size of 40, we see that the header size of each packet under PAST-DM protocol is comparable to that of DSR protocol. This is because PAST-DM lets the source node divide the receiver group evenly among its virtual neighbors. Though the size of the first level sub-groups is much greater than the average hop distance, the sub-groups in lower levels shrink rapidly as packets reach closer to receivers.

B. Performance Measurements

In this simulation, we measure the following metrics: (a) (relative) tree cost; and (b) (relative) maximum delay. The cost of a data delivery tree is the sum of physical hop lengths of all virtual links of the tree. Maximum delay is the number of physical hops along the longest path from the source to any of the receivers on the tree. In order to compare the efficiency of overlay multicast method, we compute the optimal tree cost and optimal maximum delay at each simulation step. The optimal tree cost is computed as the overall hops of a *Steiner* tree built on the physical topology for the same group of nodes. The optimal maximum delay is the maximum delay of the *Shortest Path Tree* built on the physical topology with the same source node. By definition, the optimal values are the best we can have based on existing physical network topology. The relative value of the metrics are the ratio of measured value over the corresponding optimal value, which represent quality and efficiency of the generated multicast trees.

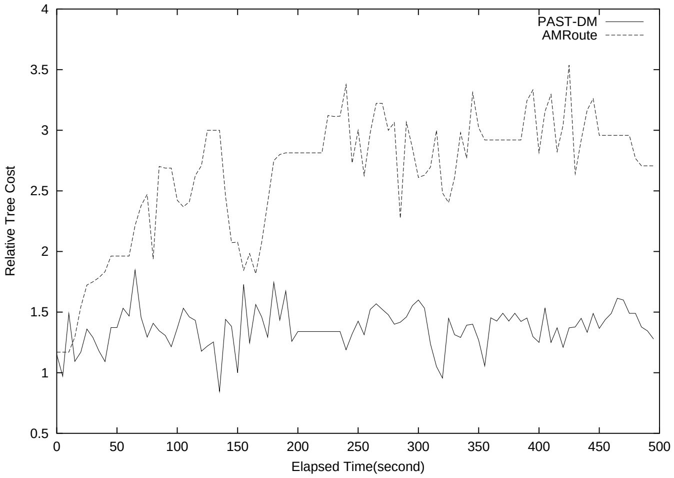
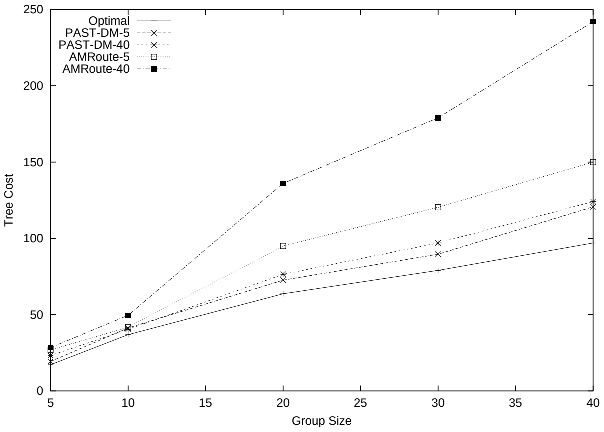


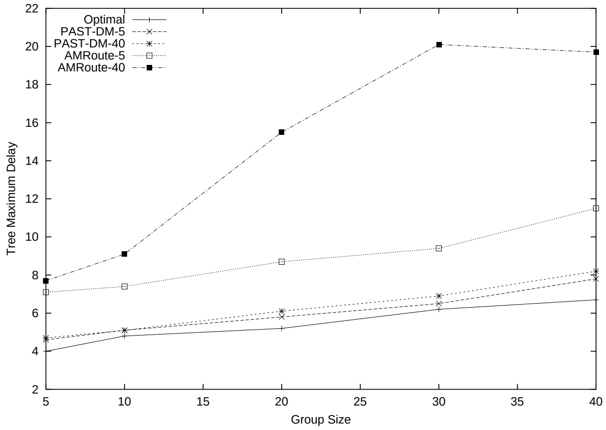
Fig. 4. Relative Tree Cost (Group size is 40).

C. Time-line of relative tree cost

Figure 4 shows the time-line of relative tree cost for a multicast group of size 40 under AMRoute and PAST-DM. The period of AMRoute tree creation process is 40 simulation seconds. The period of virtual link state exchange for PAST-DM is also set to 40 simulation seconds. As shown in the graph, PAST-DM builds more efficient multicast trees than AMRoute nearly at all times. During the beginning 20 seconds, the time lines are intermingled together. This is because the virtual mesh is just built up by AMRoute. As the mobile nodes have not moved far from their initial places, the virtual mesh still reflects the underlying physical topology. As time elapses, the relative tree cost of AMRoute increases rapidly, then it oscillates heavily at high



(a) Tree Cost.



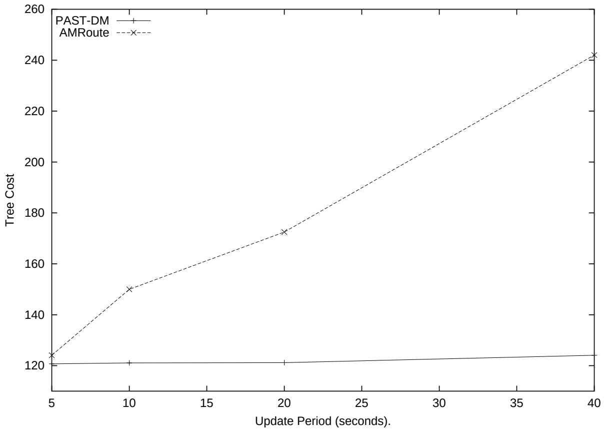
(b) Maximum Delay.

Fig. 5. Comparison of multicast tree quality with different group size.

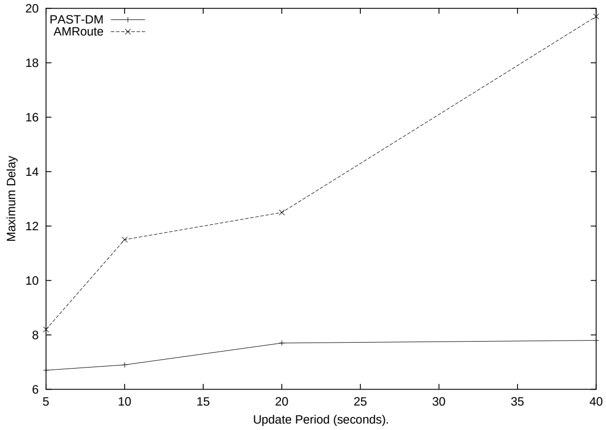
values. The time-line of PAST-DM oscillates less heavily at lower value, which means a more stable network performance.

D. *Tree quality versus group size*

Figures 5(a) and (b) show the average tree cost and maximum delay of the multicast trees versus the group size from 5 up to 40. At each simulation step of PAST-DM protocol, a data delivery tree is constructed with the source randomly chosen from the group members. The average tree cost and maximum delay is recorded for all step-wise multicast trees. As shown in the graph, PAST-DM yields close to optimal trees for all group sizes. The cost of the trees built by AMRoute increases faster than the optimal value as the group size increases. The same is true for the maximum delay. Thus the multicast trees become less efficient as the multicast group grows. We can also calculate the average hop length of virtual links in the multicast tree. For example, for any delivery tree spanning a group of 20 nodes needs just 19 virtual links. The total costs of trees by AMRoute-5 and PAST-DM-5 are 95.9 and 72.6 respectively. So the average length of virtual links in both trees are 5.0 and 3.8.



(a) Tree Cost.



(b) Maximum Delay.

Fig. 6. Comparison of multicast tree quality with different update period.

E. *Tree quality versus update period*

Figures 6(a) and (b) show the change of multicast tree quality with different update periods. The update period for AMRoute and PAST-DM refers to the tree re-creation period and the virtual link state exchange period, respectively. As higher update period results in less protocol overhead, we are looking for a high update period that does not result in much worse tree quality. As shown in the figure, tree cost and maximum delay of AMRoute increases rapidly with update period. This means it needs frequent re-creation of multicast tree to keep up with the change of physical hop lengths of the virtual links. PAST-DM yields a stable tree quality as update period increases. It is less sensitive to the frequency of link state exchange. Thus, with PAST-DM, we can make the link state exchange among group members to be less frequent in order to reduce the overhead of the protocol.

V. RELATED WORK

Besides overlay multicast, our work is also closely related to stateless multicast. This approach is similar to overlay multicast in the sense that it also employs standard unicast routing and forwarding. The difference is that there is no explicit concept of overlay virtual topology. SGM[16] was first proposed for small group multicasting on the Internet. The source node first performs a route table look up to determine the next hop for each receivers in its group. It then partitions the group based on the next hops. Each subgroup is encoded into a SGM packet header and the packet is unicast to the corresponding next hop. The receiving next hop will further partition its subgroup and forward the data to it next hops. SGM is considered as a scalable solution to support large number of small multicast groups on the Internet. DDM[17] is a stateless multicast protocol proposed for MANET. In DDM, receivers can be listed in the DDM packet header in a differential manner, which means it only includes the difference with respect to the receiver list in the last packet. Each node in the forwarding paths remembers the subset it has been forwarded to last time, together with the corresponding next hop information. By caching routing decisions in the intermediate nodes, the source does not need to list the group members in future packets. In making routing decisions, intermediate nodes need to query the unicast route table.

With GPS device, each mobile node is aware of its location within the network area. Location Guided Tree (LGT) construction scheme[18] builds overlay multicast tree using geometric distance between member nodes as the heuristic of link costs. Two tree construction algorithms are proposed: greedy k-ary tree construction (LGK) and *Steiner* tree construction (LGS). With LGK, the source nodes selects k nearest neighbors as its children, and partitions the remaining nodes according their distance to the children nodes. LGS constructs the *Steiner* tree using link costs as their geometric lengths. Each children node is then responsible for packet delivery to its own subgroup using the same algorithm.

Our work is inspired by the stateless multicast approaches. The difference is that our approach uses virtual mesh so that it has the advantages of overlay multicast, such as simple join and leave processes. In DDM, when a node joins a multicast group with multiple sources, it should unicast the join messages to all the sources. The same is true for leave operation.

VI. CONCLUSION

In this paper, we present an overlay multicast protocol aiming at solving the efficiency problem of overlay multicast approach. To achieve this, we propose dynamic virtual mesh that adapts itself to the mobility of network nodes. Virtual links with long physical hop lengths are replaced by short links. A novel tree construction algorithm is proposed that fully utilizes the latest local topology information. Simulation studies have shown that the yielded multicast tree is close to optimal in terms of total hop cost with a stable quality. Thus stable and efficient multicast performance is observed. The control overhead can be reduced to a low level since the tree quality is only moderately hampered when the periodic update behaviors are conducted less frequently.

REFERENCES

[1] M. Gerla, and J.T. Tsai, “Multicluster, mobile, multimedia radio network,” ACM-Blatzer Wireless Networks 1(1995) 255-65.

[2] S.J. Lee, M. Gerla, and C.-C. Chiang, “On Demand Multicast Routing Protocol,” In Proceedings of IEEE WCNC’99, pp. 1298-1302, September 1999.

[3] E. M. Royer, and C. E. Perkings, “Multicast Operations of the Ad-hoc On-Demand Distance Vector Routing Protocol,” In Proceedings of ACM/MOBICOM, August 1999.

[4] L. Ji, and M. S. Corson, “A Lightweight Adaptive Multicast Algorithm,” In Proceedings of IEEE/GLOBECOM’98, November 1998.

[5] J. J. Garcia-Luna-Aceves, and E. L. Madruga, “The Core-Assisted Mesh Protocol,” IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1380-94, August 1999.

[6] C.-C. Chiang, M. Gerla, and L. Zhang, “Forwarding Group Multicasting Protocol for Multihop, Mobile Wireless Networks,” ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, vol. 1, no. 2, pp. 187-96, 1998.

[7] S.K. Das, B.S. Manoj, and C.S.R. Murthy, “A Dynamic Core Based Multicast Routing Protocol for Ad hoc Wireless Networks”, In Proceedings of ACM/MOBIHOC, June 2002.

[8] H. Eriksson, “MBone: The Multicast Backbone,” Communications of the ACM, Aug. 1994, Vol. 37, pp.54-60.

[9] Y. Chu, S. Rao, and H. Zhang, “A Case for End System Multicast,” In Proceedings of ACM/Sigmetrics, June 2000.

[10] M. Kwon and S. Fahmy, “Topology-Aware Overlay Networks for Group Communication,” In Proceedings of ACM NOSSDAV, May, 2002.

[11] E. Bommaiah, M. Liu, A. MvAuley, and R. Talpade, “AMRoute: Ad hoc Multicast Routing Protocol,” Internet Draft, draft-manet-amroute-00.txt, (Work in Progress).

[12] C.E. Perkins, E.M. Royer, and S.R. Das, “Ad Hoc On Demand Distance Vector (AODV) Routing,” Internet Draft, draft-ietf-manet-aodv-10.txt, March 2002 (Work in Progress).

[13] G. Pei, M. Gerla and T.-W. Chen, “Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks,” In Proceedings of IEEE/ICC’00, June 2000.

[14] H. Takahashi, and A. Matsuyama, “An approximate solution for the Steiner problem in graphs,” Mathmatica Japonica, 1980, pp.573-77.

[15] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pp. 153-81, Kluwer Academic Publishers, 1996.

[16] R. Boivie, N. Feldman and C. Metz, “Small Group Multicast: A New Solution for Multicsting on the Internet,” IEEE Internet Computing, May.June 2000.

[17] L. Ji and M. S. Corson, “Differential Destination Multicast - A MANET Multicast Routing Protocol for Small Groups,” In Proceedings of IEEE/Infocom’01, April, 2001.

[18] K. Chen and K. Nahrstedt, “Effective Location - Guided Tree Construction Algorithm for Small Group Multicast in MANET.” In Proceedings.of IEEE/Infocom’02, May, 2002.