A Multicast Routing Algorithm Based on Mobile Multicast Agents in Ad-Hoc Networks

PAPER *Special Issue on Internet Technology*

A Multicast Routing Algorithm Based on Mobile Multicast Agents in Ad-Hoc Networks

Xin WANG[†], **Fei LI**[††], *Nonmembers*, **Susumu ISHIHARA**[†††], *and* **Tadanori MIZUNO**[†††], *Regular Members*

SUMMARY In this paper we describe a multicast routing algorithm, which builds upon mobile multicast agents of an ad-hoc network. Mobile multicast agents (MMAs) form a virtual backbone of an ad-hoc network and they provide multicast tree discovery, multicast tree maintenance and datagram delivery. First, we construct a cluster-spine hierarchy structure for an ad-hoc network. Second, we propose a multicast routing algorithm, which is inspired by Ad-hoc On-Demand Distance Vector (AODV) routing protocol. The results show that the MMA multicast algorithm can simplify the multicast tree discovery, reduce control overhead of the network, and increase the total network throughput, in comparison with general AODV multicast operation. We also overcome the deficiency of CBRP multicast routing, which places much burden on cluster heads.

key words: *multicast routing, mobile multicast agents, virtual backbone, ad-hoc networks*

1. Introduction

IP Multicasting can get efficient use of bandwidth and other network resources by sending a single copy to multiple destinations. Regardless of the network environment, multicast communication is a very useful and efficient means of supporting group-oriented applications. In common Internetworking like Multicast Backbone (MBONE) [1], there are multicast gateways to manage multicast operation by Internet Group Management Protocol (IGMP). The multicast gateways can propagate member-ship information and arrange routing information. In Mobile IP, every mobile host (MH) has its home agent (HA) and foreign agent (FA) to help it multicast datagrams and make location management. In infrastructured mobile networks, a base station (BS) can function as a multicast gateway and multicast membership manager.

However, an ad-hoc network is a multi-hop wireless network in which mobile hosts (MHs) communicate without the support of a wired backbone, HA/FA or BS for routing messages and location management. It has no fixed infrastructure. In ad-hoc networks, the network topology changes frequently. Hence the control packets are very large and the data overhead is very high. A compromise between optimal route computation and rapid topology change is of importance.

In this paper we propose a multicast routing algorithm based upon Mobile Multicast Agents (MMAs) of ad-hoc networks. We construct a cluster-spine hierarchy structure, maintain topology change and perform an extension of Ad-hoc On-Demand Distance Vector Routing (AODV) in multicast routing by use of the spine.

The remainder of this paper is organized as follows. Section 2 outlines the current ad hoc multicast routing protocols and their existing problems. Sections 3 and 4 give a detailed description of our proposed multicast routing algorithm based on MMAs. Section 5 gives some analysis results and evaluations. Section 6 summarizes our paper and presents our conclusions.

2. Background and Related Works

Haas utilizes location databases in ad-hoc networks, which form a virtual backbone [2]. The virtual backbone is dynamically distributed among the network nodes. However those databases serve only as containers for location storage and retrieval. They cannot perform route computation. P. Krishna introduces the conception of cluster-based routing in ad hoc networks [3], however in his proposal every mobile host must retain routing tables and cluster membership lists. Cluster Based Routing Protocol (CBRP) [4] is another source routing algorithm for ad hoc networks, which utilizes cluster heads as routers. It efficiently minimizes the flooding traffic during route discovery and speeds up this process as well. However we consider that it places too heavy burden on cluster heads when cluster heads are used to be not only routers but also cluster membership managers.

Core-Assisted Mesh Protocol (CAMP) [5] supports multicasting by creating a shared mesh structure. It shows better performance than general tree protocols, but with mobility, excessive control overhead causes congestion and collisions. And also it is not an on-demand protocol. On Demand Multicast Routing Protocol (ODMRP) is an on-demand ad-hoc multicast routing protocol based on a multicast mesh too [6]. In ODMRP, a multicast mesh is used to forward multi-

Manuscript received January 18, 2001.

[†]The author is with the Graduate School of Science and Technology, Shizuoka University, Hamamatsu-shi, 432-8561 Japan.

[††]The author is with the Computer Laboratory, Mailbox 25, Sanyuan, 713800 China.

[†††]The authors are with the Faculty of Information, Shizuoka University, Hamamatsu-shi, 432-8561 Japan.

cast packets via flooding. ODMRP relies on frequent network-wide flooding, which may lead to a scalability problem when the number of source nodes is large. Also the control packet overhead becomes very heavy when the multicast group is small in comparison with the entire network.

The spine structure can also act as a virtual backbone of ad hoc networks. Yet in [7]–[9] spine is just in charge of computing and storing routing information, not delivering datagram (if needed, can only be as a transient selection). And also at non-spine nodes, themselves store routing tables. When a sender host wants to send multicast datagram, it should request route information from its corresponding spine node. Thus surely an optimal route can be got. However as we know, routing optimality is of secondary importance, especially in an ad hoc network with high mobility. The most important thing is to find a multicast tree to the multicast group members as quickly as possible and maintain connectivity between MHs in a fast changing topology. So we make a tradeoff between [4] and [7]. In our proposal, in a cluster we use spine nodes as a mobile multicast agent (MMA), which can manage multicast tree discovery and management all together. Thus the capability of one MMA will not be eaten up. We also use MMAs as the relay hosts (routers), so the multicast tree will be composed of a sender host, MMAs and multicast group members. Route information is only stored in MMAs. Thus we can reduce the time to find the multicast tree and the time for a sender host to get routing information. We extend AODV to datagram multicast in spine-based ad-hoc networks.

AODV [10], [11] is an on-demand routing protocol, which builds routes between nodes only as desired by source nodes and maintains routes as long as necessary. It builds routes using a route request and route reply query cycle. Routing tables are used to store routing information. Sequence numbers are used in each relay host to determine freshness of routing information and to prevent routing loops.

Multicast operation of AODV [12], [13] is the extension of AODV unicast routing. It builds multicast routing trees to connect multicast group members. In AODV multicast tree discovery, all mobile nodes are flooded with Route Request Packets (RREQ) so that there are too much flooding traffic totally, especially in dense mode. In CBRP multicast tree discovery, only cluster heads are flooded with RREQ, it places too much burden on a cluster head since it must store all routing tables correspondent to mobile nodes in a cluster. In our proposal we shrink all the network nodes only to MMAs while computing the multicast routing tree by using AODV algorithm. As a compromise, in our proposed multicast tree discovery, only MMAs are flooded with RREQ packets. We minimize the flooding traffic during multicast tree discovery phase and also reduce the burden of a cluster head.

3. Network Architecture

3.1 Overview

An ad-hoc network can be divided into several mobile cluster networks. A cluster is composed of a subset of nodes, which can communicate directly with a cluster head and with each other (possibly). In an ad-hoc network with several mobile clusters, a spine is created by the collection of MMAs, which includes cluster heads and intermediate mobile routers. A spine is a self-organizing network structure similar to ones used in some previous packet radio protocols. In our spine structure MMAs are in charge of the multicast tree discovery, multicast tree maintenance and forwarding datagram. Non-spine nodes include all mobile nodes in the ad-hoc network but not in the spine. Multicast agents are spine nodes in our analysis.

3.2 Network Model

The ad-hoc network in consideration here is modeled as an undirected graph $G(V, E)$ where V is the set of all nodes and E is the set of all links (i, j) where $i, j \in N$. Each link signifies that two hosts are within transmission range of each other. Let S_i be the set of all nodes that can be reached by node i. We assume every link is bi-directional so that link (i, j) exists if and only if $j \in S_i$. The topology of G is the set of nodes and edges. Hence, a node movement can change the topology and this means a change in the network that results in a change in either V or E.

3.3 Cluster Partition

To partition an ad hoc network into clusters we use the following method:

1. When a node comes up, it enters the undecided state, starts a timer and broadcasts a *Hello* message.
2. When a cluster head gets this *Hello* message it responds with a triggered *Hello* message immediately.
3. If the undecided node gets this message within a time threshold it sets itself to member state.
4. If the undecided node times out, it declares itself the cluster head.

After each node performs the above steps, an ad hoc network is partitioned into clusters successfully. A cluster head will have complete knowledge about group membership in the cluster during a time once the topology within the cluster stabilizes. A cluster head will send a cluster query message every a certain period to cluster members and require their reply.

3.4 Spine Construction

The spine structure functions as a virtual backbone for

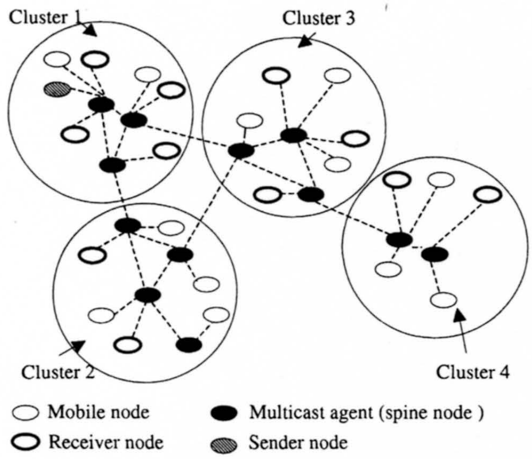

Fig. 1 A spine-based ad-hoc network with 4 clusters.

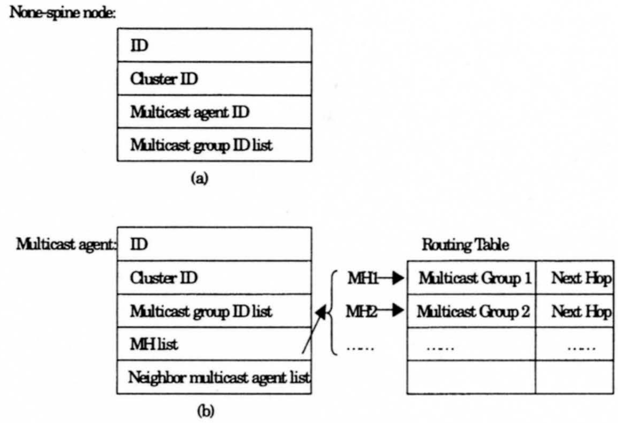

Fig. 2 Group information table.

the ad-hoc network. The uniqueness of the spine structure lies in the fact that enables storage of network state in very few nodes (in MMAs) while still minimizing the access overhead for this information. A cluster head will be selected as a MMA naturally. In Fig. 1 a MMA in the center of a cluster is the cluster head. Spine nodes function as MMAs of their dominating mobile nodes in the ad-hoc network.

Spine construction could be based on an approximation to a minimum connected dominating set (MCDS) [7]. In our proposal an entire ad hoc network has a single spine that is composed of spine nodes in each cluster and intermediate nodes connecting them. Spine construction follows the below steps:

1. Find an approximation to MCDS, S hereafter.
2. Form connected MCDS C, which consists of S and intermediate nodes connecting nodes in S.
3. We get a spine of an ad hoc network by uniting C in all the clusters.

Figure 1 shows an example of a spine-based ad-hoc network with 4 clusters. Dashed lines in Fig. 1 show a link between a MMA and its dominating nodes or a link between a MMA and its neighboring MMAs. In this system MMAs form a spine, which functions as a multicast backbone during the multicast operation. A MMA will send a query message every a certain period to its dominated nodes and require their reply. This query message includes how many nodes it can still dominate. It is possible that another MMA will manage mobile nodes within the transmission range of a MMA. We did so in order to keep balance of the number of mobile nodes that every MMA manages.

4. Multicast Routing Algorithm Based on MMAs

Here we propose a multicast routing algorithm supported by MMAs. It is an extension of AODV routing protocol. Our work focuses on its multicast extension. The two major phases of the algorithm are multicast tree discovery and multicast tree maintenance.

4.1 Data Structure

Figures 2(a) and (b) illustrate the group information table for a non-spine node and a MMA, respectively. Each non-spine node has information of unique ID, multicast group ID, cluster ID and its MMA ID. Each MMA maintains its unique ID, multicast group ID, cluster ID, a list of non-spine nodes that it dominates and a list of its neighboring MMAs. A MMA also keeps a relative routing table for its dominating nodes correspondent to every multicast group.

4.2 Multicast Tree Discovery

In our multicast routing algorithm, unlike the common meaning, a multicast group is only composed of MMAs that dominate some multicast members. So multicast routing tree discovery procedure has nothing to do with general multicast group members when they are non-spine nodes. In other words, multicast tree discovery only concerns those MMAs, which are dominators of corresponding multicast group members. Consequently we can simplify the multicast tree discovery and reduce the overhead of multicast tree computation. Also datagram delivery turns easier since we just need forward datagram to MMAs along the multicast tree and then they forward packets to their dominating mobile nodes.

When a mobile node wishes to join a multicast group, it sends a request packet to its MMA. Then its MMA will also join this group if it does not join that group before. When a sender node wants to send a packet to the multicast group members, such procedure must be done step by step like below:

1. Sender node delivers a route request (SREQ) packet to its MMA. This SREQ packet includes the sender node's ID and the desired multicast group number.

2. After receiving SREQ packet, the MMA looks up its route table to determine whether it already contains a multicast routing tree to the desired multicast group. If it already contains a valid multicast routing tree table to the multicast group that the sender host hopes, it will use this multicast routing tree to send packets from the sender node. In this case it sends back a route reply (SREP) packet to the sender node. After receiving SREP, the sender node delivers packets to its MMA and datagram multicasting begins. Otherwise, execute Step 3.

3. The MMA initiates the route discovery process by broadcasting a route request (RREQ) packet to the spine. The RREQ contains the ID of the MMA, the desired multicast group ID and the destination sequence number, which is set to the last known sequence number for that multicast group.

4. Each intermediate MMA checks whether it knows of a routing tree in its routing table to the desired multicast group. If it does, it places the route record from this RREQ packet into the route reply (RREP) packet and sends the RREP packet to the MMA of the sender node. Thus route discovery procedure is over. Otherwise, execute Step5.

5. If it does not know any route to some members of the desired multicast group, it appends its ID to the route record of the RREQ packet and broadcasts the RREQ packet to its neighboring MMAs. Otherwise, execute Step 6.

6. In this case the intermediate MMA knows some routes to other MMAs of the desired multicast group, and then it adds these MMAs' ID to the RREQ packet to signify that routes to these members have been found and also adds its own ID to the RREQ packet. Then it broadcasts the RREQ packet. On the other hand, it also appends its cached routes (to the desired multicast members) to the RREP packet and sends it back to the MMA of the sender node.

7. Repeat Step 4, 5 and 6 until the multicast routing tree is found or the RREQ packet reaches all the MMAs.

8. The MMA of the sender node collects all arrival RREP packets and forms the desired multicast routing tree.

After the multicast routing tree discovery procedure, datagram can be easily delivered to intermediate MMAs along the multicast routing tree. Then an intermediate MMA sends datagram to multicast members within its transmission range and also forwards the datagram to its neighboring MMA.

A RREQ packet has the following fields:
$\langle J\_flag, R\_flag, Broadcast\_ID, Source\_Addr, Source\_Seq\#, Dest\_Addr, Dest\_Seq\#, MMA\_ID, Hop\_Cnt \rangle$.

The J_flag and R_flag mean join and repair flags, respectively. The J_flag is set when a MMA wants to join a multicast group. The R_flag is set when a MMA wants to initiate a repair to connect two previously disconnected portions of the multicast tree. Each MMA in the network is responsible for maintaining two separate counters: a sequence number and a broadcast ID. The broadcast ID, together with the source MMA's ID, uniquely identifies each RREQ. The sequence number is increased when the MMA acquires new neighbor information, and the broadcast ID is incremented for each RREQ the MMA initiates. The MMA requesting the route places its ID, current sequence number and broadcast ID in the $Source\_Addr$, $Source\_Seq\#$ and $Broadcast\_ID$, respectively. The $Dest\_Addr$ of the RREQ is set to the multicast group ID and the $Dest\_Seq\#$ of the RREQ is set to the last known sequence number of that group. The MMA_ID is valid only when J_flag is set. It is the ID of a MMA that is selected by a coming node. The Hop_Cnt of the RREQ is initialized to zero and is incremented by each MMA that receives it.

A RREP packet has the following fields:
$\langle R\_flag, U\_flag, Dest\_Addr, Dest\_Seq\#, Hop\_Cnt, Lifetime \rangle$.

The R_flag is a repair flag, which is set when a MMA is responding to a repair request to connect two previously disconnected portions of the multicast tree. The U_flag is an update flag, which is set when a multicast tree member has repaired a broken tree link. The $Dest\_Addr$ field is set to the destination address specified in the RREQ, and the $Dest\_Seq\#$ is set to the responding MMA's record of the destination's sequence number. The Hop_Cnt field is set to the distance of the responding MMA from the destination, or zero if the destination itself sends the RREP. The $Lifetime$ field is set to the time for which MMAs receiving the RREP consider the route to be valid.

Here we point out that to limit the number of RREQ packets propagated, a MMA processes a RREQ packet only if it has not already seen the packet. We assume that in our analytic ad-hoc network, only symmetric links are supported. So an intermediate MMA can deliver the RREP packet on the reverse route of RREQ packet. In asymmetric-link ad-hoc networks, an intermediate MMA must initiate a test route discovery to the MMA of the sender node and piggyback the RREP packet on this new route request.

4.3 Multicast Tree Maintenance

Multicast tree maintenance is very important in ad-hoc networks since the network topology changes very often. Link breakages must be repaired in a timely manner to maximize multicast group connectivity. In our protocol a route error (RERR) packet and acknowledgements (ACKs) are used for route maintenance. Our multicast routing tree maintenance is based on the mobility information of both MMAs and non-spine nodes.

4.3.1 A Non-spine Node Comes to a New Place

When a mobile node moves to a new place, it will not receive any query message from its former MMA within a certain time.

If it comes to a place where there is already at least one MMA available, the MH will receive query messages from these MMAs. It checks the query messages and selects a MMA that still has dominating capability as its current MMA. Then it broadcasts a RREQ packet, in which the *J_flag* is set to one and *MMA_ID* is set to ID of the selected MMA. After receiving its RREQ, the corresponding MMA will create an item for it and broadcasts a RREP packet to it. Thus the mobile node is dominated by the MMA. If the MMA is not a member of the multicast group that the new comer joins, it must join this multicast group right now. When a MMA wants to join a multicast group, it broadcasts a RREQ packet, and only any other MMA of the desired multicast tree may respond to it. At the same time the responding MMA stores routes to the new multicast member in its buffer and broadcasts a RREP packet among the multicast group members.

If a non-spine node moves to a new place where there is not a MMA available, it cannot receive a query message from any MMA within a certain time. It will send a RREQ packet to its cluster head and declare itself as a MMA. After receiving its RREQ packet the cluster head will unite the current spine and this new MMA to repair the multicast tree. If there is not a cluster head reachable yet, it will declare itself a new cluster head.

4.3.2 A Non-spine Node Moves Away

When a mobile node moves out of range of its current MMA, it need not send any control packet to its former MMA. Every a certain time, a MMA will send a query packet and require the reply of its dominated nodes. If the MMA cannot hear the reply of a mobile node, the MMA will consider it has left this area and then delete its item. In this case, if it is a receiver node, the MMA will store the datagram in the retransmission buffer from then on; and the receiver node will receive the datagrams after it registered with a new MMA [14]. If it is a sender node, the MMA will send a RERR packet to its neighboring MMAs. Then the RERR packet is flooded to all MMAs. Section 4.3.4 gives detailed description.

4.3.3 A MMA Moves Away

Movement management of a MMA is more complicated than that of a non-spine node. Figure 3 shows a MMA moves away from position 1 to position 2. If a MMA does not receive reply from any dominated MH within

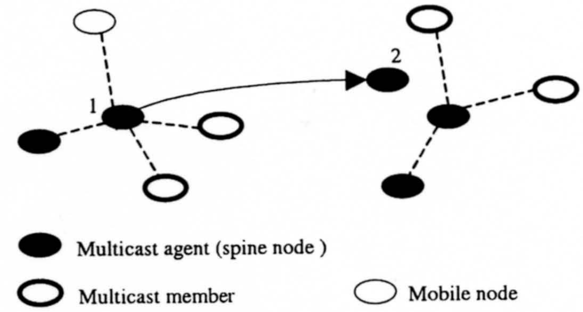

Fig. 3 A multicast agent moves from position 1 to 2.

a period of *Hello* broadcasts (query messages), it will think itself moves to a new place. Then it can join a cluster and select a spine node as its MMA. At this time itself is just a common mobile node. All the former multicast routing tree tables and information about dominated mobile nodes will be deleted. Each mobile node managed by the moving MMA will register with a new MMA by sending a RREQ packet or declare itself as a new MMA. Section 4.3.1 already describes this case in detail.

If a MMA does not receive any message from its neighboring MMAs within a period of exchange messages, it will remove all the corresponding information about that neighbor. In addition it will also produce a RERR packet and broadcasts it to the other MMAs.

4.3.4 RERR Packets

When a MMA encounters a fatal transmission problem at its data link layer, it generates a RERR packet. When its neighboring MMAs receive this RERR packet, they will remove the corresponding route information from their route table. All routes that contain the hop in error are truncated. Moreover, it may prune itself from the current multicast tree.

When a non-spine multicast member decides to terminate its membership in the multicast group or encounters a fatal transmission problem, it also generates a RERR packet. When its MMA receives this RERR packet, it will remove the corresponding route information from their route table. In addition, the MMA will check whether there is any other dominated multicast member. If no one exists, the MMA itself is not a multicast receiver, and the MMA is just a leaf node in the multicast tree, the MMA will prune itself from the multicast tree.

5. Evaluations

In this paper in order to show the multicast routing with MMAs is effective, we compare some performance parameters between multicast routing with multicast gents and multicast routing with general AODV protocol. The performance parameters include multicast

Table 1 Parameter values for numerical analysis.

Meaning	Value
Network size	50 MHs
Multicast group size	23 MHs
Network boundary size	50 meters
Wireless link bandwidth	2 Mb/second
Transmission range	10 meters
Period of query broadcasts	1 second
Period of exchange messages	5 seconds
Nodes managed by a MMA: n	4 MHs

tree structure, number of RREQ packets, number of RREP packets, etc. The main objective is to show that multicast algorithm with MMAs can be used to greatly reduce control data overhead.

Table 1 gives the parameter values of our analysis. Every mobile host has the same transmission range. Mobile nodes are free to move within the ad-hoc network. Movement is measured by the average speed. When nodes reach the boundary of the ad-hoc network, they will bounce back and continue to move. It is possible that another MMA will manage mobile nodes within the transmission range of a MMA. We did so in order to keep balance of the number of mobile nodes that every MMA manages.

In multicast operation of current AODV algorithm, every multicast tree is composed of all members of a multicast group. Each multicast receiver must keep a routing table and its neighbor information. However in multicast with MMAs, a multicast tree is only composed of MMAs, which dominate all multicast members together. Thus we minimize the memory and battery consumption of a multicast receiver that is not a MMA. CBRP multicast operation places too much burden on cluster heads, since a cluster head must manage all the multicast members within its transmission range. However in multicast operation with MMAs, each MMA manage at most 4 multicast members, from our assumption. Thus we can reduce the burden of a cluster head by distributing them to MMAs reasonably. In fact CBRP multicast is just a special case of multicast with MMAs. Multicast with MMAs turns CBRP multicast when a spine only includes cluster heads. Discovery of multicast routing tree with MMAs is only relative to MMAs and has nothing to do with their dominated MHs. We can reduce control overhead when finding a multicast tree and find it rapidly. However in AODV multicast, all multicast members must store corresponding routing information. The tradeoff of our proposal is that non-optimal routes are utilized to forwarding datagram. However, as we know, routing optimality is not the most important factor in ad-hoc networks, especially under high mobility environment. We can improve the speed of finding a multicast tree. And also quality of service (QoS) assurance can be bettered a great deal [14].

Figures 4(a), (b) illustrate multicast tree with MMAs and general AODV protocol, respectively, in static mode. From Fig. 4, we can conclude that by our proposal the size of a multicast group (only composed of MMAs) is reduced from 23 (number of multicast members) to 11 (number of MMAs) in static mode so that there are fewer RREQ packets and RREP packets, which are shown in Table 2.

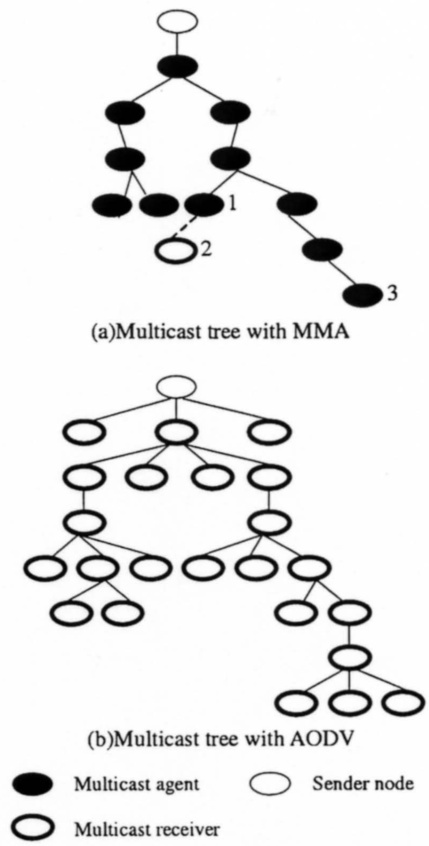

Fig. 4 Multicast tree in static mode.

Table 2 Control overhead in Fig. 4.

Protocol	RREQ	RREP	Sum
AODV multicast	10	86	96
MMA multicast	8	24	32

Taking node mobility into consideration, we think additional control overhead must be spent for multicast tree management. Nodes in mobility include MMAs and multicast members that dominated by MMAs. A node can move to a new place where there is already a MMA available or not. Based on the above analysis, we divide node movement into 4 types:

1. A non-spine node moves to a place where there is already a MMA available, e.g. in Fig. 4(a) node 2 moves close to node 3.

2. A non-spine node moves to a place where there is not a MMA available, e.g. node 2 comes to a new place where there is not a MMA available.

3. A MMA moves to a place where there is already another MMA available, e.g. node 1 moves close to node

Table 3 Control overhead when a node moves to a place, where there is already a MMA available.

Movement scenario	RREQ	RREP	Sum
A node in AODV multicast moves	1	7	8
A non-spine node moves	1	1	2
A multicast agent moves	3	3	6
A node in MMA moves (average)	1.4	1.4	2.8

Table 4 Control overhead when a node moves to a place, where there is not a MMA agent available.

Movement scenario	RREQ	RREP	Sum
A node in AODV multicast moves	1	7	8
A non-spine node moves	1	6	7
A multicast agent moves	3	8	11
A node in MMA moves (average)	1.4	6.4	7.8

3.

4. A MMA moves to a place where there is not a MMA available, e.g. node 1 comes to a new place where there is not a MMA available.

These 4 movement cases are already described in Sect. 4.

Node movement includes movement of a non-spine node and a MMA. By our definition, every MMA can manage n non-spine nodes in the analyzed ad-hoc network. Therefore a moving node is a non-spine node or MMA at the probability of $n/(n+1)$ or $1/(n+1)$, respectively. We define the number of the additional RREQ and RREP packets caused by movement of a non-spine node and a MMA as RREQ0, RREP0, RREQ1 and RREP1, respectively. We also define the average number of additional RREQ and RREP packets caused by movement of a mobile node as RREQ2 and RREP2. Thus we can get:

$$RREQ2 = RREQ0 \cdot \frac{n}{n+1} + RREQ1 \cdot \frac{1}{n+1} \quad (1)$$

$$RREP2 = RREP0 \cdot \frac{n}{n+1} + RREP1 \cdot \frac{1}{n+1} \quad (2)$$

In our analysis we assume n is equal to 4. Then we can get Table 3 and Table 4, which shows the additional RREQ and RREP packets when the movement of a node belongs to type 1 or 3, type 2 or 4, respectively. Sum means the number of total control overhead including RREQ and RREP packets.

Movement of a MMA will cause more additional control overhead than movement of a non-spine node. This is because its movement will also cause its dominated nodes to register with a new MMA.

Movement of a node to an area where there is not a MMA available will cause more additional control overhead than movement of a node to an area where there is already a MMA available. Same additional RREQ packets are needed. However, the later one need more RREP packets because after declaring itself as a new MMA it must broadcast a RREP packet and intermediate MMAs must produce corresponding RREP packets too.

In our proposal, a cluster-spine hierarchy is formed whereby broadcasts are not flooded across every node in the network, but instead across only MMAs in the network, thereby reducing the control overhead and bandwidth consumed by such broadcasts.

6. Conclusions and Future Work

In this paper we propose a multicast routing algorithm based on mobile multicast agents (MMAs) in an ad-hoc network. First, an ad-hoc network is partitioned into two-level hierarchy architecture, i.e. cluster and spine. A spine is composed of MMAs. Second a multicast tree discovery/maintenance scheme is performed based on the spine infrastructure. Not all wanting mobile nodes form a multicast group but only MMAs from these nodes form a multicast tree.

In comparison with AODV multicast we simplify the multicast tree discovery, reduce control overhead of the network, and increase the total network throughput. We also overcome the deficiency of CBRP multicast routing, which places much burden on cluster heads.

The tradeoff is the buffer and power consumption of MMAs. Time must be used to construct a spine by seeking MMAs before communications.

In future work we will consider the extension of dynamic source routing (DSR) protocol with MMAs in ad-hoc networks in order to achieve multipath routes. QoS MMAs in ad-hoc network will also be taken into consideration.

References

[1] H. Ericksson, "Mbone: The multicast backbone," Commun. ACM, vol.37, no.8, pp.54–60, Aug. 1994.

[2] Z.J. Haas and M.R. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks," Internet draft, draft-zone-routing-protocol-00.txt, Nov. 1997.

[3] P. Krishna, N.H. Vaidya, M. Chatterjee, and D.K. Pradhan, "A cluster-based approach for routing in dynamic networks," Proc. Second USENIX Symposium on Mobile and Location-Independent Computing, 1995.

[4] M. Jiang, J. Li, and Y.C. Tay, "Cluster based routing protocol (CBRP)," Internet draft, draft-ietf-manet-cbrp-spec-01.txt, July 1999.

[5] E.L. Madruga and J.J. Garcia-Luna-Aceves, "Multicast along meshes in ad-hoc networks," Proc. IEEE ICC99, Vancouver, Canada, 1999.

[6] S.-J. Lee, W. Su, and M. Gerla, "On demand multicast routing protocol (ODMRP) for ad hoc networks," Internet Draft, draft-ietf-manet-odmrp-02.txt, Jan. 2000.

[7] R. Sivakumar, B. Das, and V. Bharghavan, "Spine routing in ad hoc networks," ACM/Baltzer Publications Cluster Computing Journal, Special Issue on Mobile Computing, 1998.

[8] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in ad-hoc networks using a spine," International Conference on

Computers and Communications Networks'97, Las Vegas, NV, Sept. 1997.

[9] R. Sivakumar, B. Das, and V. Bharghavan, "The clade vertebrata: Spines and routing in ad hoc networks," Proc. International Symposium on Computer Communications (ISCC '98), 1998.

[10] C.E. Perkins and E.M. Royer, "Ad hoc on demand distance vector routing," Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp.90–100, New Orleans, LA, Feb. 1999.

[11] C.E. Perkins, E.M. Royer, and S. Das, "Ad hoc on demand distance vector (AODV) routing," IETF Internet draft, draft-ietf-manet-aodv-06.txt, July 2000 (Work in Progress).

[12] E.M. Royer and C.E. Perkins, "Multicast operation of the ad-hoc on demand distance vector routing protocol," Proc. MobiCom'99, pp.207–218, Seattle, WA, Aug. 1999.

[13] E.M. Royer and C.E. Perkins, "Multicast ad hoc on demand distance vector (MAODV) routing," IETF Internet draft, draft-ietf-manet-maodv-00.txt, July 2000 (Work in Progress).

[14] W. Xin, F. Li, S. Ishihara, and T. Mizuno, "Reliable spine-based multicast for mobile ad-hoc networks," Proc. MoMuC'2000, 3A-3-1–6, Oct. 2000.

Xin Wang was born in Sanyuan, Shaanxi, China, on August 29, 1973. He received the BE degree in information engineering and the M.E. degree in communications and electronic systems from Xidian University, China, in 1994 and 1997, respectively. From 1997 to 1998, as a visiting student, he studied mobile computing in Shizuoka University, Japan. In 1998 he worked in R&D department of Huawei Technologies Co., Ltd., China, where he was engaged in research and development on the digital video broadcasting systems. Since 1999 he has been a PhD candidate in Shizuoka University, working on wireless routing algorithms and dynamic priority queueing analysis. His research interests include mobile computing, network protocols, queueing theory and neural networks. He is a student member of the Information Processing Society of Japan.

Fei Li received the B.E. and the M.E. degrees from Air Force Engineering University, China, all in computer science, in 1994 and 1999, respectively. Now she is with a computer research institute, Xi'an, China, as a research associate. Her research interests include computer networks, personal communication systems and neural networks.

Susumu Ishihara received his B.E. and Ph.D. degrees in 1994 and 1999 from Nagoya University, Japan. From 1998 to 1999, he was a Special Researcher of JSPS fellowship. Since 1999, he is a research associate of Faculty of Information, Shizuoka University, Japan. His research interests are in the design and analysis of local area networks, distributed computer systems and mobile computing. He is a member of the Information Processing Society of Japan and ACM.

Tadanori Mizuno received the B.E. degree in Industrial Engineering from the Nagoya Institute of Technology in 1968 and received the Ph.D. degree in computer science from Kyushu University, Japan, in 1987. In 1968, he joined Mitsubishi Electric Corp. Since 1993, he is a professor of Faculty of Engineering, Shizuoka University, Japan. He moved to the Faculty of Information, Shizuoka University in 1995. His research interests include mobile computing, distributed computing, computer networks, and protocol engineering. He is a member of the Information Processing Society of Japan, the IEEE Computer Society and ACM.