Weight-Based Clustering Multicast Routing Protocol for Mobile Ad Hoc Networks

Chun-Chieh Huang and Ruay-Shiung Chang
Department of Computer Science and
Information Engineering, National Dong-Hwa
University, Hualien, Taiwan R.O.C
E-mail: rschang@mail.ndhu.edu.tw

Ming-Huang Guo
Department of Information Management,
Shih-Hsiu University, Taipei, Taiwan, R.O.C
E-mail: mhguo@cc.shu.edu.tw

Abstract—**In this paper, we propose a new multicast routing protocol, named weight-based clustering multicast protocol (WCMP), for mobile ad hoc networks. The goal of this work is to improve multicasting performance in ad hoc networks by efficient use of the available knowledge of the networks.**

Keyword—multicasting, routing, mobile ad hoc networks

1. Introduction

An ad-hoc network is a packet switching network based on wireless links for router interconnection. The topology of an ad hoc network can be very dynamic due to the mobility of routers and the characteristics of the radio channels. Therefore, multicasting in mobile ad hoc networks is more challenging than in the Internet, because of the need to optimize the use of several resources simultaneously. The traditional multicast methods proposed for wired infrastructures [1-4] do not address the dynamically changing topology of the network and may not be directly applicable to ad hoc networks.

Multicast protocols existed for wireless ad hoc networks can be classified into tree-based [5-7] and mesh-based [8-10]. The tree-based multicast routing is very attractive for wireless network and the Internet because of its simplicity. In this paper, we propose a new multicast routing protocol, named weight-based clustering multicast protocol (WCMP), for mobile ad hoc networks. The goal of this work is to improve multicasting performance in ad hoc networks by efficient use of the available knowledge of the networks. The basic idea is for the source to construct a multicast delivery tree first. Then we select some nodes in the tree to become clusterheads using a weighted algorithm. The clusterheads are used to reduce the overhead and packet loss. Additional, we carefully select and determine the number of clusterheads to avoid throughput degradation due to clusterhead's control messages.

2. WCMP Protocol

The WCMP multicast session uses the acknowledgement tree for multicast in ad hoc networks. A multicast source initiates the creation of a multicast data delivery tree. The tree creation is done by ACK messages. The

0-7803-7700-1/03/$17.00 (C) 2003 IEEE

source sends a Join_Tree_Request packet to each node. When a mobile node interested in the multicast group receives the Join_Tree_Request packet, it will reply to the source with a Join_Ack message. Figure 1 illustrates the multicast tree creation. Source S initiates tree creation by sending Join_Tree_Request messages to all mobile nodes. Nodes *A*, *B* and *D* are interested in the multicast session and they respond to the source with Join_Ack messages.

When the multicast tree has established, each node obtains the neighboring nodes information. After that, we use the method described below to find out some nodes to act as the clusterheads. Each node will belong to a certain cluster and is managed by its clusterhead. The clusterhead maintains and records group member's information and links. The nodes managed by a clusterhead have only to keep a limited number of messages in order to reduce the overhead. Before describing the method, we have two assumptions. First, each node correctly receives and transmits the messages in a limited time. Second, when a node moves into or moves out someone's transmission range, it can detect the new connection or delete the old connection in a limited time.

The cluster information is obtained by periodically flooding the beacon packets within 2-hop. After gathering the information, each node compares its weight. The main purpose is to select a clusterhead according to a weight mechanism. Each node has different weight value. Three rules are defined to compute the weight. Initially, we set weight $W_i = 0$, for $0 \leq i \leq n\text{-}1$, assuming that the number of nodes is *n*. Let $N=\{0, 1, 2, \ldots, n\text{-}1\}$ be the set of mobile nodes in the ad hoc network. $NN_2(i)$ is the number of neighboring nodes within 2-hop of *i*, including *i*. *Deg(i)* is the degree of *i*. If *Deg(i)*=1, then *i* is called a border node. The following are three parameters used in computing W_i:

1. The number of border nodes within 2-hop of *i*, denoted as BN_i
2. The number of neighboring nodes for *i* within 2-hop, i.e., $NN_2(i)$
3. Degree of *i*, i.e., *Deg(i)*.

The weight function W_i is ($0.7 \times BN_i + 0.2 \times NN_2(i) + 0.1 \times Deg(i)$), for $0 \leq i \leq n\text{-}1$.

In general, a border node *i* is a receiver. So, if a node manages more border nodes, it means that it is closer to the center of the network. It is a good candidate for a clusterhead. Therefore, BN_i is multiplied by 0.7. Next, 2-hop information is more important than 1-hop. If node *i*'s 2-hop neighbors are more than those of *j*, it means that node *i* can manage more nodes than *j* and is closer to the center of network. Thus, the multiplication coefficient is 0.2 and 0.1. We then compute the weight of a set of certain nodes to select the clusterhead. The steps are described below:

Step 1: Calculate the weight of each node at the same height in the multicast tree. When a node's weight is greater than all other nodes, then the node is the clusterhead at this height of multicast tree. The clusterhead must join the multicast tree.

Step 2: When a clusterhead is chosen, it manages all of its downstream nodes.

Step 3: If two clusterhead nodes are neighbors and in the same branch of the multicast tree, we choose the one close to the source as the clusterhead. The other is relieved of its clusterhead role.

Step 4: If a node's degree is 1 or there is only one node in a level, we do not select any clusterhead in this level of multicast tree.

Step 5: Other unattended nodes are managed by source. So source is also a clusterhead.

Assume the multicast tree is constructed as shown in Figure 2. Figure 3 redraws the tree to look as rooted at *S*. According to our weight function, nodes *A* and *B* have the same degree in height = 1. BN_A are nodes *J, L* and *M* within 2-hop. So, BN_A is 3. $NN_2(A)$ are nodes *A, S, L, C, I, B, J, M* and *E* within 2-hop. So, the value of $NN_2(A)$ is 9. The weight of node *A* is W_A= (0.7× 3+0.2×9+0.1×3) = 4.2. BN_B are nodes *F, G, H* and *J*. The value of BN_B is 4. $NN_2(B)$ are nodes *B, H, D, S, F, G, I, L* and *A*. The value of $NN_2(B)$ is 9. The weight of node B is W_B= (0.7×4+0.2× 9+0.1×3) = 4.9. In the same manner, nodes *C* and *D* have the same degree in height = 2. The weight of node *C* is W_C= (0.7×2+0.2×7+0.1×3) = 3.1. The weight of node *D* is W_D= (0.7×3+0.2× 6+0.1×3) = 3.6. Node E is the clusterhed because it has the maximum degree in the multicast tree of height = 3. After comparing the weights, nodes *B* and *D* are neighbors and clusterheads in the same branch of multicast tree. We just choose the node closer to the source as the clusterhead. This can decrease the clusterhead nodes in a multicast tree. The result of selecting the clusterheads is showing in Figure 4.

WCMP has two stages. In the first stage, a multicast delivery tree rooted at source is constructed. Then the weighted mechanism is used to select the clusterheads. Then, the source periodically sends a packet CHREQ (clusterhead request) to find out the clusterheads. Let the clusterhead request packet be denoted as CHREQ (*S, HP, FG*), where *S* represents the source, *HP*, history-path, is the path that the packet has been traveled, and *FG*, a flag, is used by a node to show it is a clusterhead. When a node receives the CHREQ packet, it records CHREQ's history-path to routing table and updates CHREQ's history-path. If a node is a clusterhead, it sends out a CHREP (clusterhead reply) message to the source. The clusterhead sets the flag *FG* to *TRUE*, and forwards the packet according to the history-path of the CHREQ to source. Figure 5 illustrates the operation of WCMP. When the clusterheads *S*, *A* and *B* are determined, source *S* sends the CHREQ to *A* and *B* to make sure the paths from *S* to *A* and from *S* to *B*. When node *A* receives the CHREQ message, it sets *FG* to *TRUE* and sends back the CHREP message to source *S*. When source *S* sends CHREQ message to *B*, the path traveled will be recorded in history-path <*S, C, D, B*>. After *B* receives the CHREQ message, it sets the *FG* to *TRUE* and sends the CHREP along the history-path to source *S*.

Furthermore, we have to properly maintain the multicast delivery tree and reconstruct the local tree when a node joins or leaves the multicast tree. A node joining or leaving the local tree may cause the local tree to merge, split or rearrange around the clusterhead. If a node wants to join the multicast group it sends a JOIN message to the clusterhead in that local tree. A JOIN message is denoted as JOIN (*ID, CH, FG*), where *ID* represents the join node issuing this request packet, *CH*, clusterhead in that local tree, *FG*, flag, being *TRUE* denotes the clusterhead agrees the node to join the cluster. Otherwise it is *FALSE*. Figure 6 shows an example when a node wants to join the multicast tree. Node *D* wants to join the multicast group, it first broadcasts to search for a neighbor. And then it sends a JOIN (*D, S, FALSE*) message toward the clusterhead. Node *C* forwards the message to the clusterhead. When the clusterhead receives the packet, it sets the flag to *TRUE* and sends back to node *D*. The join operation is complete. After that, in Figure 7 node *C*'s degree is 4. Node *C*'s degree is maximum in height = 2. Therefore, we choose node *C* as the new clusterhead.

In the same branch, node *B* is also a clusterhead. So, we choose the one closest to the source, node *C*, as the clusterhead and let it manage its downstream nodes. The source periodically sends a CHREQ to find new clusterheads. Node *C* sends back a CHREP to the source. The result of the multicast tree after rearranging the clusterhead is shown in Figure 8.

To handle mobility, node failures, and a node leaving a local multicast tree, multicast delivery tree will be periodically refreshed. When a node leaves the multicast tree, the neighbor nodes only update the Neighbor Table and the clusterhead updates its cluster information. If a clusterhead leaves the multicast tree, its downstream nodes will search for an adjacent local multicast tree and send JOIN messages to source. After that, the multicast tree adjusts the clusterheads if needed. The operation is the same as a new mobile node joins a multicast tree.

3. Simulation Results and Analysis

In this section, we compare the performance of WCMP routing protocol with that of ODMRP protocol. Results show that WCMP routing protocol outperforms ODMRP protocol.

We simulate the multicast routing in ad hoc wireless networks by Global Mobile Simulation (GloMoSim) library [11], a C language based, parallel simulation language developed at UCLA. Our simulation modeled a network of 50 mobile nodes and moving area on a rectangular 1000m by 1000m.

In order to avoiding the asymmetric routing or unidirectional links, we assume each mobile node could communicate with each other by the same routing paths. Asymmetric routing means that the routing path from *A* to *B* may differ from the path from *B* to *A*. Simulations consider one multicast group and the source node is randomly chosen. A variable number of randomly chosen receivers join the channel.

We compare several parameters for our method in the following:

1. Packet delivery ratio: defined as the number of multicast datagrams delivered to mobile nodes over the number of multicast datagrams supposed to be delivered to mobile nodes.
2. Control message overhead: defined as the average number of control messages received by the mobile nodes.
3. Number of clusterhead: defined as the average number of clusterheads when mobile nodes increase.

Figure 9 and Figure 10 show the relative performance of WCMP and ODMRP routing protocols for different number of mobile nodes. In order to keep the forwarding group current, ODMRP has to trigger global flooding of control packets. That will be a serious overhead. The concept of WCMP can avoid the global flooding of control messages, and keep the stability of multicast tree. As is expected, the performance of both routing protocols (WCMP and ODMRP) degrades with increasing number of mobile nodes. We can observe the proposed WCMP routing protocol outperforms ODMRP. Figure 11 shows the number of clusterhead variation in our method. We can control the number of clusterheads in a stable situation. When the number of mobile nodes is between 20 and 30, the network topology is stable in Figure 11.

4. Conclusions

In this research, we proposed a new multicast routing protocol, named Weight-based Clustering Multicast Routing Protocol (WCMP). The performance shows that our proposed method outperforms the existing mesh-based multicast routing protocol in mobile ad hoc networks.

Reference

[1] S. Deering, C. Partrige, and D. Waitzman, "Distance Vector Multicasting Routing Protocol," *IETF RFC 1075*, 1988.

[2] J. Moy, "Multicast Routing Extensions for OSPF," *Communication of the ACM*, vol. 37, no. 8, pp. 61-66, Aug. 1994.

[3] A. Ballaradie, J. Crowcroft, and P. Francis, "Core Based Tree (CBT) – An Architecture for Scalable Inter-Domain Routing Protocol," *SIGCOM*, pp. 85-95, Oct. 1993.

[4] S. Deering, D. Estrin, D. Fairnacci, V. Jacobson, C. Liu, and L. Wei, "An Architecture for Wide Area Multicast Routing," *SIGCOM*, pp.126-135,Oct. 1994.

[5] E. M. Royer and C. E. Perkins. "Multicast operation of the ad-hoc on-demand distance vector routing protocol," *Mobicom,* pp. 207-218, Aug. 1999

[6] Bommaiah, McAulcy, Talpade, and Liu. "AMRoute: Ad-hoc Multicast Routing protocol," draft-talpade-manet-amroute-00. txt, Aug. 1998. (work in progress)

[7] C. Wu, Y. Tay, and C.-K. Toh. "Ad hoc Multicast Routing protocol utilizing Increasing id-numberS(AMRIS) Functional specification," Nov.1998 (work in progress). Acessed Mar. 18, 2000. Available at: http://www.freenic.net/drafts/drafts-i-j/draft-ietf-manet-amris-spec-00.html.

[8] S.-J. Lee, M. Gerla, and C.-C. Chiang. "On-demand multicast routing protocol," *of IEEE WCNC,* pp. 1298-1302, Sep. 1999.

[9] J. J. Garcia-Luna-Aceves and E. L. Madruga. "The core-assisted mesh protocol," *IEEE Journal on Selected Areas in Commun.*, pp.1380-1394, Aug. 1999.

[10] P. Sinha, R. Sivakumar, and V. Bharghavan. "MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing," *IEEE WCNC*, pp. 1313-1317, Sept. 1999.

[11] University of California Los Angles Computer Science Department Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems, http://pcl.cs.ucla.edu/projects/glomosim/

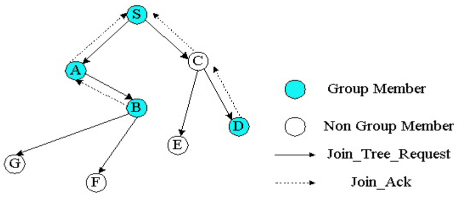

Fig. 1: Multicast Tree Creation

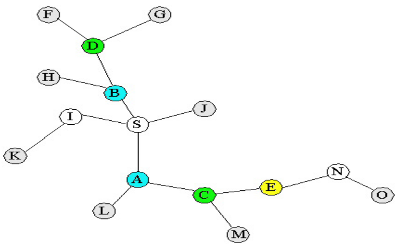

Fig. 2: Original multicast tree

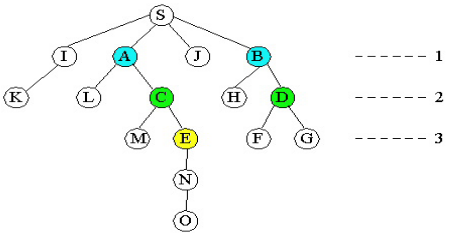

Fig. 3: Transfers the search tree

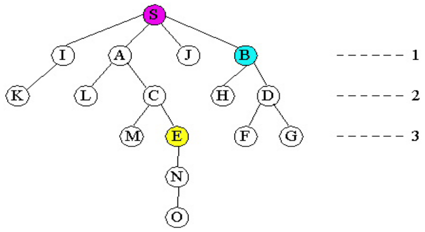

Fig. 4: Result of deciding the clusterhead for multicast tree

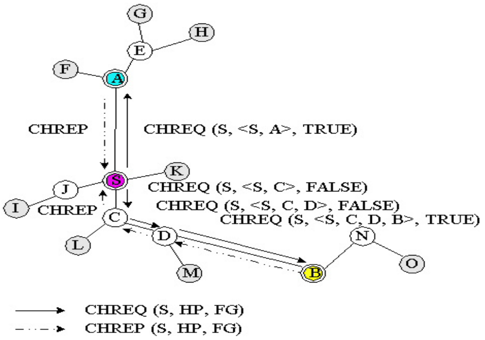

Fig. 5: An example of clusterhead request and reply process

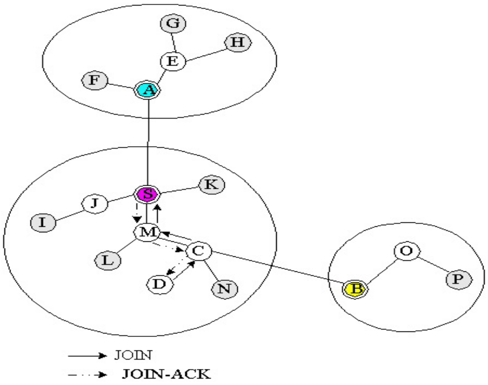

Fig. 6: An example when a node joins the multicast group

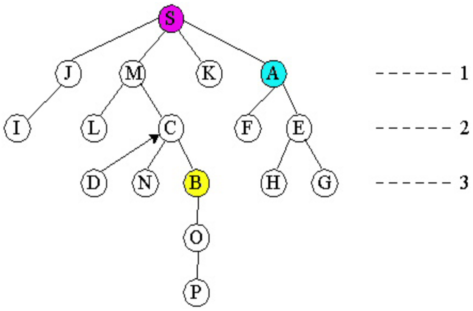

Fig. 7: C’s degree is 4 in height = 2

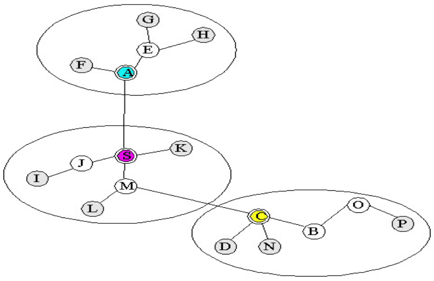

Fig. 8: The multicast tree after a node joining

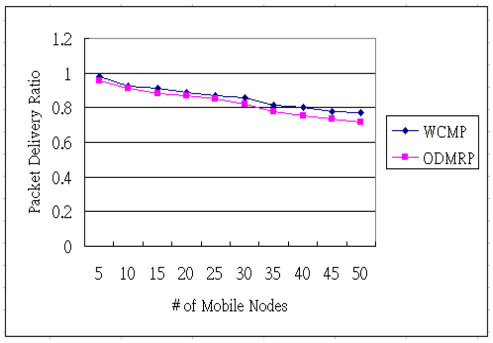

Fig. 9: Packet Delivery Ratio vs. Number of Mobile Nodes

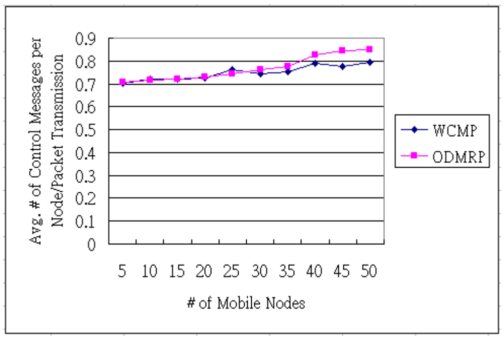

Fig. 10: Control Overhead vs. Number of Mobile Nodes

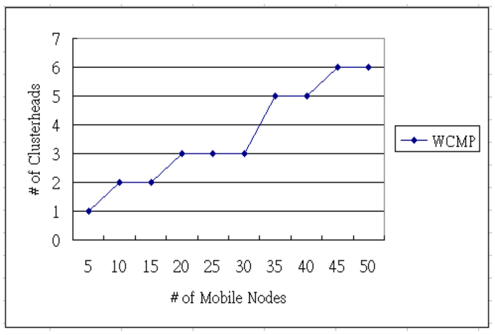

Fig. 11: Number of clusterheads when mobile node increases