Bandwidth-Efficient Multicast Routing for Multihop, Ad-Hoc Wireless Networks*

Tomochika Ozaki
Multimedia Systems R & D Division
Hitachi, Ltd.
292 Yoshida-cho, Totsuka-ku, Yokohama
244, Japan
phone: 81-45-881-1241,
fax: 81-45-860-1645,
email: ozaki@msrd.hitachi.co.jp

Jaime Bae Kim
Department of Computer Science
California State University, Northridge
Northridge, CA 91330-8281
phone: 818-677-3892,
fax: 818-677-7208,
email: jkim@ecs.csun.edu

Tatsuya Suda
Department of Information and
Computer Science
University of California, Irvine
Irvine, CA 92717
Phone: 949-824-5474,
fax: 949-824-4056,
email: suda@ics.uci.edu

Abstract-- In this paper, we propose and investigate a bandwidth-efficient multicast routing protocol for ad-hoc networks. The proposed protocol achieves low communication overhead, namely, it requires a small number of control packet transmissions for route setup and maintenance. The proposed protocol also achieves high multicast efficiency, namely, it delivers multicast packets to receivers with a small number of transmissions. In order to achieve low communication overhead and high multicast efficiency, the proposed protocol employs the following mechanisms: (1) on-demand invocation of the route setup and route recovery processes to avoid periodic transmissions of control packets, (2) a new route setup process that allows a newly joining node to find the nearest forwarding node to minimize the number of forwarding nodes, and (3) a route optimization process that detects and removes unnecessary forwarding nodes to eliminate redundant and inefficient routes. Our simulation results show that the proposed protocol achieves high multicast efficiency with low communication overhead compared with other existing multicast routing protocols, especially in the case where the number of receivers in a multicast group is large.

A. INTRODUCTION

An ad-hoc network is a collection of wireless mobile nodes, which form a temporary network without relying on the existing network infrastructure or centralized administration [1]. In traditional cellular networks, a mobile node is only one hop away from a base station, which is connected to a wired backbone. On the contrary, ad-hoc networks form a multi-hop network where all communication is over the wireless channel, hopping over several mobile nodes. Typical applications of ad-hoc networks include outdoor special events (such as conferences, concerts and festivals), as well as communications in regions with no infrastructure, in emergencies and natural disasters, and in military maneuvers.

Following are salient features of ad-hoc networks:

The network topology is highly dynamic due to frequent node migration and power outages/failures of mobile nodes.

Multi-hopping over several mobile nodes may be necessary to reach destinations due to the limited transmission power of mobile nodes.

Resources (e.g., channel bandwidth, node resources such as computational power, storage capacity, battery power, etc.) in ad-hoc networks are very limited.

These unique features of ad-hoc networks pose several new challenges in the design of routing protocols. For example, since mobile nodes act as routers in ad-hoc networks and they have very limited resources, conventional routing protocols which employ frequent route updates through periodic control packet transmissions may not be suitable for ad-hoc networks [2]. Further, routing protocols for ad-hoc networks must be highly adaptable to frequent movements and failures in ad-hoc networks. Multi-hopping further complicates the routing.

In recent years, a number of new unicast routing protocols for ad-hoc networks have been proposed[3][4][5][6][7], but little work has been done in the area of multicast routing. In this paper, we focus on multicast routing and propose a "bandwidth-efficient" multicast routing protocol for ad-hoc networks. The proposed protocol requires only a small number of control packets to setup and maintain multicast routes, and thus, it has low communication overhead. The proposed protocol also requires only a small number of packet transmissions to deliver multicast packets to receivers, and thus, it has high multicast efficiency. Most of the past work on multicast routing considers only the communication overhead, ignoring multicast efficiency. However, multicast efficiency is also an important performance measure since it reflects how efficiently the protocol makes use of broadcast nature of wireless medium, and it is directly related to the bandwidth efficiency. In this paper, we propose a multicast routing protocol that achieves low communication overhead and high multicast efficiency.

[*] This research is supported by the National Science foundation through grants NCR-9628109, NCR-9714702 and ANI-9903427. It has also been supported by grants from the University of California MICRO program, Hitachi America, Hitachi, Canon Information Systems Inc., Nippon Telegraph and Telephone Corp. (NTT), Nippon Steel Information and Communication Systems Inc. (ENICOM), Tokyo Electric Power Co., Fujitsu, Novell, Matsushita Electric Industrial Co. and California State University of Northridge Research and Grants Program.

0-7803-7016-3/01/$10.00 ©2001 IEEE

B. MULTICAST ROUTING PROTOCOL

The proposed multicast routing protocol requires low communication overhead since it does not require periodical transmission of control packets. Most of the existing multicast routing protocols, such as DVMRP (Distance-Vector Multicast Routing Protocol) [8] and FGMP (Forwarding Group Multicast Protocol) [9], require periodical transmission of control packets in order to maintain multicast group membership and multicast routes, thereby wasting a lot of bandwidth. In the proposed protocol, route setup and route recovery are invoked only when they are required; route setup process is invoked only when a new node joins a multicast group, and route recovery process is invoked only when a multicast route breaks due to the node movements. Further, in the route recovery process, control packets used to recover multicast routes are flooded only to limited network area scoped by TTL (time-to-live). (In our protocol, hop count is used as TTL.) Limiting the scope of route search further decreases the communication overhead since control packets are not flooded to the entire network. MAODV (Multicast Ad-hoc On Demand Distance Vector) [15] also tries to minimize the communication overhead by invoking the route discovery process on-demand. However, unlike the proposed protocol, MAODV ignores multicsat efficiency.

The proposed multicast routing protocol also achieves high multicast efficiency, i.e., it requires a small number of multicast transmission. Multicast transmission is kept minimal by keeping the number of forwarding nodes small. Forwarding nodes are the nodes which broadcasts (forwards) multicast packets to neighboring nodes. Most of the existing multicast routing protocols use unicast protocols such as DSDV (Destination Sequenced Distance Vector) [12] and AODV (Ad-hoc On Demand Distance Vector) [13] to select the shortest paths from a source to each receiver. For example, in CBT (Core Based Tree)/PIM (Protocol Independent Multicast) based protocols [9][11], when a new node needs to join a multicast group, these unicast protocols are used to set up the shortest path to a core or Rendezvous Point. In FGMP, forwarding nodes are selected along the shortest paths chosen by these unicast protocols. In multicast environment, using the shortest paths from a source to each receiver does not always result in efficient multicast. Unlike these existing multicast protocols, the proposed protocol does not try to find a shortest path, instead, it tries to find the nearest forwarding node in the multicast group when a node wants to join the group. Nodes along the path between the nearest forwarding node and the new node become new forwarding nodes. This results in the minimum number of added forwarding nodes.

In addition, the proposed protocol provides a mechanism to detect unnecessary forwarding nodes and delete them from a multicast group. Due to the dynamic nature of ad-hoc environment, there may be unnecessary forwarding nodes in a multicast group. Route optimization process employed in the proposed protocol can detect and delete them from a multicast group to reduce unnecessary transmissions of multicast packets. This further increases multicast efficiency.

B.1 Route Setup Process

In the proposed protocol, a route setup process is invoked when a new node joins a multicast group. The route setup process finds the nearest forwarding node of the multicast group (to the newly joining node) and sets up a path between this nearest forwarding node and the newly joining node. Figure 1 illustrates this route setup process. First, a new node joining a multicast group, node X in this figure, broadcasts a JOIN packet. JOIN packets are flooded until they reach a forwarding node or a receiver node of a multicast group G. When a node floods a JOIN packet from node X, it records its node ID in the JOIN packet and increments the hop count contained in the JOIN packet. Therefore, a JOIN packet contains a list of nodes it has traversed and the number of hops it has taken. Note that the hop count indicates the number of new forwarding nodes which need to be added in order to add the newly joining node to the multicast group G.

A forwarding node or a receiver node in the multicast group G may receive more than one JOIN packets originated from node X, and the first received JOIN packet does not necessarily have the smallest hop count. (An example of this case is given later in Figure 2.) Therefore, in the proposed protocol, a forwarding node or a receiver node waits until they receive a predetermined number of JOIN packets or wait for a predetermined time period, and then choose a JOIN packet with the smallest hop count. REPLY packets are sent back to node X, following the path that the selected JOIN packet has traversed in reverse direction. The REPLY packet contains the ID of the forwarding node or the receiver node who generated the REPLY packet, the list of nodes traversed by the selected JOIN packet, and the hop count contained in the selected JOIN packet. When a receiver node which is not a forwarding node sends a REPLY packet, it increments the hop count by 1 since the receiver node will become a forwarding node, if the path contained in the REPLY packet is selected in the route setup process. As mentioned earlier, the hop count indicates the number of new forwarding nodes to be added.

Since multiple nodes can send REPLY packets to the newly joining node X, node X also waits until it receives a predetermined number of REPLY packets or wait for a predetermined time period, and then chooses a REPLY packet with the smallest hop count. A RESERVE packet is sent along the path that the selected REPLY packet has traversed.

Upon receiving a RESERVE packet, each node along the selected path updates its multicast routing table. A multicast routing table contains multicast group IDs, and for each multicast group ID, its upstream node ID and downstream node IDs. The final node on the selected path (i.e., the source in Figure 1) only adds a downstream ID to its multicast routing table.

After a new route is established from node X, multicast packets are forwarded along the new route. When a node receives a multicast packet of the multicast group G, if it has at least one downstream node of the multicast group G, it re-broadcasts the multicast packet. A multicast packet contains a sequence number and a hop count in addition to multicast data. The sequence number is used for duplicate detection.

The hop count is incremented at each forwarding node, and this hop count information is recorded locally before being forwarded. The hop count information recorded locally at each node is used in the route recovery process, which is described in Section B.3.

As mentioned earlier, a forwarding node or a receiver node in the multicast group G may receive more than one JOIN packets originated from the newly joining node, and the first received JOIN packet does not necessarily have the smallest hop count. This case is illustrated in Figure 2. In this figure, node A is a forwarding node and node E is a newly joining node. Lines between nodes represent connectivity between nodes. Two nodes are connected if they are within the transmission range of each other. When node E broadcasts a JOIN packet, nodes C and D receive it. Assume that node D forwards the JOIN packet earlier than node C. Since radio channel is busy, node C refrains from forwarding the JOIN packet while node D is transmitting. Further assume that node B forwards the JOIN packet received from node D earlier than node C. In this case, node A will receive the JOIN packet which has traversed nodes D and B first, but this JOIN packet has longer hop count than the JOIN packet through node C.

B.2 Route Prune Process

When a receiver node X of a multicast group G leaves the multicast group, it sends a PRUNE packet to its upstream node. Upon receiving the PRUNE packet, the upstream node checks if it has any downstream node other than node X. If it has, it simply deletes node X from the downstream entry in its multicast routing table. Otherwise, it sends a PRUNE packet to its upstream node and then becomes a non-forwarding node of the multicast group G by deleting the entry for the multicast group G from its routing table.

B.3 Route Recovery Process

A route recovery process is invoked when a multicast route is broken due to the node movements. In this paper, we propose and investigate the following two route recovery schemes:

Local-flooding scheme: This scheme finds a new route between the two end nodes of the broken link.

Local-rejoin scheme: This scheme finds a new route between the downstream node of the broken link and any forwarding node of the multicast group.

These two schemes are described below. Assume that two neighboring nodes A and B belong to a multicast group G, and that node B is the downstream node of node A. Assume also that the link between nodes A and B is broken because node B moved out of transmission range of node A. Refer to Figure 3. In local-flooding scheme, node A tries to find a new route to node B. When node A receives a multicast packet from the source of group G after the link is broken, it creates a special packet called a multicast-route-recovery packet, which contains the original multicast packet, and floods it. This flooding is done with a limited TTL (i.e., a limited hop count). When a node receives a multicast-route-recovery packet, it adds its node ID to the packet and re-floods it. Therefore, when node B receives a flooded multicast-route-recovery packet, it knows the exact route that the packet has traversed. Node B then sends a RESERVE packet to node A along the path that the multicast-route-recovery packet has traversed in the reverse direction. The REVERSE packet sets up a new route between node A and node B.

In local-rejoin scheme, the downstream node, node B, tries to find a new route to any Group G's forwarding node, which is not a downstream node of node B. When node B detects the link breakage, it floods a JOIN packet following the same procedure used in the route setup process described in Section B.1. However, this JOIN packet differs from the JOIN packet used in the route setup process in two ways: (1) it has a limited TTL to limit the scope of flooding, and (2) it includes Max_hop field which contains a hop count from the source to node B. As mentioned in Section B.1, each multicast packet contains a hop count from the source, and this hop count information is recorded locally in each node before the multicast packet is forwarded. Therefore, node B knows the hop count from the source to itself. When node B generates a JOIN packet, it records this hop count information in the Max_hop field of the JOIN packet. This Max_hop field is used to prevent the nodes in the downstream of node B to send REPLY packets. Only forwarding nodes whose hop counts from the source are smaller than or equal to the Max_hop contained in the JOIN packet send REPLY packets, and thus, the nodes in the downstream of node B will not send REPLY packets. Node A will become a non-forwarding node of the multicast group G, if it does not receive a RESERVE packet from node B for a predetermined time period, and if it does not have any other downstream nodes.

An advantage of the local-flooding scheme is that when a multicast packet arrives at the upstream node before the route recovery is completed, that packet will not be lost since the multicast packet is contained in the multicast-route-recovery packet and flooded. In local-rejoin scheme, the multicast packet will be lost in such a case. A drawback of local-flooding scheme is that it requires more bandwidth than the local-rejoin scheme, because the multicast-route-recovery packets used in the local-flooding scheme are much larger than the control packets used in the local-rejoin scheme.

In both schemes, packets for route recovery are not flooded to the entire network. They are flooded only to a limited network area scoped by TTL. This reduces the communication overhead of the protocol. The value of TTL, however, should be carefully chosen. Large TTL values increase the probability of successfully finding a new route, however, they also increase the communication overhead. Small TTL values reduce the communication overhead, however, they also reduce the chance of finding a new route. In this paper, we employ an adaptive TTL adjustment mechanism. In this mechanism, TTL is set to 2 for the initial route search, and every time a route search fails, TTL is incremented by 1 and a route search is performed again with the incremented TTL. The impact of TTL on performance is presented in the numerical result section.

Note that a route recovery process may sometimes fail. For example, the route recovery fails when a RESERVE packet is lost because one of the links in the path that the RESERVE packet traverses is broken after the path is selected. The route recovery also fails when node B (in Figure 3) is not in the flooding scope of node A in the local-flooding scheme or when there is no forwarding node in the flooding scope of node B in the local-rejoin scheme. In order to handle the failure of a route recovery process, a timer is associated with each multicast group in the multicast routing table. The timer for the multicast group G is refreshed every time a multicast packet for the group G is received. When a forwarding node of the multicast group G does not receive multicast packets for a while, the timer for the group G will eventually expire. When the timer for the multicast group G expires, the forwarding node removes the entry for the group G from its multicast routing table and becomes a non-forwarding node. When the timer at a receiver of the multicast group G expires, the receiver assumes a route fa ilure and invokes the route setup process.

B.4 Route Optimization Process

A route optimization process is invoked when a shorter route is found. As it will be illustrated later in Figure 4, shorter route is created when a forwarding node or a receiver node moves into the transmission range of forwarding nodes that are in the upstream of its upstream node. When a forwarding node or a receiver node receives a multicast packet whose hop count is smaller than the hop count of a multicast packet received from its upstream node, it changes its upstream node to the node from which the multicast packet with a smaller hop count is received. It also sends a PRUNE packet to the old upstream node to remove a redundant and less efficient route. When the old upstream node receives a PRUNE packet, it becomes a non-forwarding node, if it does not have any other downstream nodes. If it is not a receiver node, it further forwards the PRUNE packet to its upstream node. As a result, unnecessary forwarding nodes are removed, and a shorter route is established.

Figure 4 illustrates the route optimization process. Assume that nodes A, B, C and D are forwarding nodes, and node E is a receiver node of the multicast group G. Node E currently receives multicast packets through the route A-B-C-D-E. Assume that node E moves into the transmission range of node A. In this case, the multicast packet that node E receives from node A will have the smaller hop count than the multicast packet received from node D. This triggers a route optimization process; node E sends a RESERVE packet to node A to setup a new route directly to A and sends a PRUNE packet to node D to remove redundant and less efficient route. Since node D has no other downstream node, it becomes a non-forwarding node and sends a PRUNE packet to node C. The PRUNE packet is forwarded to node B and then to node A in a similar way. As a result, unnecessary forwarding nodes, B, C and D, are deleted.

As seen in the above example, the route optimization process removes unnecessary transmissions of multicast packets by detecting and removing unnecessary forwarding nodes. The route optimization process also helps to decrease the packet transfer delay since new routes are always shorter than old routes.

C. SIMULATION MODEL

C.1 Simulation Model Description

In our simulation, a flat network is assumed (i.e., no clusters). The following describes the MAC (Medium Access Control) layer protocol used in our simulation. For unicast, before a node sends a unicast packet, it sets RTS (Request-to-Send) flags of its neighbors and the intended receiver sets CTS (Clear-to-Send) flags of its neighbors. Nodes whose RTS or CTS flag is set cannot transmit data, except the sender. When the sender finishes sending the data, RTS/CTS flags are cleared by the nodes which originally set those flags. This MAC scheme represents existing schemes like MACA (Multiple Access Collision Avoidance) [14]. Similar scheme is also used for multicast; the node which wants to send a multicast packet sets RTS flags of its neighbors, and each intended receiver sets CTS flags of its neighbors. For broadcast used in flooding, only RTS flags are set by the sending node, and CTS flags are not set by any node. Therefore, in broadcast, collision may occur. However, collisions are ignored in our simulation. Using this relatively simple and generic MAC scheme allows us to investigate the proposed routing protocol without being strongly tied to the MAC layer scheme.

The simulated network area is a M x M meter square, and N mobile nodes are roaming randomly in all directions at a predefined speed in this area. Each node has a finite buffer, and packets are lost when buffer overflow occurs. Control packets have higher priority over data packets in our simulations[1]. Control packets used in our protocol are JOIN, REPLY, RESERVE, PRUNE and multicast-route-recovery packets. Propagation delay is assumed to be negligible, and it is assumed that packets always arrive without any bit error.

A multicast group has one source and a number of receivers. CBR (Constant Bit Rate) video/audio multicast application is assumed in our simulation, and thus, a source node generates multicast packets at a constant rate.

C.2 Simulation Parameters

In this section, parameter values used in the simulation are described. The channel speed of wireless link is 2 Mbps. The radio transmission range of a mobile node is 200 meters. A source node generates a multicast packet every 100 ms, and the size of a multicast packet is 1.6 Kbytes. This corresponds to the constant bit generation rate of 128 Kbps at a source. The size of all types of control packets except the multicast-route-recovery packets is around 100 bytes. Note that the size of control packets is variable, since some control packets contain the list of hops they traversed. The size of a multicast-route-recovery packet is at least 1.6 Kbytes, since it contains a

[1] The case where higher priority is not given to control packets is also simulated, and similar results are obtained without assuming priority of control packets.

multicast data packet. Buffer size at each node is 5Kbytes. The simulation time for each run is 100 seconds, and the position of each node is updated every 100 ms in the simulation.

The node mobility, the multicast group size and the network size are varied in the simulation to investigate the impact of each of these parameters on the performance of the proposed protocol. Table 1 shows the default values and the range of each parameter. When one parameter is varied, other parameters are set to the default values shown in this table. The default value of the simulated network area is a 1 Km x 1 Km square, and when the network size is varied from 30 nodes to 500 nodes, the simulated network area is also varied from a 548 meter square to a 2.2 Km square to keep the node density constant.

Table 1 Simulation Parameters

Parameter	Typical Value	Range
Mobility	36 Km/h	3.6– 72 Km/h
Group Size	1 source, 5 destinations	1 - 40 destinations
Network Size	100 nodes	500 nodes

C.3 Performance Metrics

To evaluate the performance of the proposed protocol, the following metrics are defined:

Communication Overhead: It is defined as the total number of control packets transmitted during the simulation. For control packets sent over multiple hops, each transmission of the control packet counts as one transmission.

Multicast efficiency: It is defined as the ratio of the total number of multicast packets received by all receivers to the total number of multicast packets transmitted during the simulation. Note that each time a node forwards a multicast packet, it is counted toward the total number of multicast packets transmitted.

The communication overhead is an important performance measure since control packets do consume network bandwidth and node battery power. In addition, control packets may delay the transmission of multicast packets since they have higher priority than multicast packets. The multicast efficiency shows how efficiently the multicast routing protocol uses network resources. High multicast efficiency is achieved when the protocol uses minimum number of forwarding nodes from a multicast source to destinations.

D. NUMERICAL RESULTS

D.1 Impact of TTL and Route Optimization

Figure 5 and Figure 6 show the impact of TTL on the communication overhead and the multicast efficiency, respectively. Recall that TTL (time-to-live) is the maximum number of hops that a packet is allowed to traverse, and it is used in the proposed protocol to limit the scope of route search. The proposed multicast routing protocol with the TTL value of 2 and of 3, as well as the proposed protocol with the adaptive TTL adjustment mechanism, are simulated. In the adaptive TTL adjustment mechanism, TTL is initially set to 2, and whenever a route search fails, TTL is incremented by 1, and a route search is performed again. This mechanism is labeled as Adaptive in Figure 5 and Figure 6.

For local-flooding scheme, TTL=2 gives the best performance (i.e., the smallest communication overhead the highest multicast efficiency). This is because larger TTL values create more number of multicast-route-recovery packets (which are considerably larger than the control packets used in the local-rejoin scheme), and this increases network congestion. For local-rejoin scheme, the adaptive TTL adjustment mechanism gives the best performance as expected.

Figure 7 and Figure 8 show the impact of route optimization (described in Section B.4) on the communication overhead and the multicast efficiency, respectively. In these figures, TTL is set to 2 for the local-flooding scheme, and the adaptive TTL adjustment mechanism is used for the local-rejoin scheme since they give the best performance. "Optimization-ON" refers to the case where a route optimization process is used, and "Optimization-OFF" refers to the case where a route optimization process is not used. These figures show that with route optimization, the communication overhead decreases, and the multicast efficiency increases in both local-flooding and local-rejoin schemes. As mentioned in Section B.4, the route optimization process detects the forwarding nodes which create redundant and inefficient routes and makes them into non-forwarding nodes. This results in shorter multicast paths. Shorter paths are less likely to break than longer ones, since they consist of less number of forwarding nodes, which can potentially move. Therefore, the route recovery process is less often invoked with router optimization, and this decreases the communication overhead. The multicast efficiency increases, since route optimization reduces the number of transmissions by reducing the number of forwarding nodes.

D.2 Performance Comparison

In this section, performance of the proposed protocol is compared to that of existing multicast routing protocols by varying a number of parameters such as the node mobility, the multicast group size and the network size. Note that when the network size is varied in simulations, the simulated network area is also varied to keep the node density constant. In all the figures presented in this section, the route optimization process is employed for the proposed protocol since it is shown to be effective in Section D.1. TTL=2 is used for the local-flooding scheme, and the adaptive TTL adjustment mechanism is used for the local-rejoin scheme. Multicast routing protocols compared to the proposed protocol include FGMP (Forwarding Group Multicast Protocol) -SA (Source Advertising), FGMP-RA (Receiver Advertising) and a simple flooding.

The following section D.2.1 describes how the communication overhead and the multicast efficiency of FGMP-SA, FGMP-RA and a simple flooding are estimated. The communication overhead and the multicast efficiency of the proposed protocol are measured through simulation. Section D.2.2 presents comparison of the communication overhead, and Section D.2.3 presents comparison of the multicast efficiency.

D.2.1 Performance Estimation of FGMP-SA, FGMP-RA and Simple Flooding

In FGMP-SA and FGMP-RA, DSDV or AODV is used as the underlying unicast protocol to establish the shortest paths between the source and receivers. It is assumed that the source generates a multicast packet every 100 ms. (Note that this is the same assumption made in the simulation of the proposed protocol (Section C.2).) Therefore, the shortest path between the source and each receiver is calculated every 100 ms. When the shortest path between the source and each receiver is calculated, the nodes along the shortest path are chosen as forwarding nodes.

The communication overhead of FGMP-RA is caused by periodical transmission of two types of control packets, receiver advertisement packets and packets containing forwarding table. In FGMP-RA, each receiver periodically floods a receiver advertisement packet, and each node forwards it once during the interval of receiver advertisement. Therefore, the total number of receiver advertisement transmissions, *Ara*, is given by

$$Ara = N \times R \times \frac{Tsim}{Tra} \quad (1)$$

where *Tsim* is the total simulation time; *Tra* is the interval of receiver advertisement; *N* is the total number of nodes in the network; and *R* is the number of receivers.

In FGMP-RA, since a forwarding table is forwarded along the shortest path to each receiver and the nodes along the shortest path are chosen as forwarding nodes, the number of forwarding table transmission is equal to the number of forwarding nodes. Therefore, the total number of forwarding table transmissions, *FTra*, is given by

$$FTra = \sum_j FN_j \quad (2)$$

where FN_j is the number of forwarding nodes at the time of $j \times Tft$ ms; *Tft* is the interval of forwarding table transmission. From equations (1) and (2), the total communication overhead of FGMP-RA, *Cfgmp-ra*, is given by

$$Cfgmp-ra = Ara + FTra = N \times R \times \frac{Tsim}{Tra} + \sum_j FN_j \quad (3)$$

The communication overhead of FGMP-SA is estimated in a similar way. In FGMP-SA, the sender periodically floods a sender advertisement packet, and each node forwards it once during the interval of sender advertisement. Therefore, the total number of sender advertisement transmissions, *Asa*, is given by

$$Asa = N \times \frac{Tsim}{Tsa} \quad (4)$$

where *Tsa* is the interval of sender advertisement.

In FGMP-SA, each receiver periodically sends a joining table along the shortest path to the sender. Therefore, the total number of joining table transmission, *JTsa*, is given by

$$JTsa = \sum_i \sum_k H_{ki} \quad (5)$$

where H_{ki} is the hop count of the shortest path from the receiver *k* to the sender at the time of $i \times Tjt$; *Tjt* is the time interval of joining table transmission. From equations (4) and (5), the total communication overhead of FGMP-SA, *Cfgmp-sa*, is given by

$$Cfgmp-sa = Asa + JTsa = N \times \frac{Tsim}{Tsa} + \sum_i \sum_k H_{ki} \quad (6)$$

According to [9], the typical values of *Tsa*, *Tra*, *Tft*, and *Tjt* are 400, 400, 200 and 200 ms respectively, and thus, these values are used in our estimation.

The multicast efficiency of FGMP is estimated as follows. Note that FGMP-RA and FGMP-SA have the same multicast efficiency since the number of forwarding nodes is same in both protocols. Since each forwarding node transmits a given multicast packet once, the number of transmissions of a multicast packet is equal to the number of forwarding nodes. That is, the total number of multicast transmissions, *MT*, is given by

$$MT = \sum_i FN_i \quad (7)$$

where FN_i is the number of forwarding nodes at the time of *i* th multicast transmission at the sender (i.e., at the time $i \times 100$ ms). Since the total simulation time is 100 s and the source generates a multicast packet at each 100 ms, the total number of packets received by all receivers, P*received*, is given by

$$\mathrm{P}received = R \times \frac{100s}{100ms} = 1000R \quad (8)$$

assuming no packet loss. *R* is the number of receivers. The multicast efficiency of FGMP, *Mfgmp*, is, therefore, given by

$$Mfgmp = \frac{\mathrm{P}received}{MT} = \frac{1000R}{\sum_i FN_i} \quad (9)$$

Since a simple flooding scheme uses no control packet, its communication overhead, *Cfl*, is 0. Since each node transmits a given multicast packet once, the multicast efficiency of simple flooding, *Mfl*, is given by

$$Mfl = \frac{R}{N} \quad (10)$$

assuming no packet loss.

Note that, since no packet loss is assumed in our estimation, the estimated multicast efficiency of FGMP and simple flooding gives the upper bound (best case) of actual multicast efficiency. In Section D.2.3, it will be shown that the proposed protocol gives higher multicast efficiency than FGMP and simple flooding even when the upper bound is used for the multicast efficiency of FGMP and simple flooding.

D.2.2 Comparison of Communication Overhead

Figures 9, 10 and 11 show the communication overhead (the total number of control packet transmissions during the simulation) of the proposed protocol (both the local-flooding and local-rejoin schemes), FGMP-SA, FGMP-RA and simple flooding, when the node mobility, multicast group size and network size are varied, respectively. For the proposed protocol, control packets include JOIN, REPLY, RESERVE, PRUNE and multicast-route-recovery packets. For FGMP-SA, control packets include sender advertisement packets and packets containing joining table, whereas for FGMP-RA, they include receiver advertisement packets and packets containing forwarding tables. The simple flooding scheme does not use any control packets. The communication overhead of FGMP-SA and FGMP-RA is estimated as described in Section D.2.1. (To estimate FN_j in Eq.(3), the shortest paths from the multicast source to receivers are calculated every 100ms using simple flooding. The nodes along the shortest paths are then counted as forwarding nodes. H_{ki} in Eq.(6) is estimated in the same way.) The communication overhead of the simple flooding scheme is 0 since no control packet is used in simple flooding.

Note that in our simulation, multicast-route-recovery packets are counted in calculation of both communication overhead and multicast efficiency. Recall that multicast-route-recovery packets are special packets flooded in a local-flooding scheme, and they contain the original multicast packets. Ideally, multicast-route-recovery packets that reach the downstream node should be counted in multicast efficiency and others in communication overhead. Since multicast-route-recovery packets are counted as both control packets (in calculating communication overhead) and multicast packets (in calculating multicast efficiency) in our simulation, the actual communication overhead (multicast efficiency) of the local-flooding scheme would be lower (higher) than the values presented in Section D.2.2 (D.2.3).

Figure 9 shows the impact of the node mobility on the communication overhead. In this figure, it is shown that the communication overhead of the proposed protocol (both local-flooding and local-rejoin schemes) increases as the node mobility increases. This is because as the node mobility increases, a link breakage occurs more often, and thus, a route recovery process is more frequently invoked. The communication overhead of FGMP-SA and FGMP-RA is constant, because their overhead depends on the time intervals *Tsa* (flooding interval of sender advertisements), *Tra* (flooding interval of receiver advertisements), *Tft* (transmission interval of forwarding tables) and *Tjt* (transmission interval of joining tables), and these parameters are assumed to be independent of the node mobility in our estimation. In reality, *Tft* and *Tjt* become shorter as the node mobility increases, and thus, the actual communication overhead of FGMP-SA and FGMP-RA would be even higher than the values obtained through our estimation. It is shown in this figure that the communication overhead of the proposed protocol is much smaller than that of FGMP-SA and FGMP-RA in all node mobility range.

Figure 10 shows the impact of the multicast group size on the communication overhead. It is shown in this figure that the communication overhead increases as the multicast group size increases in the proposed protocol and FGMP. Especially in FGMP-RA, the communication overhead increases very rapidly as the multicast group size increases. This is because in FGMP-RA, each receiver periodically floods receiver advertisement to the entire network. This figure also shows that the communication overhead of the proposed protocol is smaller than that of FGMP. The communication overhead of the local-flooding scheme is, however, relatively large (compared to that of the local–rejoin scheme) at large multicast group sizes. This is because of the following. With a larger multicast group size, the number of forwarding nodes is also larger. This decreases the available bandwidth of the network since each forwarding node consumes bandwidth by forwarding multicast packets. Therefore, when multicast-route-recovery packets are flooded to recover from a broken route, network congestion may occur, and consequently, multicast-route-recovery packets may be lost or delayed. As a result, receivers may time out and start a join process, and this increases the number of control packet transmissions.

Figure 11 shows the impact of the network size on the communication overhead. The communication overhead of each protocol investigated increases as the network size increases. The increase in the communication overhead is significant especially with FGMP-SA and FGMP-RA. Periodical flooding of sender/receiver advertisements of FGMP accounts for this large communication overhead.

In this section, it is shown that the communication overhead of the proposed protocol is much smaller than that of FGMP. This is because unlike FGMP, the proposed protocol does not flood control packets periodically. The proposed protocol floods JOIN packets only in the route setup process, and the route setup process is invoked only when necessary. As mentioned earlier, the communication overhead of simple flooding is 0. However, simple flooding has very low multicast efficiency, and this is shown in the following Section D.2.3.

D.2.3 Comparison of Multicast Efficiency

Figures 12, 13 and 14 show the multicast efficiency (the ratio of the total number of multicast packets received by receivers to the total number of multicast packets transmitted within a network) of the proposed protocol, FGMP and a simple flooding scheme, when the node mobility, multicast group size and network size are varied, respectively. As explained in Section D.2.1, FGMP-RA and FGMP-SA have the same multicast efficiency, since the number of forwarding nodes is same in both protocols. Therefore, in Figure 12 through Figure 14, the multicast efficiency of FGMP-RA and FGMP-SA is noted as FGMP. Note that as explained in Section D.2.1, for the multicast efficiency of FGMP and a simple flooding scheme, values shown in these figures are the best case values.

Figure 12 shows the impact of the node mobility on the multicast efficiency. The multicast efficiencies of the local-rejoin scheme and FGMP remain almost constant when the node mobility is varied. This is because the multicast efficiency of the local-rejoin scheme and FGMP depends on the number of forwarding nodes, which stays almost constant as the node mobility is varied. The multicast efficiency of the simple flooding scheme also remain constant, since it depends on the number of receivers and the number of nodes in the network (refer to equation (10) in Section D.2.1), and these numbers remain constant as a node moves. The multicast efficiency of the local-flooding scheme, on the other hand, slightly decreases as the node mobility increases. This is because the route recovery process is more frequently invoked as the node mobility increases, and the multicast-route-recovery packets used in the route recovery process are counted in the number of multicast packets transmitted. In this figure, it is shown that the multicast efficiency of the local-rejoin scheme is almost same as that of FGMP and that the simple flooding scheme has the lowest multicast efficiency.

Figure 13 shows the impact of the multicast group size on the multicast efficiency. As expected, the multicast efficiency of each protocol increases as the multicast group size increases, and the simple flooding scheme has the lowest multicast efficiency. The proposed protocol (both the local-flooding scheme and the local-rejoin scheme) has higher multicast efficiency than FGMP, unless the multicast group size (i.e., the number of receivers) is very small. To further investigate the impact of the multicast group size on the multicast efficiency, we obtained the average number of forwarding nodes in cases of 5 receivers and 40 receivers in a multicast group. Simulation results show that the average numbers of forwarding nodes for the local-flooding scheme, local-rejoin scheme and FGMP are 11.6, 9.6 and 9.6, respectively, for the case of 5 receivers in a multicast group. For the case of 40 receivers, they are 19.0, 18.6 and 28.6, respectively. These results show that the proposed protocol delivers multicast packets to receivers with the less number of forwarding nodes than FGMP, especially at large multicast group sizes. This contributes to the high multicast efficiency of the proposed protocol.

Figure 14 shows the impact of the network size on the multicast efficiency. The multicast efficiency of each protocol investigated decreases as the network size increases. This is because as the network size increases, the average hop count of the paths from the source to receivers increases, and thus, the number of forwarding nodes increases. Therefore, the multicast efficiency decreases. In this figure, it is shown that the local-rejoin scheme has almost the same multicast efficiency as FGMP. Again, the simple flooding scheme has the lowest multicast efficiency.

In this section, it is shown that the multicast efficiency of the proposed protocol is same or higher than that of FGMP. Note that in Section D.2.2, it is shown that the communication overhead of the proposed protocol is significantly smaller than that of FGMP. Therefore, it is concluded that the proposed protocol achieves high multicast efficiency with low communication overhead, compared to other existing multicast routing protocols. Since the simple flooding scheme has very low multicast efficiency, it is not suitable for ad-hoc networks where bandwidth is very scarce.

E. CONCLUSION

In this paper, a bandwidth-efficient multicast routing protocol is proposed for ad-hoc wireless networks. Through simulations, the performance of the proposed protocol was investigated and compared with that of other existing multicast routing protocols, such as FGMP-SA, FGMP-RA and a simple flooding scheme. It was shown that the proposed protocol achieves high multicast efficiency with low communication overhead.

REFERENCES

[1] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad-hoc wireless networks. Mobile Computing, chapter 5, Kluwer Academic Publishers, 1996.

[2] C-K Toh. Wireless ATM and Ad-Hoc Networks: Protocols and Architectures. Kluwer Academic Publishers, 1997.

[3] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers, SIGCOMM Conf. Proc, 1994.

[4] S. Corson and A. Emphremides. A Distributed Routing Algorithm for Mobile Wireless Networks, ACM/Baltzer Wireless Networks J., vol. 1, no.1, 1995.

[5] S. Murthy and J. J. Garcia-Luna-Aceves. A Routing Protocol for Packet Radio Networks, MOBICOM, 1995.

[6] R. Dube et. Al. Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Network, IEEE Personal Communications, 1997.

[7] Z. J. Hass. A New Routing Protocol for the Reconfigurable Wireless Network, ICUPC, 1997.

[8] S. E. Deering and D. R. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. ACM Transaction on Computer Systems, May 1990.

[9] C-C. Chiang and M. Gerla. On-Demand Multicast in Mobile Wireless Networks. Proceedings of IEEE ICNP '98, 1998.

[10] C-C. Chiang and M. Gerla. Routing and Multicast in Multihop, Mobile Wireless Networks. Proceedings of ICUPC '97, 1997.

[11] C-C. Chiang, M. Gerla and L. Zhang. Shared Tree Wireless Network Multicast. Proceedings of IEEE 6th International Conference on Computer Communications and Networks (ICCCN'97), 1997.

[12] C. E. Perkins and P. Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. Proceedings of the SIGCOMM '94 Conference on Communications Architecture, Protocols and Applications, page 234-244, August 1994.

[13] C. E. Perkins. Ad-hoc On Demand Distance Vector (AODV) routing. Internet-Draft, draft-ietf-manet-aodv-00.txt, November 1997.

[14] T. Ballardie, P. Francis, and J. Crowcroft. Core Based Tree (CBT) an architecture for scalable interdomain multicast routing. Proceeding of ACM SIGCOM, 1993.

[15] E. M. Royer and C. E. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing . draft-ietf.manet-maodv-00.txt, July 2000.

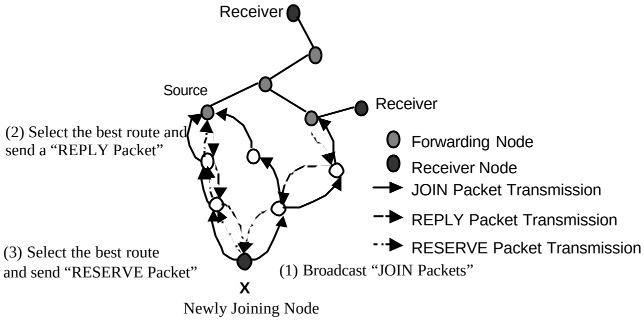

Receiver
Source
Receiver
(2) Select the best route and send a "REPLY Packet"
Forwarding Node
Receiver Node
JOIN Packet Transmission
REPLY Packet Transmission
RESERVE Packet Transmission
(3) Select the best route and send "RESERVE Packet"
X
(1) Broadcast "JOIN Packets"
Newly Joining Node

Figure 1. Route setup

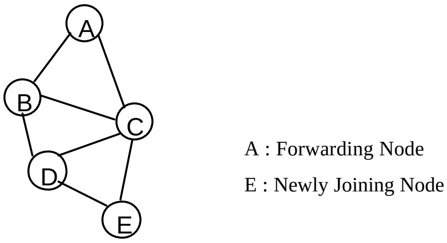
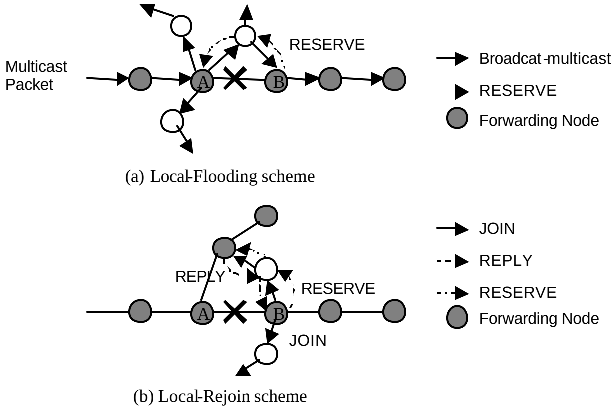


Figure 2.An Example of Network Topology



(a) Local-Flooding scheme

(b) Local-Rejoin scheme

Figure 3. Route recovery

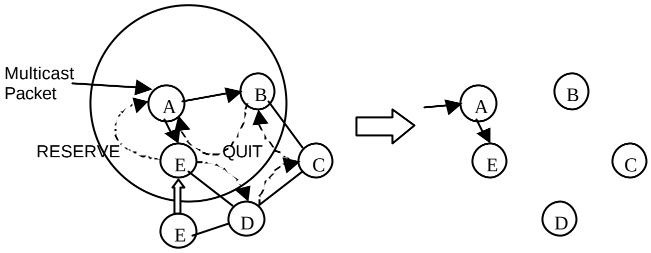


Figure 4. Route optimization

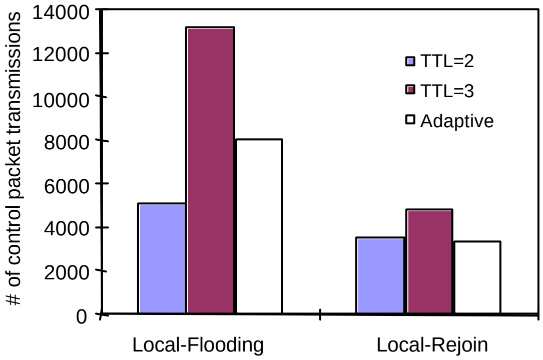


Figure 5. Impact of TTL on communication overhead

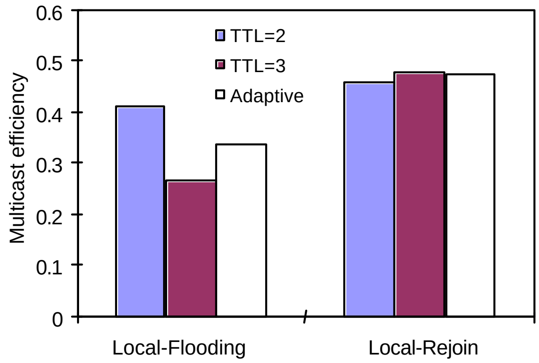


Figure 6. Impact of TTL on multicast efficiency

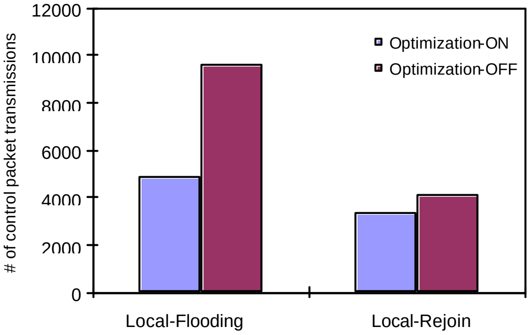


Figure 7. Impact of route optimization on communication overhead

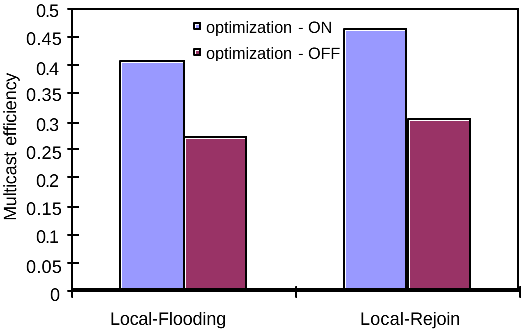


Figure 8. Impact of route optimization on multicast efficiency

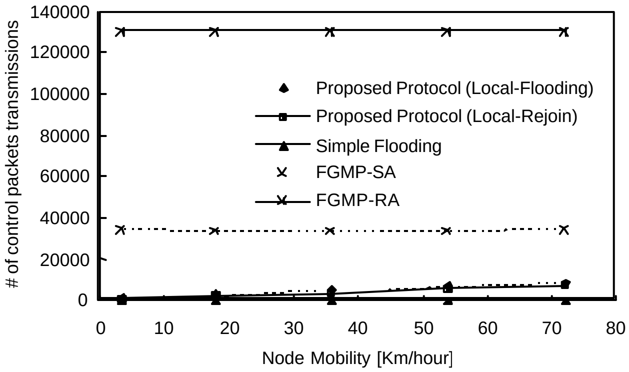


Figure 9. Communication overhead vs. mobility

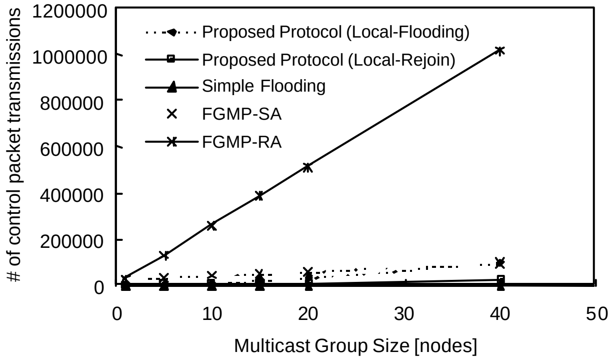


Figure 10. Communication overhead vs. multicast group size

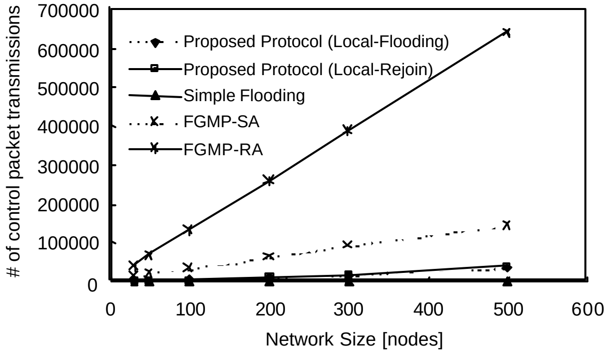


Figure 11. Communication overhead vs. network size

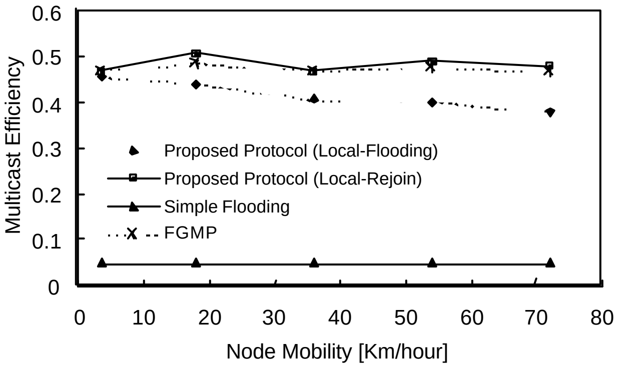


Figure 12. Multicast efficiency vs. mobility

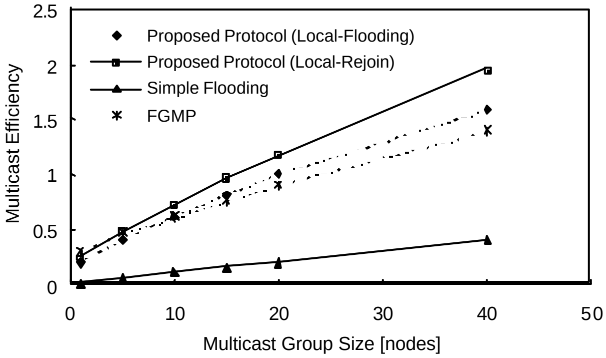


Figure 13. Multicast efficiency vs. multicast group size

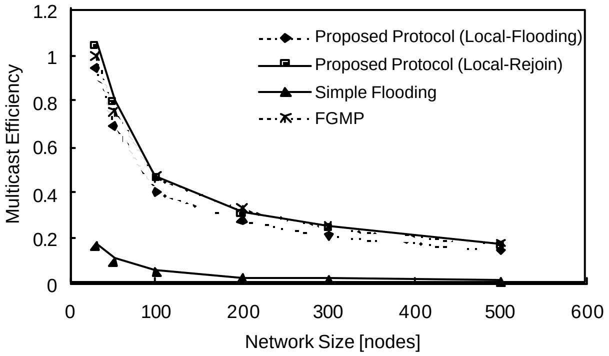


Figure 14. Multicast efficiency vs. network size