IETF MANET Working Group
INTERNET-DRAFT
Expiration: July 2000

Sung-Ju Lee
William Su
Mario Gerla
University of California, Los Angeles

January 2000

On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks

<draft-ietf-manet-odmrp-02.txt>

Status of This Memo

Abstract

On-Demand Multicast Routing Protocol (ODMRP) is a multicast routing protocol designed for ad hoc networks with mobile hosts. ODMRP is a mesh-based, rather than a conventional tree-based, multicast scheme and uses a forwarding group concept (only a subset of nodes forwards the multicast packets via scoped flooding). It applies on-demand procedures to dynamically build routes and maintain multicast group membership. ODMRP is well suited for ad hoc wireless networks with mobile hosts where bandwidth is limited, topology changes frequently and rapidly, and power is constrained.

Contents

1. Introduction

This document describes the On-Demand Multicast Routing Protocol (ODMRP) [14][15] developed by the Wireless Adaptive Mobility (WAM) Laboratory [20] at University of California, Los Angeles. ODMRP applies "on-demand" routing techniques to avoid channel overhead and improve scalability. It uses the concept of "forwarding group," [5] a set of nodes responsible for forwarding multicast data, to build a forwarding mesh for each multicast group. By maintaining and using a mesh instead of a tree, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. A soft-state approach is taken to maintain multicast group members. No explicit control message is required to leave the group. We believe the reduction of channel/storage overhead and the relaxed connectivity make ODMRP more attractive in mobile wireless networks.

The following properties of ODMRP highlight its advantages.

* Simplicity

* Low channel and storage overhead

* Usage of up-to-date shortest routes

* Reliable construction of routes and forwarding group

* Robustness to host mobility

* Maintenance and exploitation of multiple redundant paths

* Exploitation of the broadcast nature of wireless environments

* Unicast routing capability

2. Terminology

2.1. General Terms

This section defines terminology used in ODMRP.

node

A device that implements IP.

neighbor

Nodes that are within the radio transmission range.

forwarding group

A group of nodes participating in multicast packet forwarding.

multicast mesh

The topology defined by the link connection between forwarding group members.

join query

The special data packet sent by multicast sources to establish and update group memberships and routes.

join reply

The table broadcasted by each multicast receiver and forwarding node to establish and update group membership and routes

2.2. Specification Language

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [4].

3. Protocol Overview

3.1. Multicast Route and Mesh Creation

In ODMRP, group membership and multicast routes are established and updated by the source on demand. Similar to on-demand unicast routing protocols, a request phase and a reply phase comprise the protocol. When a multicast source has packets to send but no route and group membership is known, it floods a member advertising packet with data payload piggybacked. This packet, called "Join Query" (format shown in Section 4.1) is periodically broadcasted to the entire network to refresh the membership information and update the routes. When a node receives a Join Query packet, it stores the source address and the unique identifier of the packet to its "Message Cache" to detect duplicates. The upstream node address is inserted or updated as the next node for the source node in its "Routing Table." If the Join Query packet is not a duplicate and the Time-To-Live value is greater than zero, appropriate fields are updated and it is rebroadcast (operation details are illustrated in Section 5.1.2).

When a Join Query packet reaches the multicast receiver, it creates and broadcasts a "Join Reply" to its neighbors. When a node receives a Join Reply, it checks if the next node address of one of the entries matches its own address. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group; it sets the FG_FLAG (Forwarding Group Flag). It then broadcasts its own Join Reply built upon matched entries. The next node address field is filled in by extracting the information from its routing table. This way, the Join Reply is propagated by each forward group member until it reaches the multicast source via the selected path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the forwarding group.

```
   +--+       +--+       +--+
   |S1|-------|I1|-------|R1|
   +--+\      +--+      /+--+  Join Replies of Node R1 and Node I1
        \              /        +----------------+  +----------------+
         \            /         |Sender|Next Node|  |Sender|Next Node|
          \          /          |------+---------|  |------+---------|
           \        /           |  S1  |    I1   |  |  S1  |    S1   |
            \      /            |------+---------|  +----------------+
   +--+      \+--+/      +--+   |  S2  |    I2   |
   |S2|-------|I2|-------|R2|   +----------------+
   +--+       +--+       +--+
```

Let us consider the above figure as an example of Join Reply forwarding process. Nodes S1 and S2 are multicast sources, and nodes R1 and R2 are multicast receivers. Node R2 sends its Join Reply to both S1 and S2 via I2, and R1 sends its packet to S1 via I1 and to S2 via I2. When receivers send their Join Replies to next hop nodes, an intermediate node I1 sets the FG_FLAG and builds its own Join Reply since there is a next node ID entry in the Join Reply received from R1 that matches its ID. Note that the Join Reply built by I1 has an entry for sender S1 but not for S2 because the next node address for S2 in the received Join Reply is not I1. In the meanwhile, node I2 sets the FG_FLAG, constructs its own Join Reply and sends it to its neighbors. Note that I2 broadcasts the Join Reply only once even though it receives two Join Replies from the receivers because the second table arrival carries no new source information. Channel overhead is thus reduced dramatically in cases where numerous multicast receivers share the same links to the source.

After this group establishment and route construction process, a source can multicast packets to receivers via selected routes and forwarding groups. While outgoing data packets exist, the source sends Join Query every REFRESH_INTERVAL. This Join Query and Join Reply propagation process refreshes forwarding group and routes. When receiving the multicast data packet, a node forwards it only when it is not a duplicate and the setting of the FG_FLAG for the multicast group has not expired. This procedure minimizes the traffic overhead and prevents sending packets through stale routes.

3.2. Reliability

The reliable transmission of Join Replies plays an important role in establishing and refreshing multicast routes and forwarding groups. Hence, if Join Replies are not properly delivered, effective multicast routing cannot be achieved by ODMRP. The IEEE 802.11 MAC (Medium Access Control) protocol [8], which is the emerging standard in wireless networks, performs reliable transmission by retransmitting the packet if no acknowledgment is received. However, if the packet is broadcasted, no acknowledgments or retransmissions are sent. In ODMRP, the transmission of Join Replies are often broadcasted to more than one upstream neighbors since we are handling multiple sources (e.g., see the Join Reply from node R1 in the example of Section 3.1.). In such cases, the hop-by-hop verification of Join Reply delivery and the retransmission cannot be handled by the MAC layer. It must be done indirectly by ODMRP. Another option for reliable delivery is to subdivide the Join Reply into separate sub-tables, one for each distinct next node. In the figure of Section 3.1. for example, the Join Reply at node R1 is split into two Join Replies, one for neighbor I1 and the other for neighbor I2. These Join Replies are separately unicasted using a reliable MAC protocol such as IEEE 802.11 or MACAW [3]. Since the number of neighbors is generally limited (typically, about six neighbors in the optimum in a multihop network [12]), the scheme still scales well to large number of sources. This option can actually be used as backup to the passive acknowledgment option as discussed below.

We adopt a scheme that was used in [10]. When a node transmits a Join Reply packet to the immediate upstream node of the route, the immediate downstream node can hear the transmission if it is within the transmitter's radio range. Hence, the packet is used as an "passive acknowledgment." We can utilize this passive acknowledgment to verify the delivery of a Join Reply. Note that the source itself must send an active acknowledgment to the previous hop since it does not have any next hop to send a Join Reply to unless it is also a forwarding group node for other sources.

Considering the case in figure of Section 3.1. again, we note that once the nodes I1 and I2 receive the Join Reply from node R1, they will construct and forward their own Join Replies to next hops (in this case, sources S1 and S2). In transmitting their Join Replies, nodes I1 and I2 may overlap with each other. If I1 and I2} are within receiving range, they will recover because of the carrier sense feature in CSMA (Carrier Sense Multiple Access) [13]. However, if they are out of range, they will be unaware of the "hidden terminal" condition of node R1, which cannot hear the (overlapped) passive acknowledgments. Thus, a node may not hear the passive acknowledgments of its upstream neighbor because of conflicts due to the hidden terminal problem. It will also not hear the passive acknowledgment if the upstream neighbor has moved away. In either case, when no acknowledgment is received within the timeout interval, the node retransmits the message. Note that the node may get acknowledgments from some, but not all upstream neighbors. As an option, the retransmission could be carried out in unicast mode, to selected neighbors, with reduced sub-tables. If packet delivery cannot be verified after an appropriate number of retransmissions, the node considers the route to be invalidated. At this point, the most likely cause of route failure is the fact that a node on the route has failed or has moved out of range. An alternate route must be found "on the spot." The node thus broadcasts a message to its neighbors specifying that the next hop to a set of sources cannot be reached. Upon receiving this packet, each neighbor builds and unicasts the Join Reply to its next hop if it has a route to the multicast sources. If no route is known, it simply broadcasts the packet specifying the next hop is not available. In both cases, the node sets its FG_FLAG. In practical implementations, this redundancy is sufficient to establish alternate paths until a more efficient route is established during the next refresh phase. The FG_FLAG setting of every neighbor may create excessive redundancy, but most of these settings will expire because only necessary forwarding group nodes will be refreshed in the next Join Reply propagation phase.

3.3. Soft State

In ODMRP, no explicit control packets need to be sent to leave the group. If a multicast source wants to leave the group, it simply stops sending any Join Query packets since it does not have any multicast data to send to the group. If a receiver no longer wants to receive from a particular multicast group, it does not send the Join Reply for that group. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no Join Replies received) before they timeout.

3.4. Selection of Timer Values

Timer values for route refresh interval and forwarding group timeout interval can have impacts on ODMRP performance. The selection of these soft state timers should be adaptive to network environment (e.g., traffic type, traffic load, mobility pattern, mobility speed, channel capacity, etc.). When small route refresh interval values are used, fresh route and membership information can be obtained frequently at the expense of producing more packets and causing network congestion. On the other hand, when large route refresh values are selected, even though less control traffic will be generated, nodes may not know up-to-date route and multicast membership. Thus in highly mobile networks, using large route refresh interval values can yield poor protocol performance. The forwarding group timeout interval should also be carefully selected. In networks with heavy traffic load, small values should be used so that unnecessary nodes can timeout quickly and not create excessive redundancy. In situations with high mobility, however, large values should be chosen so that more alternative paths can be provided. It is important to note that the forwarding group timeout value must be larger (e.g., 3 to 5 times) than the value of route refresh interval.

3.5. Unicast Capability

One of the major strengths of ODMRP is its unicast routing capability. Not only can ODMRP coexist with any unicast routing protocol, it can also operate very efficiently as an unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol. Other ad hoc multicast routing protocols such as AMRoute [5], CAMP [7], RBM [6], and LAM [9] must be run on top of a unicast routing protocol. CAMP, RBM, and LAM in particular, only work with certain underlying unicast protocols.

3.6. Contents of Tables

Nodes running ODMRP are required to maintain the following tables. These tables MAY be implemented in any format, but MUST include the fields specified in this document.

3.6.1. Routing Table

A routing table is created on demand and is maintained by each node. An entry is inserted or updated when a non-duplicate Join Query is received. The node stores the destination (i.e., the source of the Join Query) and the next hop to the destination (i.e., the last node that propagated the Join Query). The routing table provides the next hop information when transmitting Join Replies.

3.6.2. Forwarding Group Table

When a node is a forwarding group node of the multicast group, it maintains the group information in the forwarding group table. The multicast group ID and the time when the node was last refreshed are recorded.

3.6.3. Message Cache

The message cache is maintained by each node to detect duplicates. When a node receives a new Join Query or data, it stores the source address and the unique identifier of the packet. Note that entries in the message cache need not be maintained permanently. Schemes such as LRU (Least Recently Used) or FIFO (First In First Out) can be employed to expire and remove old entries and prevent the size of the message cache to be extensive.

3.7. Mobility Prediction

3.7.1 Adapting the Refresh Interval via Mobility Prediction

ODMRP requires periodic flooding of Join Query to build and refresh routes. Excessive flooding, however, is not desirable in ad hoc networks because of bandwidth constraints. Furthermore, flooding often causes congestion, contention, and collisions. Finding the optimal flooding interval is critical in ODMRP performance. In highly mobile networks where nodes are equipped with GPS [11] (e.g., tactical networks with tanks, ships, aircrafts, etc.), we can efficiently adapt the REFRESH_INTERVAL to mobility patterns and speeds by utilizing the location and movement information. We use the location and movement information to predict the duration of time routes will remain valid. With the predicted time of route disconnection, Join Queries are only flooded when route breaks of ongoing data sessions are imminent. Note that ODMRP can still operate efficiently in networks where no such information is available, but the protocol can be further improved if those information can be utilized.

In our prediction method, we assume a free space propagation model, where the received signal strength solely depends on its distance to the transmitter. We also assume that all nodes in the network have their clock synchronized (e.g., by using the NTP (Network Time Protocol) [16] or the GPS clock itself). Therefore, if the motion parameters of two neighbors (e.g., speed, direction, radio propagation range, etc.) are known, we can determine the duration of time these two nodes will remain connected. Assume two nodes i and j are within the transmission range r of each other. Let (x_{i}, y_{i}) be the coordinate of node i and (x_{j}, y_{j}) be that of node j. Also let v_{i} and v_{j} be the speeds, and theta_{i} and theta_{j} (0 <= theta_{i}, theta_{j} < 2 * pi) be the moving directions of nodes i and j, respectively. Then, the duration of time that the link between two nodes will stay connected, D_{t}, is given by:

```
        -(a*b + c*d) + sqrt((a^{2} + c^{2})*r^{2} - (a*d - b*c)^{2})
D_{t} = ------------------------------------------------------------
                          a^{2} + c^{2}

   where
      a = v_{i}*cos(theta_{i}) - v_{j}*cos(theta_{j}),
      b = x_{i} - x_{j},
      c = v_{i}*sin(theta_{i}) - v_{j}*sin(theta_{j}), and
      d = y_{i} - y_{j}.
```

Note that when v_{i} = v_{j} and theta_{i} = theta_{j}, D_{t} is set to infinity without applying the above equation.

To utilize the information obtained from the prediction, extra fields must be added into Join Query and Join Reply packets. When a source sends Join Query, it appends its location, speed, and direction. It sets the MIN_LET (Minimum Link Expiration Time) field to the MAX_LET_VALUE since the source does not have any previous hop node. The next hop neighbor, upon receiving a Join Query, predicts the link expiration time between itself and the previous hop using the above equation. The minimum between this value and the MIN_LET indicated by the Join Query is included in the packet. The rationale is that as soon as a single link on a path is disconnected, the entire path is invalidated. The node also overwrites the location and mobility information field written by the previous node with its own information. When a multicast member receives the Join Query, it calculates the predicted LET of the last link of the path. The minimum between the last link expiration time and the MIN_LET value specified in the Join Query is the RET (Route Expiration Time). This RET value is enclosed in the Join Reply and broadcasted. If a forwarding group node receives multiple Join Replies with different RET values (i.e., lies in paths from the same source to multiple receivers), it selects the minimum RET among them and sends its own Join Reply with the chosen RET value attached. When the source receives Join Replies, it selects the minimum RET among all the Join Replies received. Then the source can build new routes by flooding a Join Query before the minimum RET approaches (i.e., route breaks).

In addition to the estimated RET value, other factors need to be considered when choosing the refresh interval. If the node mobility rate is high and the topology changes frequently, routes will expire quickly and often. The source may propagate Join Query excessively and this excessive flooding can cause collisions and congestion, and clogs the network with control packets. Thus, the MIN_REFRESH_INTERVAL should be enforced to avoid control message overflow. On the other hand, if nodes are stationary or move slowly and link connectivity remains unchanged for a long duration of time, routes will hardly expire and the source will rarely send Join Query. A few problems arise in this situation. First, if a node in the route suddenly changes its movement direction or speed, the predicted RET value becomes obsolete and routes will not be reconstructed in time. Second, when a non-member node which is located remotely to multicast members wants to join the group, it cannot inform the new membership or receive data until a Join Query is received. Hence, the MAX_REFRESH_INTERVAL should be set. The selection of the MIN_REFRESH_INTERVAL and the MAX_REFRESH_INTERVAL values should be adaptive to network environments.

3.7.2. Route Selection Criteria

In ODMRP, a multicast receiver selects routes based on the minimum delay (i.e., routes taken by the first Join Query received. A different route selection method is applied when we use the mobility prediction. The idea is inspired by the Associativity-Based Routing (ABR) protocol [18] which chooses associatively stable routes. In our algorithm, instead of using the minimum delay path, we can choose a route that is the most stable (i.e., the one that will remain connected for the longest duration of time). To select a route, a multicast receiver must wait for an appropriate amount of time after receiving the first Join Query so that all possible routes and their route qualities will be known. The receiver then chooses the most stable route and broadcasts a Join Reply. Route breaks will occur less often and the number of Join Query propagation will reduce because stable routes are used.

3.7.3. Alternative Method of Prediction

Since GPS may not work properly in certain situations (e.g., indoor, fading, etc.), we may not always be able to accurately predict the link expiration time for a particular link. However, there is an alternative method to predict the LET. This method is based on a more realistic propagation model and has been proposed in [1] and [17]. Basically, transmission power samples are measured periodically from packets received from a mobile's neighbor. From this information it is possible to compute the rate of change for a particular neighbor's transmission power level. Therefore, the time when the transmission power level will drop below the acceptable value (i.e., hysteresis region) can be predicted. We plan to investigate this option in our future work.

4. Packet and Table Formats

4.1. Join Query Packet Header

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Type     |   Reserved    |  Time To Live |   Hop Count   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Multicast Group IP Address                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       Sequence Number                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Source IP Address                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Previous Hop IP Address                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Previous Hop X Coordinate                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Previous Hop Y Coordinate                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Previous Hop Moving Speed   | Previous Hop Moving Direction |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Minimum Link Expiration Time                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

01; ODMRP Join Query.

Reserved

Sent as 0; ignored on reception.

Time To Live

Number of hops this packet can traverse.

Hop Count

The number of hops traveled so far by this packet.

Multicast Group IP Address

The IP address of the multicast group.

Sequence Number

The sequence number assigned by the source to uniquely identify the packet.

Source IP Address

The IP address of the node originating the packet.

Previous Hop IP Address

The IP address of the last node that has processed this packet.

Previous Hop X Coordinate (Optional)

The x-coordinate of the last node that has processed this packet. The information can be obtained from the GPS. This field is required only when network hosts are GPS equipped.

Previous Hop Y Coordinate (Optional)

The y-coordinate of the last node that has processed this packet. The information can be obtained from the GPS. This field is required only when network hosts are GPS equipped..

Previous Hop Moving Speed (Optional)

The mobility speed of the last node that has processed this packet. The information can be obtained from the GPS or the node's own instruments and sensors (e.g., campus, odometer, speed sensors, etc.). This field is required only when network hosts are GPS equipped.

Previous Hop Moving Direction (Optional)

The moving direction of the last node that has processed this packet. The information can be obtained from the GPS or the node's own instruments and sensors (e.g., campus, odometer, speed sensors, etc.). This field is required only when network hosts are GPS equipped.

Minimum Link Expiration Time (Optional)

The minimum expiration time among the links taken by this packet so far. This field is required only when network hosts are GPS equipped.

4.2. Join Reply Packet

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Type     |    Count      |R|F|     Reserved              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Multicast Group IP Address                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Previous Hop IP Address                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Sequence Number                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Sender IP Address [1]                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Next Hop IP Address [1]                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Route Expiration Time [1]                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               :                               |
    |                               :                               |
    |                               :                               |
    |                               :                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                    Sender IP Address [n]                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Next Hop IP Address [n]                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Route Expiration Time [n]                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type

02; ODMRP Join Reply.

Count

Number of (Sender IP Address, Next Hop IP Address) combinations.

R

Acknowledgment request flag. This flag is set when active acknowledgment packet is requested.

F

Forwarding group flag. This flag is set when the packet is transmitted by a forwarding group node.

Reserved

Sent as 0; ignored on reception.

Multicast Group IP address

The IP address of the multicast group.

Previous Hop IP Address

The IP address of the last node that has processed this packet.

Sequence Number

The sequence number assigned by the previous hop node to uniquely identify the packet.

Sender IP Address [1..n]

The IP addresses of the sources of this multicast group.

Next Hop IP Address [1..n]

The IP addresses of next nodes that this packet is target to.

Route Expiration Time [1..n] (Optional)

The minimum route expiration times of this multicast group. This field is required only when network hosts are GPS equipped.

5. Operation

5.1. Forwarding Group Setup

5.1.1. Originating a Join Query

When a multicast source has data packets to send but no route is known, it originates a "Join Query" packet. The Type field MUST be set to 01. TTL MAY be set to TIME_TO_LIVE_VALUE, but SHOULD be adjusted based on network size and network diameter. The Sequence Number MUST be large enough to prevent wraparound ambiguity, and the Hop Count is initially set to zero. The source puts its IP address in the Source IP Address and Last Hop IP Address field. It appends its location, speed, and direction into Join Query if nodes in the network are equipped with GPS.

When location and movement information is utilized, it sets the MIN_LET (Link Expiration Time) field to the MAX_LET_VALUE since the source does not have any previous hop node. When the source receives Join Replies from multicast receivers, it selects the minimum RET (Route Expiration Time) among all the Join Replies received. Then the source can build new routes by originating a Join Query before the minimum RET approaches (i.e., route breaks of ongoing data sessions are imminent).

5.1.2. Processing a Join Query

When a node receives a Join Query packet:

1. Check if it is a duplicate by comparing the (Source IP Address, Sequence Number) combination with the entries in the message cache. If a duplicate, then discard the packet. DONE.

2. If it is not a duplicate, insert an entry into the message cache with the information of the received packet (i.e., sequence number and source IP address) and insert/update the entry for routing table (i.e., backward learning).

3. If the node is a member of the multicast group, it originates a Join Reply packet with the RET value enclosed (see Section 5.1.4).

4. Increase the Hop Count field by 1 and decrease the TTL field by 1.

5. If the TTL field value is less than or equal to 0, then discard the packet. DONE.

6. If the TTL field value is greater than 0, then set the node's IP Address into Last Hop IP Address field and broadcast. DONE.

5.1.3. Processing a Join Query When GPS is Used

When a node receives a Join Query packet:

1. Check if it is a duplicate by comparing the (Source IP Address, Sequence Number) combination with the entries in the message cache. If a duplicate, then discard the packet. DONE.

2. If it is not a duplicate, insert an entry into the message cache with the information of the received packet (i.e., sequence number and source IP address) and insert/update the entry for routing table (i.e., backward learning).

3. Predict the duration of time the link between the node and the upstream node will remain connected using the equation given in Section 3.7.1.

   The minimum between the newly obtained D_{t} value and the indicated value in MIN_LET field of the Join Query is included in the packet. The rationale is that as soon as a single link on the path is disconnected, the entire path is invalidated. The node also overwrites the location and mobility information field written by the previous node with its own information.

4. If the node is a member of the multicast group, it calculates the predicted LET of the last link of the path. The minimum between the last link expiration time and the MIN_LET value specified in the Join Query is the RET (Route Expiration Time).

   To select a route, a multicast receiver must wait for an appropriate amount of time after receiving the first Join Query so that all possible routes and their RET will be known. The receiver then chooses the most stable route (i.e., the route with the largest RET) and originates a Join Reply packet with the RET value enclosed (see Section 5.1.3.).

5. Increase the Hop Count field by 1 and decrease the TTL field by 1.

6. If the TTL field value is less than or equal to 0, then discard the packet. DONE.

7. If the TTL field value is greater than 0, then set the node's IP Address into Last Hop IP Address field and broadcast. DONE.

5.1.4. Originating a Join Reply

A multicast receiver transmits a "Join Reply" packet after selecting the multicast route. Each sender IP address and next hop IP address of a multicast group are contained in the Join Reply packet. The route expiration time is also included if the network hosts operate with GPS.

5.1.5. Processing a Join Reply

When a Join Reply is received:

1. The node looks up the Next Hop IP Address field of the received Join Reply entries. If no entries match the node's IP Address, do nothing. DONE.

2. If one or more entries coincide with the node's IP Address, set the FG_FLAG and build its own Join Reply. The next hop IP address can be obtained from the routing table.

3. Broadcast the Join Reply packet to the neighbor nodes. DONE.

5.1.6. Processing a Join Reply When GPS is Used

When a Join Reply is received:

1. The node looks up the Next Hop IP Address field of the received Join Reply entries. If no entries match the node's IP Address, do nothing. DONE.

2. If one or more entries coincide with the node's IP Address, set the FG_FLAG and build its own Join Reply. If multiple Join Replies with different RET values are received (i.e., the node lies in paths from the same source to multiple receivers), it selects the minimum RET among them and attaches the chosen RET value. Next hop IP address can be obtained from the routing table.

3. Broadcast the Join Reply packet to the neighbor nodes.

4. If the node is a source, it selects the minimum RET among all the Join Replies received. Then the source can build new routes by flooding a Join Query before the minimum RET approaches (i.e., route breaks of ongoing data sessions are imminent).

5.2. Handling a Multicast Data Packet

Multicast sources send the data whenever they have packets to send. Nodes relay data packets only if the packet is not a duplicate and the setting of FG_FLAG for the multicast group has not expired.

6. Simulation and Implementation Status

ODMRP has been implemented in both simulation and real ad hoc network testbed. For simulation, the protocols is implemented within the GloMoSim library [19], which is written in PARSEC[1]. As for the real system implementation, ODMRP is developed on Linux kernel version 2.0.36, the kernel version provided by the Red Hat Linux version 5.2. All tools and software packages that are used in our development originate from software bundle incorporated within the Red Hat Linux version 5.2 operating system package with the singular exception of Lucent WaveLan IEEE 802.11 device driver.

The papers presenting our simulation results and working wireless testbed implementation can be downloaded from the following website:

http://www.cs.ucla.edu/NRL/wireless

7. Protocol Applicability

7.1. Networking Context

ODMRP is best suited for mobile ad hoc wireless networks.

7.2. Protocol Characteristics and Mechanisms

* Does the protocol provide support for unidirectional links? (if so, how?)

  - No. We assume bidirectional links.

* Does the protocol require the use of tunneling? (if so, how?)

  - No.

* Does the protocol require using some form of source routing? (if so, how?)

  - No.

* Does the protocol require the use of periodic messaging? (if so, how?)

  - Yes, but only when multicast sources have data packets to send.

* Does the protocol require the use of reliable or sequenced packet delivery? (if so, how?)

  - No.

* Does the protocol provide support for routing through a multi-technology routing fabric? (if so, how?)

  - No.

* Does the protocol provide support for multiple hosts per router? (if so, how?)

  - No. In this document, we assume each mobile host is combined with a router, sharing the same IP address. It is possible, however, to extend the protocol to handle multiple hosts per router.

* Does the protocol support the IP addressing architecture? (if so, how?)

  - Yes. The message contains host IP address as its identification.

* Does the protocol require link or neighbor status sensing (if so, how?)

  - No.

* Does the protocol have dependence on a central entity? (if so, how?)

  - No.

* Does the protocol function reactively? (if so, how?)

  - Yes. For example, the source creates and maintains routes and multicast group membership only when it has data packets to send.

* Does the protocol function proactively? (if so, how?)

  - No.

* Does the protocol provide loop-free routing? (if so, how?)

  - Yes. By using the Message Cache, duplicate packets are detected and packets can only go through the loop-free route.

* Does the protocol provide for sleep period operation? (if so, how?)

  - TBD. The work is in progress.

* Does the protocol provide some form of security? (if so, how?)

  - TBD. The work is in progress.

* Does the protocol provide support for utilizing multi-channel, link-layer technologies? (if so, how?)

  - This document assumed an arbitrary single channel link-layer protocol. The protocol can work with any MAC and link-layer technology. It can also support multi-channel link-layer technology (e.g., separate channels for data, control packets, etc.).

Acknowledgments

Authors thank Ching-Chuan Chiang and Guangyu Pei for their initial contributions. We also send our gratitude to Sang Ho Bae who implemented ODMRP in a real ad hoc network testbed.

References

[1] P. Agrawal, D.K. Anvekar, and B. Narendran. Optimal Prioritization of Handovers in Mobile Cellular Networks. In Proceedings of IEEE PIMRC'94, The Hague, Netherlands, Sep. 1994, pp. 1393-1398.

[2] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song. PARSEC: A Parallel Simulation Environment for Complex Systems. IEEE Computer, vol. 31, no. 10, Oct. 1998, pp.77-85.

[3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A Media Access Protocol for Wireless LANs. In Proceedings of ACM SIGCOMM'94, London, UK, Sep. 1994, pp. 212-225.

[4] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119, March 1997.

[5] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade. AMRoute: Adhoc Multicast Routing Protocol. Internet Draft, draft-talpade-manet-amroute-00.txt, Aug. 1998. Work in progress.

[5] C.-C. Chiang, M. Gerla, and L. Zhang. Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks. ACM/Baltzer Cluster Computing, vol. 1, no. 2, 1998.

[6] M.S. Corson and S.G. Batsell. A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Initial Route Construction Phase. ACM/Baltzer Wireless Networks, vol. 1, no. 4, Dec. 1999, pp. 427-450.

[7] J.J. Garcia-Luna-Aceves and E.L. Madruga. A Multicast Routing Protocol for Ad-Hoc Networks. In Proceedings of IEEE INFOCOM'99, New York, NY, Mar. 1999, pp. 784-792.

[8] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification. IEEE std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[9] L. Ji and M.S. Corson.   A Lightweight Adaptive Multicast Algorithm. In Proceedings of IEEE GLOBECOM'98, Sydney, Australia, Nov. 1998, pp. 1036-1042.

[10] J. Jubin and J.D. Tornow.   The DARPA Packet Radio Network Protocols.   Proceedings of the IEEE, vol. 75, no. 1, Jan. 1987, pp. 21-32.

[11] E.D. Kaplan (Editor).   Understanding the GPS: Principles and Applications, Artech House, Boston, MA, Feb. 1996.

[12] L. Kleinrock and J. Silvester.   Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number.   In Proceedings of National Telecommunications Conference, Birmingham, AL, Dec. 1978, pp. 4.3.2-4.3.5.

[13] L. Kleinrock and F.A. Tobagi.   Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics.   IEEE Transactions on Communications, vol. COM-23, no. 12, Dec. 1975, pp. 1400-1416.

[14] S.-J. Lee, M. Gerla, and C.-C. Chiang.   On-Demand Multicast Routing Protocol.   In Proceedings of IEEE WCNC'99, New Orleans, LA, Sep. 1999, pp. 1298-1302.

[15] S.-J. Lee, W. Su, and M. Gerla.   Ad hoc Wireless Multicast with Mobility Prediction.   In Proceedings of IEEE ICCCN'99, Boston, MA, Oct. 1999, pp. 4-9.

[16] D.L. Mills.   Internet Time Synchronization: the Network Time Protocol.   IEEE Transactions on Communications, vol. 39, no. 10, Oct. 1991, pp. 1482-1493.

[17] B. Narendran, P. Agrawal, and D.K. Anvekar.   Minimizing Cellular Handover Failures Without Channel Utilization Loss. In Proceedings of IEEE GLOBECOM'94, San Francisco, CA, Dec. 1994, pp. 1679-1685.

[18] C.-K. Toh. Associativity-Based Routing for Ad-Hoc Mobile Networks.   Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems, Kluwer Academic Publishers, vol. 4, no. 2, Mar. 1997, pp. 103-139.

[19] UCLA 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/domains/glomosim.html

[20] UCLA Wireless Adaptive Mobility (WAM) Laboratory.
http://www.cs.ucla.edu/NRL/wireless

Chair's Address

The Working Group can be contacted via its current chairs:

M. Scott Corson
Institute for Systems Research
University of Maryland
College Park, MD 20742
USA

Phone: +1 301 405-6630
Email: corson@isr.umd.edu

Joseph Macker
Information Technology Division
Naval Research Laboratory
Washington, DC 20375
USA

Phone: +1 202 767-2001
Email: macker@itd.nrl.navy.mil

Authors' Addresses

Questions about this document can also be directed to the authors:

Sung-Ju Lee
3771 Boelter Hall
Computer Science Department
University of California
Los Angeles, CA 90095-1596
USA

Phone: +1 310 206-8589
Fax: +1 310 825-7578
Email: sjlee@cs.ucla.edu

William Su
3771 Boelter Hall
Computer Science Department
University of California
Los Angeles, CA 90095-1596
USA

Phone: +1 310 206-8589
Fax: +1 310 825-7578
Email: wsu@cs.ucla.edu

Mario Gerla
3732F Boelter Hall
Computer Science Department
University of California
Los Angeles, CA 90095-1596
USA

Phone: +1 310 825-4367
Fax: +1 310 825-7578
Email: gerla@cs.ucla.edu