Mobile Ad hoc Networking (MANET)
Internet-Draft
Intended status: Experimental
Expires: August 18, 2014

Y. Yi
S. Lee
W. Su
M. Gerla
A. Colin de Verdiere
University of California, Los Angeles

February 14, 2014

On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks
draft-gerla-manet-odmrp-00

Abstract

The 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.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Table of Contents

1. Introduction

This document describes the On-Demand Multicast Routing Protocol (ODMRP) [ODMRP-Journal]. ODMRP applies "on-demand" routing techniques to avoid channel overhead and improve scalability. It uses the concept of "forwarding group" [FGMP], 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.

The following properties of ODMRP highlight its advantages.

- o Simplicity
- o Low channel and storage overhead
- o Usage of up-to-date shortest routes
- o Reliable construction of routes and forwarding group
- o Robustness to host mobility
- o Maintenance and exploitation of multiple redundant paths
- o Exploitation of the broadcast nature of wireless environments
- o Unicast routing capability
- o Scalability using efficient flooding

2. Terminology and Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Additionally, it uses the notations defined in Section 2.1 and the terminology defined in Section 2.2 are used throughout this document.

2.1. Notation

ODMRP Routers generate and process messages, each of which has a number of distinct fields. For describing the protocol operation, specifically the generation and processing of such messages, the following notation is employed:

MsgType.field

where:

MsgType - is the type of message (e.g., JQ or JR);

field - is the field in the message (e.g., SourceAddress).

Furthermore, the following notational conventions are used:

a := b an assignment operator, whereby the left side (a) is assigned the value of the right side (b)

c = d a comparison operator, returning TRUE if and only if the value of the left side (c) is equal to the value of the right side (d)

The different messages, their fields and their meaning are described in Section 7.

2.2. Terminology

ODMRP Router - A router that implements this protocol. An ODMRP Router is equipped with at least one, and possibly more, ODMRP Interfaces

ODMRP Interface - An ODMRP Router's attachment to a communication medium, over which it receives and generates control messages, according to this specification. An ODMRP Router is assigned one or more addresses

Forwarding group - A group of ODMRP Routers participating in multicast packet forwarding

Multicast mesh - The topology defined by the link connection between forwarding group members

Multicast session - The entity defined by a (multicast group, source) pair, representing the group to which a source sends multicast packets

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

upstream - An ODMRP Router (A) is said to be "upstream" compared to another ODMRP Router (B), relative to a given multicast session, if A is on the path, which is discovered by a Join Query-Join Reply exchange and used by data packets, between B and the multicast source

downstream - Conversely, B is said to be downstream

3. Applicability Statement

This protocol is a reactive multicast routing protocol, intended for use in Mobile Ad Hoc Networks. MANETs generally have constrained resources (processing power, battery, etc.) and very dynamic topologies. With ODMRP, multicast meshes are created and maintained in and on-demand fashion, which avoids the issue of frequent tree reconfiguration seen with more classic multicast routing protocols.

4. Protocol Overview and Functioning

The objective of this protocol is to allow each ODMRP Router to:

- Build a multicast overlay only when there is data traffic to be sent to a multicast group
- Maintain a multicast overlay for as long as necessary, until there is no more data to be sent to the multicast group
- Join any multicast session in order to receive multicast packets from the corresponding source

4.1. Overview

These objectives are achieved, for each ODMRP Router, by the following operations:

- When having data to send to a multicast group, for which no overlay is already established, an ODMRP Router generates a Join Query, piggybacked on a data packet, and transmit it to all of its neighbors

- Upon receiving a Join Query, an ODMRP Router install or refresh a tuple in the Multicast Routing Set indicating the reverse path towards the source of the Join Query, then consider it for forwarding, according to the forwarding mechanism specified for the network

- If this Router belongs to the multicast group, to which the Join Query is addressed, it originates a Join Reply and transmits it to all of its neighbors

- Upon receiving a Join Reply, an ODMRP Router inspects the next hop address carried by this packet:

  * If it corresponds to an address of this Router, and if the ODMRP Router has a tuple in its Multicast Routing Set, corresponding to the advertised source, it installs or refreshes a tuple in its Forwarding Table, indicating that this ODMRP Router belongs to the Forwarding Group, which correspond to this multicast group and source. It then considers the Join Reply for forwarding, according to the forwarding mechanism specified for the network

  * Otherwise, it silently discards the Join Reply

- After sending a Join Reply, an ODMRP Router look in its Pre-acknowledgement Set for a corresponding Overheard tuple.

  * If such a tuple exists, it is discarded and no further action is taken

  * Otherwise, i.e. if the Pre-acknowledgement Set does not contain any corresponding Overheard tuple, it creates a Pending Acknowledgement tuple in the Pending Acknowledgement Set.

- While it has data to send to the multicast group, an ODMRP Router periodically originates a Join Query and transmit it to all of its neighbors

4.2. Information Base Overview

The protocol state is recorded in four disctinct information sets: the Multicast Routing Set, the Forwarding Table, the Pending Acknowledgement Set, the Pre-acknowledgement Set and the Blacklist.

The Multicast Routing Set contains tuples, each representing the address of a multicast group, the address of a source sending data to this multicast group, and the next hop towards the multicast source

The Forwarding Table contains tuples, each representing a (multicast group, multicast source) pair for which the ODMRP Router forwards packets

The Pending Acknowlegement Set contains tuples, each corresponding to a Join Reply which has been sent and is waiting for an acknowledgement from the designated upstream Forwarding Group Router

The Pre-acknowledgement Set contains tuples, representing overheard Join Reply messages, that are not destined to this Router but may pre-acknowledge a future Join Reply from this Router

The Blacklist contains tuples, corresponding to neighbour ODMRP Routers, with which connectivity has been detected to be unidirectional.

4.3. Signaling Overview

This protocol generates and processes the following routing messages:

Join Query - Generated by an ODMRP Router when it first has data packets to send to a multicast group, and flooded periodically to maintain the multicast overlay necessary to deliver these data packets. A Join Query message contains:

* The multicast group address
* The source address
* A sequence number

Join Reply - Generated by an ODMRP Router belonging to a multicast session in reply to a Join Query message advertising this multicast session, then forwarded by ODMRP Routers belonging to the corresponding forwarding group along the reverse path to the multicast source. A Join Reply message contains:

* The multicast group address
* The source address of the corresponding Join Query
* The sequence number carried by the corresponding Join Query
* The address of the next hop on the path towards the multicast session source

5. Parameters and Constants

This specification uses the following parameters and constants:

ROUTE_REFRESH_INTERVAL - is the interval between two perodic Join Queries sent by a Multicast Source

ROUTE_TIMEOUT - is the minimum time a Routing Tuple SHOULD be kept in the Routing Set after it was last refreshed

FG_TIMEOUT - is the minimum time a Forwarding Tuple SHOULD be kept in the Forwarding Table after it was last refreshed

ACK_TIMEOUT - is the time after which a Pending Tuple expires and MUST be considered invalid, as well as trigger the appropriate action according to Section 10

PRE_ACK_TIMEOUT - is the time after which an Overheard Tuple expires and MUST be considered invalid

6. Sequence Numbers

Each ODMRP Router maintains a single sequence number, which must be included in each Join Query or Join Reply message it generates. Each ODMRP Router MUST make sure that no two messages (both Join Query and Join Reply) are generated with the same sequence number, and MUST generate sequence numbers such that these are monotonically increasing. This sequence number is used as freshness information for when comparing routes to the ODMRP Router having generated the message.

However, with a limited number of bits for representing sequence numbers, wrap-around (that the sequence number is incremented from the maximum possible value to zero) will occur. To prevent this from interfering with the operation of the protocol, the following MUST be observed. The term MAX_SEQ_NUM designates in the following the largest possible value for a sequence number. The sequence number S1 is said to be "greater than" (denoted '>') the sequence number S2 if:

S2 < S1 AND S1 - S2 <= MAX_SEQ_NUM/2 OR

S1 < S2 AND S2 - S1 > MAX_SEQ_NUM/2

7. Packets and Messages

This section describes the protocol messages generated and processed by ODMRP, according to the notations defined in Section 2.

7.1. Join Query Format

A Join Query (JQ) message has the following fields:

JQ.AddressLength is a 4 bit unsigned integer field, encoding the length of the addresses carried by this message as follows:

JQ.AddressLength := the length of an address in octets - 1

JQ.MulticastGroupAddress is an unsigned integer field, of length JQ.AddressLength + 1 octets, encoding the address of the Multicast Group, to which this Join Query is addressed

JQ.SourceAddress is an unsigned integer field, of length JQ.AddressLength + 1 octets, encoding the address of the source of this Join Query

JQ.SequenceNumber is a 16 bit unsigned integer field, containing the sequence number (see Section 6) of the ODMRP Router, generating the Join Query message

7.2. Join Reply Format

A Join Reply (JR) message has the following fields:

JR.AddressLength is a 4 bit unsigned integer field, encoding the length of the addresses carried by this message as follows:

JR.AddressLength := the length of an address in octets - 1

JR.MulticastGroupAddress is an unsigned integer field, of length JR.AddressLength + 1 octets, encoding the address of the Multicast Group, to which this Join Reply is addressed

JR.AckRequired is a boolean flag. When set ('1'), it specify that the recipient of the Join Reply MUST acknowledge its reception by a Join Reply. If cleared ('0'), the recipient of this message MAY suppress it's Join Reply transmission, according to Section 9

JR.SourceAddress is an unsigned integer field, of length JQ.AddressLength + 1 octets, encoding the address of the Source of the Multicast Session

JR.SequenceNumber is a 16 bit unsigned integer field, containing the sequence number (see Section 6) of the corresponding Join Query message

JR.NextHopAddress is an unsigned integer field, of length JQ.AddressLength + 1 octets, encoding the the address of the next hop on the path towards the source of the multicast session

8. Information Bases

Each router maintains an Information Base, containing a Multicast Routing Set and a Forwarding Table, as described in the following sections. These information sets are given so as to facilitate description of message generation, forwarding and processing rules. In particular, an implementation may chose any representation or structure for when maintaining this information.

8.1. Multicast Routing Set

The Multicast Routing Set contains Routing Tuples, indicating the path towards Multicast Sources, and containing the following fields:

(R_source, R_next_hop, R_seq_num, R_exp_time)

Where:

R_source - is the address of the Multicast Source

R_next_hop - is the address of the next hop along the path to the Multicast Source,i.e., the address of the neighbor ODMRP Router, from which the last valid Join Query message from this source was received)

R_seq_num - corresponds to the JQ.SequenceNumber of the last valid Join Query originated by the Multicast Source and received by the ODMRP Router

R_exp_time - is the time at which the tuple MUST be considered expired and thus MUST NOT be taken into consideration by the operations of this protocol

8.2. Forwarding Table

The Forwarding Table contains Forwarding Tuples, representing Multicast Sessions for which the ODMRP Router forwards messages, i.e., the ODMRP Router is part of these Multicast Sessions' Forwarding Groups. These tuples are as follows:

(F_multicast_group, F_multicast_source, F_seq_num, F_exp_time)

Where:

F_multicast_group - is the address of the Multicast Group of the Multicast Session, for which the ODMRP Router forwards messages

F_source - is the address of the Multicast Source of the Multicast Session, for which the ODMRP Router forwards messages

F_seq_num - is the sequence number, corresponding to the last Join Query sent by the multicast source for the multicast session

F_exp_time - is the time at which the tuple MUST be considered expired and thus MUST NOT be taken into consideration by the operations of this protocol

8.3. Pending Acknowledgements

The Pending Acknowledgements Set contains Pending Acknowledgement tuples, representing Join Reply messages that are waiting to be acknowledged by the selected upstream Forwarding Group member. These tuples are as follows:

(P_multicast_group, P_multicast_source, P_seq_num,
P_next_hop, P_nth_time, P_exp_time)

Where:

P_multicast_group - is the JR.MulticastGroupAddress carried in the Join Reply awaiting acknowledgement (hence corresponding Join Reply)

P_multicast_source - is the JR.SourceAddress field carried in the corresponding Join Reply

P_seq_num - is the JR.SequenceNumber field of the corresponding Join Reply

P_next_hop - is the JR.NextHopAddress field of the corresponding Join Reply

P_nth_time - corresponds to the number of times this Join Reply has been previously sent without being acknowledged

P_exp_time - is the time at which this tuple MUST be considered expired

P_acknowledged - is a boolean indicating whether the corresponding Join Reply has been acknowledged

8.4. Pre-acknowledgements

The Pre-acknowledgements Set contains Overheard Tuples, corresponding to Join Reply messages, which have been sent by neighbors of this ODMRP Router but do not contain an address of this Router and do not acknowledge any tuple in the Pending Acknowledgement Set. The Overheared Tuples are as follows:

(O_multicast_group, O_multicast_source, O_seq_num,
O_originator, O_exp_time)

Where:

O_multicast_group - is the JR.MulticastGroupAddress carried in the overhread Join Reply

O_multicast_source - is the JR.SourceAddress field carried in the corresponding Join Reply

O_seq_num - is the JR.SequenceNumber field of the corresponding Join Reply

O_originator - is the address of the ODMRP Router's interface which has sent the Join Reply

O_exp_time - is the time at which this tuple expires MUST be considered invalid

8.5. Blacklist

The Blacklist contains Blacklisted Tuples, corresponding to neighbour ODMRP Routers, with which connectivity has been detected to be unidirectional, i.e., which have not acknowledged Join Replies from this Router, as specified in Section 9. The Blacklist Tuples are as follows:

(B_neighbour, B_exp_time)

Where:

B_neighbour - is the address of an interface of the blacklisted ODMRP neighbour

B_exp_time - is the time at which this tuple expires and MUST be considered invalid

9. Protocol Details

This protocol generates and processes Join Query and Join Reply messages, according to the operations described in the following sections. This section uses the additional notation and variables:

previous-hop-address - refers to the address of the interface of the ODMRP Router, from which the message currently processed (Join Query or Join Reply) has been received

this Router - refers to the ODMRP Router generating, processing or forwarding the message (Join Query or Join Reply)

9.1. Join Query

A Join Query is generated by an ODMRP Router, which has data to send to a multicast group, but no multicast session has been initialized. Join Queries are then periodically originated by the ODMRP Router while it has data to send to the multicast group.

9.1.1. Invalid Join Queries

A Join Query, received by an ODMRP Router, is invalid and MUST be discarded without processing (and in particular, MUST NOT be considered for forwarding) if:

- The address length carried by the Join Query (see Section 7) differs from the length of the addresses of this Router
- The Routing Set of this Router contains a tuple for which:
  - R_multicast_source = JQ.SourceAddress, and
  - R_seqnum > JQ.SequenceNumber or R_seqnum = JQ.SequenceNumber

- JQ.SourceAddress is an address of this Router
- The Blacklist contains a Blacklisted Tuple, for which B_neighbour = previous-hop

9.1.2. Join Query Generation

A Join Query is generated according to Section 7 with the following content:

- JQ.AddressLength set to the length of the address of this Router minus 1, as specified in Section 7
- JQ.MulticastGroupAddress set to the address of the multicast group, to which this Router is sending data
- JQ.SourceAddress set to an address of this ODMRP Router
- JQ.SequenceNumber set to the current sequence number of this Router, as specified in Section 6

9.1.3. Join Query Processing

Upon receiving a valid Join Query message, an ODMRP Router proceeds as follows:

1. Find the Routing Tuple which satisfies: R_source = JQ.SourceAddress
2. If no such tuple exists, create a Routing Tuple with the following fields:
   - R_source := JQ.SourceAddress
   - R_next_hop := previous-hop
   - R_seq_num := JQ.SequenceNumber
   - R_exp_time := current_time + ROUTE_TIMEOUT

   and insert this tuple in the Routing Set
3. Else, i.e., if such a tuple exist, update it as follows:
   - R_next_hop := previous-hop
   - R_seq_num := JQ.SequenceNumber

* R_exp_time := current_time + ROUTE_TIMEOUT

4. Consider the Join Query for forwarding, according to Section 9.1.4

5. If this Router is a member of the Multicast Group, addressed by JQ.MulticastGroupAddress, create a new Join Reply according to Section 9.2 and transmit it to all of this Router's neighbors

9.1.4. Join Query Forwarding

A Join Query considered for forwarding MUST be updated as follows:

- JQ.PreviousHop is set to an address of this Router

The Join Query is then forwarded to all of this Router's neighbors

9.2. Join Reply

A Join Reply is generated by an ODMRP Router in response to a Join Query, for which JQ.MulticastGroupAddress is the address of a Multicast Group, which this Router is part of.

9.2.1. Invalid Join Replies

A Join Reply, received by an ODMRP Router, is invalid and MUST be discarded without processing (and in particular, MUST NOT be considered for forwarding) if:

- The address length carried by the Join Reply (see Section 7) differs from the length of the address of the ODMRP Router

- There exists a Forwarding Tuple in this Router's Forwarding Group table, such as:

  * F_source = JR.MulticastSourceAddress

  * F_seq_num > JR.SequenceNumber

9.2.2. Join Reply Generation

A Join Reply MUST be generated in response to a received Join Query advertising a Multicast Group, which this Router belongs to. A Join Reply is generated according to Section 7 with the following content:

- JR.AddressLength is set to the length of the address of this router minus 1, as specified in Section 7

- JR.MulticastGroupAddress is set to JQ.MulticastGroupAddress for the Join Query corresponding to this Join Reply
- JR.SourceAddress is set to JQ.SourceAddress for the Join Query corresponding to this Join Reply
- JR.SequenceNumber is set to JQ.SequenceNumber for the Join Query corresponding to this Join Reply
- JR.NextHopAddress is set to the address of the interface of the ODMRP Router, which transmitted the Join Query message

9.2.3. Join Reply Processing

Upon receiving a valid Join Reply, an ODMRP Router proceeds as follows:

1. If JR.NextHopAddress is an address of this ODMRP Router:
    1. Find the Forwarding Tuple (henceforth Matching Forwarding Tuple) such that:
        + F_multicast_group = JR.MulticastGroupAddress
        + F_multicast_source = JR.MulticastSourceAddress
    2. If no such tuple exists, insert in the Forwarding Table a new Forwarding Tuple such that:
        + F_multicast_group = JR.MulticastGroupAddress
        + F_multicast_source = JR.MulticastSourceAddress
        + F_seq_num = JR.SequenceNumber
        + F_exp_time = current_time + FG_TIMEOUT

        And set new-jr to TRUE
    3. Else, the variable "new-jr" is set to TRUE if JR.SequenceNumber > F_seq_num, and to FALSE otherwise. Then, the pre-existing Matching Forwarding Tuple is updated as follows:
        + F_seq_num := JR.SequenceNumber
        + F_exp_time := current_time + FG_TIMEOUT

4. If new-jr = TRUE or if JR.AckRequired is set the Join Reply is considered for forwarding. Otherwise, it is not processed further.

2. Else, find the Routing Tuple in the Routing Set (henceforth "Matching Routing Tuple"), such as:

* R_source = JR.SourceAddress

* R_seq_num <= JR.SequenceNumber

If previous-hop-address = R_next_hop, then:

3. If the Pending Acknowledgement Set contains a Pending Tuple (henceforth "Matching Pending Tuple") such as:

+ P_multicast_group = JR.MulticastAddress

+ P_multicast_source = JR.SourceAddress

+ P_seq_num = JR.SequenceNumber

+ P_next_hop = previous-hop-address

The Matching Pending Tuple MUST be updated as follows:

+ P_acknowledged = TRUE

+ P_exp_time = EXPIRED

The Join Reply is not processed further, and in particular MUST NOT be considered for forwarding

4. Else, if the Pre-Acknowledgement Set does not contain any Overheard Tuple such as:

+ O_multicast_group = JR.MulticastGroupAddress

+ O_multicast_source = JR.SourceAddress

+ O_seq_num = JR.SequenceNumber

+ O_originator = previous-hop-address

Insert a tuple with these fields, and O_exp_time = current_time + PRE_ACK_TIMEOUT in the Pre-Acknowledgement Set. The Join Reply is not processed further, and in particular MUST NOT be considered for forwarding

3. Else, the Join Reply is silently discarded without further processing

9.2.4. Join Reply Forwarding

A Join Reply, considered for forwarding, MUST be updated as follows:

- Find the Matching Routing Tuple, such that:
  - R_source = JR.MulticastSourceAddress
  - R_seq_num = JR.SequenceNumber
- Set JR.NextHop to R_next_hop

The Join Reply is then transmitted according to Section 9.2.5

9.2.5. Join Reply Transmission

A Join Reply is transmitted to all of an ODMRP Router's neighbors, in order to achieve two objectives:

- Set up or refresh the corresponding Forwarding Tuple for the upward ODMRP neighbor
- If the Join Reply was not originated by this router, acknowledge its reception to the previous hop.

To this end, a Join Reply is updated in the following way prior to being transmitted:

1. Find the Routing Tuple in the Routing Set (henceforth "Matching Routing Tuple"), such as:
   - R_source = JR.SourceAddress
   - R_seq_num <= JR.SequenceNumber
2. If no such tuple exists, then the Join Reply is not processed further and is silently discarded
3. Else, the Join Reply is updated as follows:
   - JR.NextHopAddress := R_next_hop from the Matching Routing Tuple
   - If the Pre-acknowledgement Set contains a tuple, such that:

+ O_multicast_group = JR.MulticastGroupAddress

+ O_multicast_source = JR.SourceAddress

+ O_seq_num = JR.SequenceNumber

+ O_originator = JR.NextHopAddress

Then clear the JR.AckRequired flag, and set O_exp_time to EXPIRED

* Else, if the Pending Acknowledgement Set contains a Pending Tuple such as:

+ P_multicast_group = JR.MulticastGroupAddress

+ P_multicast_source = JR.SourceAddress

+ P_seq_num = JR.SequenceNumber

+ P_next_hop = JR.NextHopAddress

Then set JR.AckRequired, and increase P_nth_time by 1

* Finally, if neither the Pre-acknowledgement Set nor the Pending Acknowledgement Set contain a corresponding tuple:

1. Insert a Pending Tuple in the Pending Acknowledgement Set, such as:

   - P_multicast_group = JR.MulticastGroupAddress

   - P_multicast_source = JR.SourceAddress

   - P_seq_num = JR.SequenceNumber

   - P_next_hop = JR.NextHopAddress

   - P_nth_time = 1

   - P_acknowledged = FALSE

   - P_expiration_time = current_time + ACK_TIMEOUT

2. Clear the JR.AckRequired flag

9.3. Multicast Mesh Maintenance

While an ODMRP Router has data to send to a Multicast Group, it MUST maintain the multicast mesh generated by the initial Join Query. To this end, Join Queries are periodically generated by this Router according to Section 9.1.2. The interval between two Join Queries SHOULD be no less than ROUTE_REFRESH_INTERVAL.

9.4. Message Transmission

When using physical media subject to collisions and packet loss, both Join Query and Join Reply messages SHOULD be jittered to minimize the effect of collisions, as described in [RFC5148]

10. Unidirectional links handling

Upon sending a Join Reply, an ODMRP Router MUST verify that the upstream neighbor has joined the forwarding group. To this end, the following three mechanisms are used after transmitting a given Join Reply:

- If the ODMRP Router overhears a corresponding Join Reply from the upstream neighbor (see Section 9.2.3), this verifies that the link is bidirectional and that the upstream neighbor has joined the forwarding group (passive acknowledgement)
- If the ODMRP Router has already overheard a corresponding Join Reply from the upstream neighbor prior to transmitting its own Join Reply, this means that the upstream neighbor has already joined the forwarding group (see Section 9.2.3) (pre-acknowledgement)
- Else, i.e. if neither the pre-acknowledgement nor the passive acknowledgement have verified that the upstream neighbor joined the forwarding group (i.e. if the corresponding Pending Tuple expires with P_acknowledged set to False), the ODMRP Router SHOULD retransmit the Join Reply with the JR.AckRequired flag set.

11. Multicast Packets Forwarding

ODMRP Routers originating and forwarding multicast packets MUST implement a duplicate packet detection (DPD) mechanism. If using IPv4 or IPV6 addresses, the use of SMF [RFC6621] is RECOMMENDED.

An ODMRP Router, receiving a non-duplicate multicast data packet, transmits it to all of its neighbors if it is a member of the

forwarding group for this data packet, i.e. there exists a tuple in the Forwarding Group Table such as:

F_multicast_group correspond to the multicast address of this packet

F_multicast_source corresponds to the source of this packet

12. Security Considerations

This document does currently not have any security considerations.

13. IANA considerations

This specification defines two new Message Types, which must be allocated from the "Message Type" repository of [RFC5444].

13.1. Join Query Registries

Type	Description	Allocation Policy
128-223	Unassigned	Expert Review

Table 1: Join Query Message-Type-specific Message TLV Types

Type	Description	Allocation Policy
128	ADDR-TYPE	
129-223	Unassigned	Expert Review

Table 2: Join Query Message-Type-specific Addres Block TLV Types

Allocation of the ADDR-TYPE TLV from the Join Query specific Address Block TLV Types will create a new Type Extension Registry with assignments as specified in Table 3.

Name	Type	Type Extension	Description	Allocation Policy
ADDR-TYPE	128	0	MULTICAST-GROUP-ADDRESS	
ADDR-TYPE	128	1-255	Unassigned	Expert Review

Table 3: Address Block TLV Type assignment for ADDR-TYPE

13.2. Join Reply Registries

Type	Description	Allocation Policy
128	ACKREQUIRED	
129-223	Unassigned	Expert Review

Table 4: Join Reply Message-Type-specific Message TLV Types

Type	Description	Allocation Policy
128	ADDR-TYPE	
129-223	Unassigned	Expert Review

Table 5: Join Reply Message-Type-specific Addres Block TLV Types

Allocation of the ADDR-TYPE TLV from the Join Reply specific Address Block TLV Types will create a new Type Extension Registry with assignments as specified in Table 6.

Name	Type	Type Extension	Description	Allocation Policy
ADDR-TYPE	128	0	MULTICAST-GROUP-ADDRESS	
ADDR-TYPE	128	1	NEXT-HOP-ADDRESS	Expert Review
ADDR-TYPE	128	2-255	Unassigned	Expert Review

Table 6: Address Block TLV Type assignment for ADDR-TYPE

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", RFC 5148, February 2008.

[RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized MANET Packet/Message Format", RFC 5444, February 2009.

[RFC6621] Macker, J., "Simplified Multicast Forwarding", RFC 6621, May 2012.

14.2. Informative References

[FGMP] Chiang, C., Gerla, M., and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks", Avril 1998.

[ODMRP-Journal] Lee, S., Su, W., and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Networks".

Appendix A. RFC5444 Encoding

This section describes the encoding of ODMRP messages using [RFC5444].

A.1. Join Query Encoding

This protocol defines the Join Query message type. Hence, according to [RFC5444], all Join Query messages are generated, processed and transmitted following this specification. Table 7 shows the mapping between the Join Query elements described in Section 7.1 and their encoding. All elements described in Table 7 MUST be included in every Join Query message.

JQ Element	RFC5444 Element	Considerations
JQ.AddressLength	<msg-addr-length>	Encodes from 1 to 16 bytes addresses
JQ.SourceAddress	<msg-orig-addr>	
JQ.MulticastGroupAddress	Address in address block + TLV	Is encoded by way of an address block with associated message type-specific TLV of type ADDR-TYPE and value MULTICAST-GROUP-ADDRES S.
JQ.SequenceNumber	<msg-seq-num>	16 bits, hence MAX_SEQ_NUM is 65535

Table 7: Join Query Message Elements

A.2. Join Reply Encoding

This protocol defines the Join Reply message type. Hence, according to [RFC5444], all Join Reply messages are generated, processed and transmitted following this specification. Table 8 shows the mapping between the Join Reply elements described in Section 7.2 and their encoding. With the exception of the ACKREQUIRED TLV, all elements described in Table 8 MUST be included in every Join Reply message.

JR Element	RFC5444 Element	Considerations
JR.AddressLength	<msg-addr-length>	Encodes from 1 to 16 bytes addresses
JR.SourceAddress	<msg-orig-addr>	
JR.MulticastGroupAddress	Address in address block + TLV	Is encoded by way of an address block with associated message type-specific TLV of type ADDR-TYPE and value MULTICAST-GROUP-ADDRES S.
JR.SequenceNumber	<msg-seq-num	16 bits, hence MAX_SEQ_NUM is 65535

JR.NextHopAddress	Address in address block + TLV	Is encoded by way of an address block with associated message type-specific TLV of type ADDR-TYPE and value NEXT-HOP-ADDRESS.	
JR.AckRequired	ACKREQUIRED TLV	Encoded by way of a Message-type-specific TLV of type ACKREQUIRED that has no value. If JR.AckRequired is set, the ACKREQUIRED TLV MUST be included; otherwise, it MUST NOT be included	

Table 8: Join Reply Message Elements

Appendix B. Illustrations

This section shows examples of ODMRP control messages encoded using [RFC5444]. [RFC5444] specifies that a packet is formed by a packet header, an optional TLV block and zero or more messages. This specification does not use or require any packet TLV. Additionally, the minimal packet header required by ODMRP is shown in Figure 1.

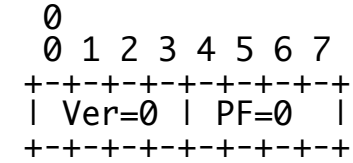


Figure 1: Packet Header

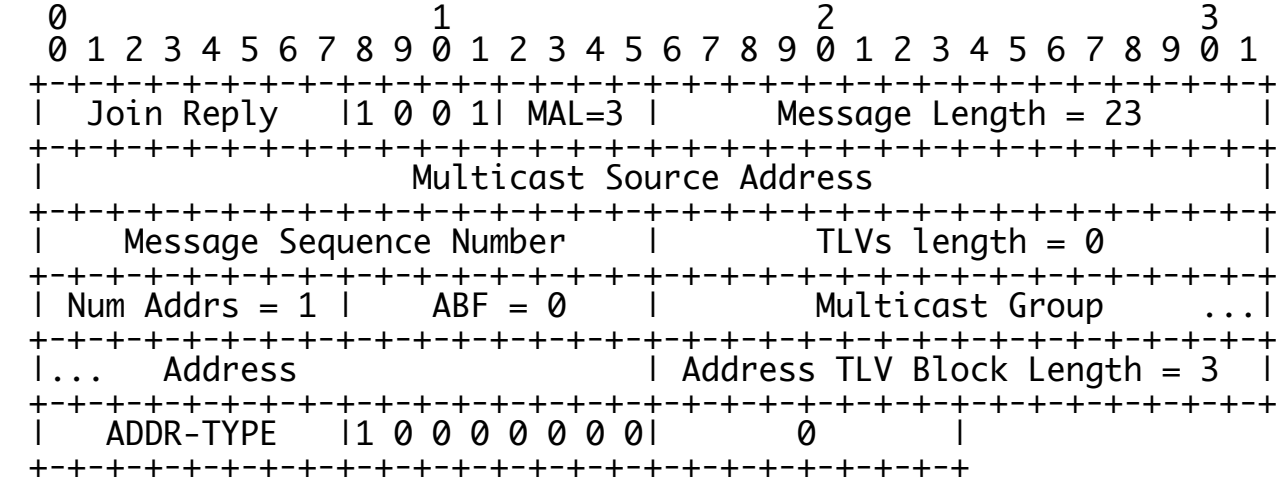
B.1. Join Query Message

JQ messages are instances of [RFC5444] messages. This section illustrates an example of one such message.

The JQ message's header's flag octet has a value of 9, meaning that the sequence number and source address fields are present, encoding respectively the sequence number and the address of the multicast source that originated the message. Additionally, the address length field (MAL) is set to 3, corresponding to an address length of 4

octets (i.e., the length of an IPv4 address). The overall message size is 23 octets.

An additional Message-Type specific address block is present, with one address and a flag octet (ABF) having value 0, meaning that the address block has no Head or Tail element. The Mid element encodes the Multicast group address. The associated TLV is of type ADDR-TYPE and value 0, i.e. MULTICAST-GROUP-ADDRESS.

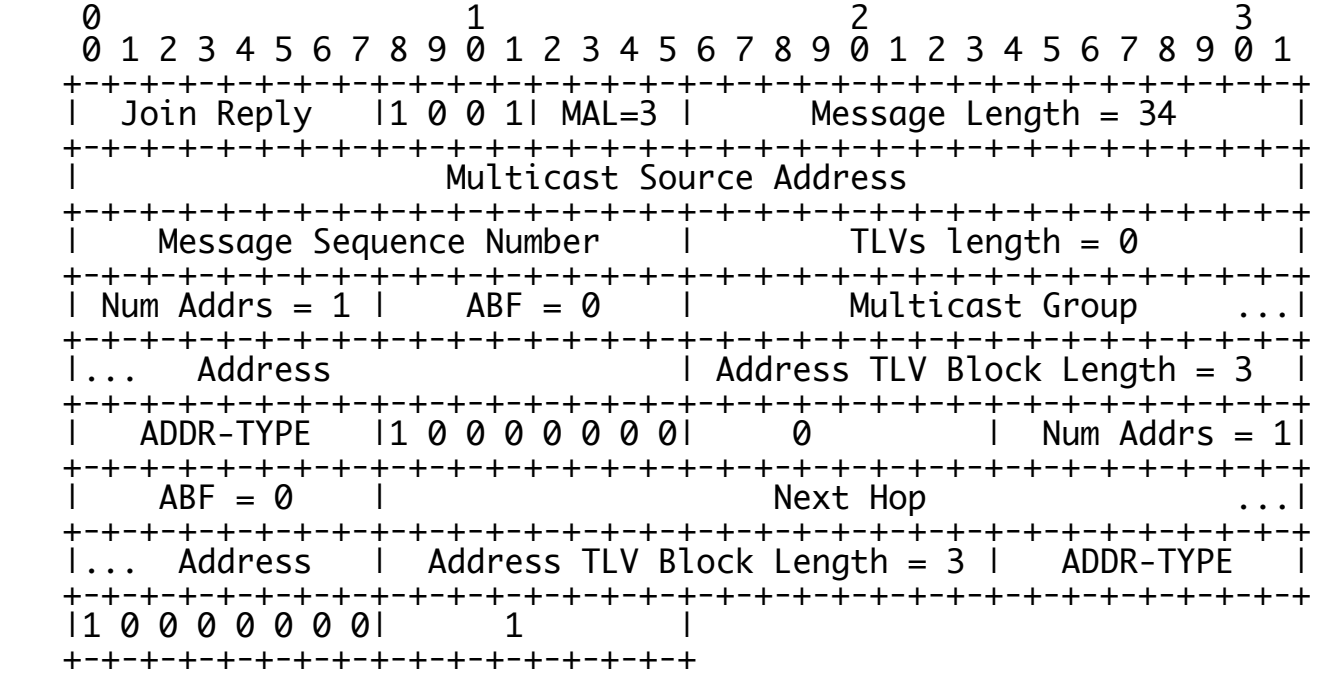


B.2. Join Reply Message

JR messages are instances of [RFC5444] messages. This section illustrates an example of one such message.

The JR message's header's flag octet has a value of 9, meaning that the sequence number and source address fields are present, encoding respectively the sequence number and the address of the multicast source that originated the message. Additionally, the address length field (MAL) is set to 3, corresponding to an address length of 4 octets (i.e., the length of an IPv4 address). The overall message size is 34 octets.

Two additional Message-Type specific address blocks are present, both with one address and a flag octet (ABF) having value 0, meaning that the address block has no Head or Tail element. For the first address block, the Mid element encodes the Multicast group address; the associated Message-Type-specific TLV is of type ADDR-TYPE and value 0, i.e. MULTICAST-GROUP-ADDRESS. The second address block's Mid element encodes the Next Hop address; its associated Message-Type-specific TLV is of type ADDR-TYPE and value 1, i.e., NEXT-HOP-ADDRESS.



Authors' Addresses

Yunjung Yi
University of California, Los Angeles

Sung-Ju Lee
University of California, Los Angeles

William Su
University of California, Los Angeles

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

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

Axel Colin de Verdiere
University of California, Los Angeles

Email: axel@axelcdv.com