ROLL
Internet-Draft
Intended status: Standards Track
Expires: July 28, 2013

J. Hui
Cisco
R. Kelsey
Silicon Labs

January 24, 2013

Multicast Protocol for Low power and Lossy Networks (MPL) draft-ietf-roll-trickle-mcast-03

Abstract

This document specifies the Multicast Protocol for Low power and Lossy Networks (MPL) that provides IPv6 multicast forwarding in constrained networks. MPL avoids the need to construct or maintain any multicast forwarding topology, disseminating messages to all MPL forwarders in an MPL domain. MPL uses the Trickle algorithm to manage message transmissions for both control and data-plane messages. Different Trickle parameter configurations allow MPL to trade between dissemination latency and transmission efficiency.

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 July 28, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Applicability Statement 5
4. Protocol Overview 6
4.1. Information Base Overview 6
4.2. Overview 6
4.3. Signaling Overview 7
5. MPL Constants 9
5.1. Multicast Addresses 9
5.2. Message Types 9
5.3. MPL Forwarder Parameters 9
5.4. Trickle Parameters 9
6. Protocol Message Formats 11
6.1. MPL Option 11
6.2. MPL Control Message 12
6.3. MPL Seed Info 13
7. Information Base 15
7.1. Local Interface Set 15
7.2. Domain Set 15
7.3. Seed Set 15
7.4. Buffered Message Set 15
8. MPL Domains 17
9. MPL Seed Sequence Numbers 18
10. MPL Data Messages 19
10.1. MPL Data Message Generation 19
10.2. MPL Data Message Transmission 19
10.3. MPL Data Message Processing 20
11. MPL Control Messages 22
11.1. MPL Control Message Generation 22
11.2. MPL Control Message Transmission 22
11.3. MPL Control Message Processing 23
12. Acknowledgements 25
13. IANA Considerations 26
13.1. MPL Option Type 26
13.2. MPL ICMPv6 Type 26
13.3. Well-known Multicast Addresses 26
14. Security Considerations 27
15. Normative References 28
Authors' Addresses 29

1. Introduction

Low power and Lossy Networks typically operate with strict resource constraints in communication, computation, memory, and energy. Such resource constraints may preclude the use of existing IPv6 multicast routing and forwarding mechanisms. Traditional IP multicast delivery typically relies on topology maintenance mechanisms to discover and maintain routes to all subscribers of a multicast group. However, maintaining such topologies in LLNs is costly and may not be feasible given the available resources.

Memory constraints may limit devices to maintaining links/routes to one or a few neighbors. For this reason, the Routing Protocol for LLNs (RPL) specifies both storing and non-storing modes [RFC6550]. The latter allows RPL routers to maintain only one or a few default routes towards a LLN Border Router (LBR) and use source routing to forward messages away from the LBR. For the same reasons, a LLN device may not be able to maintain a multicast routing topology when operating with limited memory.

Furthermore, the dynamic properties of wireless networks can make the cost of maintaining a multicast routing topology prohibitively expensive. In wireless environments, topology maintenance may involve selecting a connected dominating set used to forward multicast messages to all nodes in an administrative domain. However, existing mechanisms often require two-hop topology information and the cost of maintaining such information grows polynomially with network density.

This document specifies the Multicast Protocol for Low power and Lossy Networks (MPL), which provides IPv6 multicast forwarding in constrained networks. MPL avoids the need to construct or maintain any multicast routing topology, disseminating multicast messages to all MPL forwarders in an MPL domain. By using the Trickle algorithm [RFC6206], MPL requires only small, constant state for each MPL device that initiates disseminations. The Trickle algorithm also allows MPL to be density-aware, allowing the communication rate to scale logarithmically with density.

2. Terminology

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].

The following terms are used throughout this document:

MPL Forwarder - A router that implements this protocol. A MPL Forwarder is equipped with at least one MPL Interface.

MPL Interface - An MPL Forwarder's attachment to a communications medium, over which it transmits and receives MPL Data Messages and MPL Control Messages according to this specification. An MPL Interface is assigned one or more unicast addresses and is subscribed to one or more MPL Domain Addresses.

MPL Domain Address - A multicast address that identifies the set of MPL Interfaces within an MPL Domain. MPL Data Messages disseminated in an MPL Domain have the associated MPL Domain Address as their destination address.

MPL Domain - A scope zone, as defined in [RFC4007], in which MPL Interfaces subscribe to the same MPL Domain Address and participate in disseminating MPL Data Messages.

MPL Data Message - A multicast message that is used to communicate a multicast payload between MPL Forwarders and contains an MPL Option in the IPv6 header. A MPL Data Message has its destination address set to the MPL Domain Address.

MPL Control Message - A link-local multicast message that is used to communicate information about recently received MPL Data Messages to neighboring MPL Forwarders.

MPL Seed - An MPL Forwarder that generates MPL Data Messages and serves as an entry point into an MPL Domain.

3. Applicability Statement

This protocol is an IPv6 multicast forwarding protocol for Low-Power and Lossy Networks. By implementing a controlled dissemination using the Trickle algorithm, this protocol is designed for networks that communicate using low-power and lossy links with widely varying topologies in both the space and time dimensions.

4. Protocol Overview

The goal of MPL is to deliver multicast messages to all interfaces that subscribe to the multicast messages' destination address within an MPL Domain.

4.1. Information Base Overview

A node records necessary protocol state in the following information sets:

- The Local Interface Set records the set of local MPL Interfaces and the unicast addresses assigned to those MPL Interfaces.

- The Domain Set records the set of MPL Domain Addresses and the local MPL Interfaces that subscribe to those addresses.

- The Seed Set records information about received MPL Data Messages received from an MPL Seed. The Seed Set maintains the minimum sequence number that the MPL Forwarder is willing to receive or has buffered in its Buffered Message Set. MPL uses the Seed Set and Buffered Message Set to determine when to accept an MPL Data Message, process its payload, and retransmit it.

- The Buffered Message Set records recently received MPL Data Messages from an MPL Seed. MPL Data Messages resident in the Buffered Message Set have sequence numbers that are greater than or equal to the minimum threshold maintained in the Seed Set. MPL uses the Buffered Message Set to store MPL Data Messages that may be transmitted by the MPL Forwarder for forwarding.

4.2. Overview

MPL achieves its goal by implementing a controlled flood that attempts to disseminate the multicast data message to all interfaces within an MPL Domain. MPL performs the following tasks to disseminate a multicast message:

- When having a multicast message to forward into an MPL Domain, the MPL Seed generates an MPL Data Message that includes the MPL Seed Identifier, a newly generated sequence number, and the multicast message. If the multicast destination address is not the MPL Domain Address, IP-in-IP [RFC2473] is used to encapsulate the multicast message in the MPL Data Message.

- Upon receiving an MPL Data Message, the MPL Forwarder extracts the MPL Seed and sequence number and determines whether or not the MPL Data Message was previously received using the Seed Set and

Buffered Message Set.

* If the sequence number is less than the lower-bound sequence number maintained in the Seed Set or a message with the same sequence number exists within the Buffered Message Set, the MPL Forwarder marks the MPL Data Message as old.

* Otherwise, the MPL Forwarder marks the MPL Data Message as new.

o For each newly received MPL Data Message, an MPL Forwarder updates the Seed Set, adds the MPL Data Message into the Buffered Message Set, processes its payload, and multicasts the MPL Data Message a number of times on all MPL Interfaces participating in the same MPL Domain to forward the message.

o Each MPL Forwarder may periodically link-local multicast MPL Control Messages on MPL Interfaces to communicate information contained in the MPL Forwarder's Seed Set and Buffered Message Sets.

o Upon receiving an MPL Control Message, an MPL Forwarder determines whether there are any new MPL Data Messages that have yet to be received by the MPL Control Message's source and multicasts those MPL Data Messages.

MPL's configuration parameters allow two forwarding strategies for disseminating MPL Data Messages.

Proactive Forwarding - With proactive forwarding, an MPL Forwarder schedules transmissions of MPL Data Messages using the Trickle algorithm, without any prior indication that neighboring nodes have yet to receive the message. After transmitting the MPL Data Message a limited number of times, the MPL forwarder may terminate proactive forwarding for the MPL Data Message message.

Reactive Forwarding - With reactive forwarding, an MPL Forwarder link-local multicasts MPL Control Messages using the Trickle algorithm [RFC6206]. MPL Forwarders use MPL Control Messages to discover new MPL Data Messages that have not yet been received. When discovering that a neighboring MPL Forwarder has not yet received a new MPL Data Message, the MPL Forwarder schedules those MPL Data Messages for transmission using the Trickle algorithm.

4.3. Signaling Overview

This protocol generates and processes the following messages:

MPL Data Message - Generated by an MPL Seed to deliver a multicast message across an MPL Domain. The MPL Data Message's source is an address in the Local Interface Set of the MPL Seed that generated the message and is valid within the MPL Domain. The MPL Data Message's destination is the MPL Domain Address corresponding to the MPL Domain. An MPL Data Message contains:

* The Seed Identifier of the MPL Seed that generated the MPL Data Message.

* The sequence number of the MPL Seed that generated the MPL Data Message.

* The original multicast message.

MPL Control Message - Generated by an MPL Forwarder to communicate information contained in the Seed Set and Buffered Message Set to neighboring MPL Forwarders. An MPL Control Message contains a list of tuples for each entry in the Seed Set. Each tuple contains:

* The minimum sequence number maintained in the Seed Set for the MPL Seed.

* A bit-vector indicating the sequence numbers of MPL Data Messages resident in the Buffered Message Set for the MPL Seed, where the first bit represents a sequence number equal to the minimum threshold maintained in the Seed Set.

* The length of the bit-vector.

5. MPL Constants

This section describes various program and networking constants used by MPL.

5.1. Multicast Addresses

MPL makes use of MPL Domain Addresses to identify MPL Interfaces of an MPL Domain. By default, MPL Forwarders subscribe to the ALL_MPL_FORWARDERS multicast address with a scope value of 3 (subnet-local).

For each MPL Domain Address that an MPL Interface subscribes to, the MPL Interface MUST also subscribe to the MPL Domain Address with a scope value of 2 (link-local) when reactive forwarding is in use. MPL Forwarders use the link-scoped MPL Domain Address to communicate MPL Control Messages to neighboring (i.e. on-link) MPL Forwarders.

5.2. Message Types

MPL defines an IPv6 Option for carrying an MPL Seed Identifier and a sequence number within an MPL Data Message. The IPv6 Option Type has value MPL_OPT_TYPE.

MPL defines an ICMPv6 Message (MPL Control Message) for communicating information contained in its Seed Set and Buffered Message Set to neighboring MPL Forwarders. The MPL Control Message has ICMPv6 Type MPL_ICMP_TYPE.

5.3. MPL Forwarder Parameters

PROACTIVE_FORWARDING A boolean value that indicates whether the MPL Forwarder should schedule MPL Data Message transmissions after receiving them for the first time.

SEED_SET_LIFETIME The minimum lifetime for an entry in the Seed Set.

5.4. Trickle Parameters

As specified in [RFC6206], a Trickle timer runs for a defined interval and has three configuration parameters: the minimum interval size Imin, the maximum interval size Imax, and a redundancy constant k.

This specification defines a fourth Trickle configuration parameter, TimerExpirations, which indicates the number of Trickle timer expiration events that occur before terminating the Trickle algorithm.

Each MPL forwarder maintains a separate Trickle parameter set for MPL Data Message and MPL Control Message transmissions. The Trickle parameters are listed below:

DATA_MESSAGE_IMIN The minimum Trickle timer interval, as defined in [RFC6206], for MPL Data Message transmissions.

DATA MESSAGE_IMAX The maximum Trickle timer interval, as defined in [RFC6206], for MPL Data Message transmissions.

DATA_MESSAGE_K The redundancy constant, as defined in [RFC6206], for MPL Data Message transmissions.

DATA_MESSAGE_TIMER_EXPIRATIONS The number of Trickle timer expirations that occur before terminating the Trickle algorithm for MPL Data Message transmissions.

CONTROL_MESSAGE_IMIN The minimum Trickle timer interval, as defined in [RFC6206], for MPL Control Message transmissions.

CONTROL_MESSAGE_IMAX The maximum Trickle timer interval, as defined in [RFC6206], for MPL Control Message transmissions.

CONTROL_MESSAGE_K The redundancy constant, as defined in [RFC6206], for MPL Control Message transmissions.

CONTROL_MESSAGE_TIMER_EXPIRATIONS The number of Trickle timer expirations that occur before terminating the Trickle algorithm for MPL Control Message transmissions.

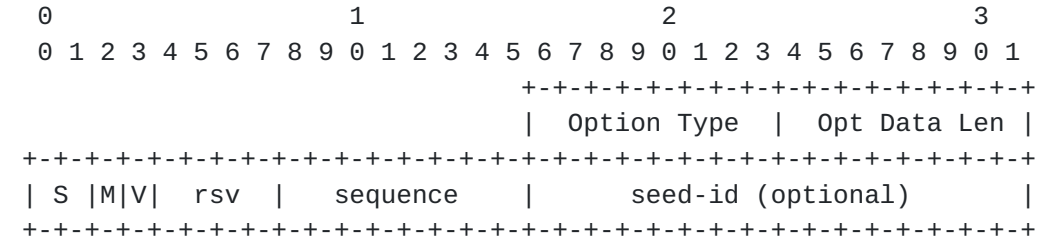
It is RECOMMENDED that all MPL Forwarder within an MPL Domain use the same values for the Trickle Parameters above, as specified in [RFC6206].

6. Protocol Message Formats

The protocol messages generated and processed by an MPL Forwarder are described in this section.

6.1. MPL Option

The MPL Option is carried in MPL Data Messages in an IPv6 Hop-by-Hop Options header, immediately following the IPv6 header. The MPL Option has the following format:



Option Type — MPL_OPT_TYPE

Opt Data Len — Length of the Option Data field in octets.

S — 2-bit unsigned integer. Identifies the length of seed-id. 0 indicates that the seed-id is the IPv6 Source Address and not included in the MPL Option. 1 indicates that the seed-id is a 16-bit unsigned integer. 2 indicates that the seed-id is a 64-bit unsigned integer. 3 indicates that the seed-id is a 128-bit unsigned integer.

M — 1-bit flag. 1 indicates that the value in sequence is known to be the largest sequence number that was received from the MPL Seed.

V — 1-bit flag. 0 indicates that the MPL Option conforms to this specification. MPL Options received in which this flag is 1 MUST be dropped.

rsv — 4-bit reserved field. MUST be set to 0 on transmission and ignored on reception.

sequence — 8-bit unsigned integer. Identifies relative ordering of MPL Data Messages from the MPL Seed identified by seed-id.

seed-id — Uniquely identifies the MPL Seed that initiated dissemination of the MPL Data Message. The size of seed-id is indicated by the S field.

The Option Data (in particular the M flag) of the MPL Option is updated by MPL Forwarders as the MPL Data Message is forwarded. Nodes that do not understand the MPL Option MUST discard the MPL Data Message. Thus, according to [RFC2460] the three high order bits of the Option Type are set to '011'. The Option Data length is variable.

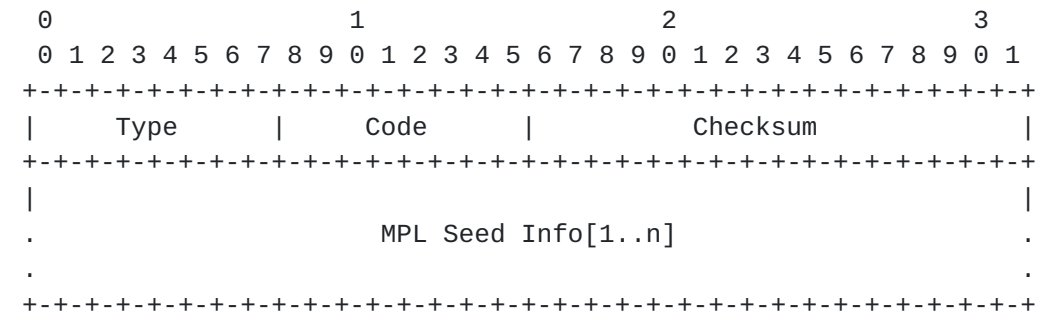
The seed-id uniquely identifies an MPL Seed. When seed-id is 128 bits (S=3), the MPL seed MAY use an IPv6 address assigned to one of its interfaces that is unique within the MPL domain. Managing MPL Seed Identifiers is not within scope of this document.

The sequence field establishes a total ordering of MPL Data Messages generated by an MPL Seed for an MPL Domain. The MPL Seed MUST increment the sequence field's value on each new MPL Data Message that it generates for an MPL Domain. Implementations MUST follow the Serial Number Arithmetic as defined in [RFC1982] when incrementing a sequence value or comparing two sequence values.

Future updates to this specification may define additional fields following the seed-id field.

6.2. MPL Control Message

An MPL Forwarder uses ICMPv6 messages to communicate information contained in its Seed Set and Buffered Message Set to neighboring MPL Forwarders. The MPL Control Message has the following format:



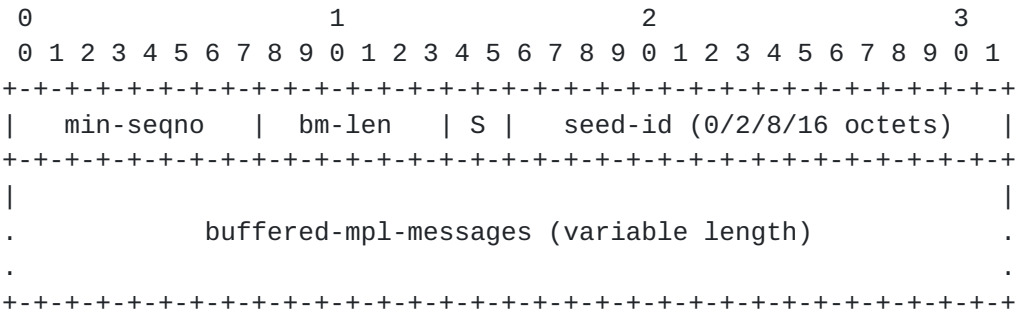
IP Fields:

Source Address	A link-local address assigned to the sending interface.
Destination Address	The link-scoped MPL Domain Address corresponding to the MPL Domain.
Hop Limit	255
ICMPv6 Fields:	
Type	MPL_ICMP_TYPE
Code	0
Checksum	The ICMP checksum. See [RFC4443].
MPL Seed Info[1..n]	List of one or more MPL Seed Info entries.

The MPL Control Message indicates the sequence numbers of MPL Data Messages that are within the Buffered Message Set. The MPL Control Message also indicates the sequence numbers of MPL Data Messages that an MPL Forwarder is willing to receive. The MPL Control Message allows neighboring MPL Forwarders to determine whether there are any new MPL Data Messages to exchange.

6.3. MPL Seed Info

An MPL Seed Info encodes the minimum sequence number for the MPL Seed maintained in the Seed Set. The MPL Seed Info also indicates the sequence numbers of MPL Data Messages generated by the MPL Seed within the Buffered Message Set. The MPL Seed Info has the following format:



min-seqno	8-bit unsigned integer. The lower-bound sequence number for the MPL Seed.
bm-len	6-bit unsigned integer. The size of buffered-mpl-messages in octets.
S	2-bit unsigned integer. Identifies the length of seed-id. 0 indicates that the seed-id value is the IPv6 Source Address and not included in the MPL Seed Info. 1 indicates that the seed-id value is a 16-bit unsigned integer. 2 indicates that the seed-id value is a 64-bit unsigned integer. 3 indicates that the seed-id is a 128-bit unsigned integer.
seed-id	Variable-length unsigned integer. Indicates the MPL Seed associated with this MPL Seed Info.
buffered-mpl-messages	Variable-length bit vector. Identifies the sequence numbers of MPL Data Messages maintained in the Buffered Message Set for the MPL Seed. The sequence number is determined by min-seqno + i, where i is the bit offset within buffered-mpl-messages.

The MPL Seed Info does not have any octet alignment requirement.

7. Information Base

7.1. Local Interface Set

The Local Interface Set records the local MPL Interfaces of an MPL Forwarder. The Local Interface Set consists of Local Interface Tuples, one per MPL Interface: (AddressSet).

AddressSet - a set of unicast addresses assigned to the MPL Interface.

7.2. Domain Set

The Domain Set records the MPL Interfaces that subscribe to each MPL Domain Address. The Domain Set consists of MPL Domain Tuples, one per MPL Domain: (MPLInterfaceSet).

MPLInterfaceSet - a set of MPL Interfaces that subscribe to the MPL Domain Address that identifies the MPL Domain.

7.3. Seed Set

The Seed Set records a sliding window used to determine the sequence numbers of MPL Data Messages that an MPL Forwarder is willing to accept generated by the MPL Seed. It consists of MPL Seed Tuples: (SeedID, MinSequence, Lifetime).

SeedID - the identifier for the MPL Seed.

MinSequence - a lower-bound sequence number that represents the sequence number of the oldest MPL Data Message the MPL Forwarder is willing to receive or transmit. An MPL Forwarder MUST ignore any MPL Data Message that has sequence value less than than MinSequence.

Lifetime - indicates the minimum lifetime of the Seed Set entry. An MPL Forwarder MUST NOT free a Seed Set entry before its expires.

7.4. Buffered Message Set

The Buffered Message Set records recently received MPL Data Messages from an MPL Seed. An MPL Forwarder uses the Buffered Message Set to buffer MPL Data Messages while the MPL Forwarder is forwarding the MPL Data Messages. The Buffered Message Set consists of Buffered Message Tuples: (SeedID, SequenceNumber, DataMessage).

SeedID - the identifier for the MPL Seed that generated the MPL Data Message.

SequenceNumber - the sequence number for the MPL Data Message.

DataMessage - the MPL Data Message.

All MPL Data Messages within the Buffered Message Set MUST have a sequence number greater than or equal to MinSequence for the corresponding SeedID. When increasing MinSequence for an MPL Seed, the MPL Forwarder MUST delete any MPL Data Messages from the Buffered Message Set that have sequence numbers less than MinSequence.

8. MPL Domains

An MPL Domain is a scope zone, as defined in [RFC4007], in which MPL Interfaces subscribe to the same MPL Domain Address and participate in disseminating MPL Data Messages.

By default, an MPL Forwarder MUST participate in an MPL Domain identified by the ALL_MPL_FORWARDERS multicast address with a scope value of 3 (subnet-local).

An MPL Forwarder MAY participate in additional MPL Domains identified by other multicast addresses. An MPL Interface MUST subscribe to the MPL Domain Addresses for the MPL Domains that it participates in. The allocation of other multicast addresses is out of scope.

For each MPL Domain Address that an MPL Interface subscribes to, the MPL Interface MUST also subscribe to the same MPL Domain Address with a scope value of 2 (link-local) when reactive forwarding is in use (i.e. when communicating MPL Control Messages).

9. MPL Seed Sequence Numbers

Each MPL Seed maintains a sequence number for each MPL Domain that it serves. The sequence numbers are included in MPL Data Messages generated by the MPL Seed. The MPL Seed MUST increment the sequence number for each MPL Data Message that it generates for an MPL Domain. Implementations MUST follow the Serial Number Arithmetic as defined in [RFC1982] when incrementing a sequence value or comparing two sequence values. This sequence number is used to establish a total ordering of MPL Data Messages generated by an MPL Seed for an MPL Domain.

10. MPL Data Messages

10.1. MPL Data Message Generation

MPL Data Messages are generated by MPL Seeds when they enter the MPL Domain. All MPL Data messages have the following properties:

- The IPv6 Source Address MUST be an address in the AddressSet of a corresponding MPL Interface and MUST be valid within the MPL Domain.

- The IPv6 Destination Address MUST be set to the MPL Domain Address corresponding to the MPL Domain.

- A MPL Data Message MUST contain an MPL Option in its IPv6 Header to identify the MPL Seed that generated the message and the ordering relative to other MPL Data Messages generated by the MPL Seed.

When the source address is in the AddressList of an MPL Interface corresponding to the MPL Domain Address and the destination address is the MPL Domain Address, the application message and the MPL Data Message MAY be identical. In other words, the MPL Data Message may contain a single IPv6 header that includes the MPL Option.

Otherwise, IPv6-in-IPv6 encapsulation MUST be used to satisfy the MPL Data Message requirements listed above [RFC2473]. The complete IPv6-in-IPv6 message forms an MPL Data Message. The outer IPv6 header conforms to the MPL Data Message requirements listed above. The encapsulated IPv6 datagram encodes the multicast data message that is communicated beyond the MPL Domain.

10.2. MPL Data Message Transmission

An MPL Forwarder manages transmission of MPL Data Messages in the Buffered Message set using the Trickle algorithm [RFC6206]. An MPL Forwarder MUST use a separate Trickle timer for each MPL Data Message that it is actively forwarding. In accordance with Section 5 of RFC 6206 [RFC6206], this document defines the following:

- This document defines a "consistent" transmission as receiving an MPL Data Message that has the same seed-id and sequence value as the MPL Data Message managed by the Trickle timer.

- This document defines an "inconsistent" transmission as receiving an MPL Data Message that has the same seed-id value and the M flag set, but has a sequence value less than MPL Data Message managed by the Trickle timer.

- This document does not define any external "events".
- This document defines MPL Data Messages as Trickle messages.
- The actions outside the Trickle algorithm that the protocol takes involve managing Seed Set and Buffered Message Set

As specified in [RFC6206], a Trickle timer has three variables: the current interval size I, a time within the current interval t, and a counter c. MPL defines a fourth variable, e, which counts the number of Trickle timer expiration events since the Trickle timer was last reset.

After DATA_MESSAGE_TIMER_EXPIRATIONS Trickle timer events, the MPL Forwarder MUST disable the Trickle timer. When a buffered MPL Data Message does not have an associated Trickle timer, the MPL Forwarder MAY delete the message from the Buffered Message Set by advancing MinSequence of the corresponding MPL Seed in the Seed Set. When the MPL Forwarder no longer buffers any messages for an MPL Seed, the MPL Forwarder MUST NOT increment MinSequence for that MPL Seed.

When transmitting an MPL Data Message, the MPL Forwarder MUST either set the M flag to zero or set it to a level that indicates whether or not the message's sequence number is the largest value that has been received from the MPL Seed.

10.3. MPL Data Message Processing

Upon receiving an MPL Data Message, the MPL Forwarder first processes the MPL Option and updates the Trickle timer associated with the MPL Data Message if one exists.

Upon receiving an MPL Data Message, an MPL Forwarder MUST perform one of the following actions:

- Accept the message and enter the MPL Data Message in the Buffered Message Set.
- Accept the message and update the corresponding MinSequence in the Seed Set to 1 greater than the message's sequence number.
- Discard the message without any change to the MPL Information Base.

If a Seed Set entry exists for the MPL Seed, the MPL Forwarder MUST discard the MPL Data Message if its sequence number is less than MinSequence or exists in the Buffered Message Set.

If a Seed Set entry does not exist for the MPL Seed, the MPL Forwarder MUST create a new entry for the MPL Seed before accepting the MPL Data Message.

If memory is limited, an MPL Forwarder SHOULD reclaim memory resources by:

- Incrementing MinSequence entries in the Seed Set and deleting MPL Data Messages in the Buffered Message Set that fall below the corresponding MinSequence value.

- Deleting other Seed Set entries that have expired and the corresponding MPL Data Messages in the Buffered Message Set.

If the MPL Forwarder accepts the MPL Data Message, the MPL Forwarder MUST perform the following actions:

- If PROACTIVE_PROPAGATION is true, the MPL Forwarder MUST initialize and start a Trickle timer for the MPL Data Message.

- If the MPL Control Message Trickle timer is not running and CONTROL_MESSAGE_TIMER_EXPIRATIONS is non-zero, the MPL Forwarder MUST initialize and start the MPL Control Message Trickle timer.

- If the MPL Control Message Trickle timer is running, the MPL Forwarder MUST reset the MPL Control Message Trickle timer.

11. MPL Control Messages

11.1. MPL Control Message Generation

An MPL Forwarder generates MPL Control Messages to communicate its Seed Set and Buffered Message Set to neighboring MPL Forwarders. Each MPL Control Message is generated according to Section 6.2, with an MPL Seed Info for each entry in Seed Set. Each MPL Seed Info entry has the following content:

- S set to the size of the seed-id field in the MPL Seed Info entry.
- min-seqno set to MinSequence of the MPL Seed.
- bm-len set to the size of buffered-mpl-messages in octets.
- seed-id set to the MPL seed identifier.
- buffered-mpl-messages with each bit representing whether or not an MPL Data Message with the corresponding sequence number exists in the Buffered Message Set. The i'th bit represents a sequence number of min-seqno + i. '0' indicates that the corresponding MPL Data Message does not exist in the Buffered Message Set. '1' indicates that the corresponding MPL Data Message does exist in the Buffered Message Set.

11.2. MPL Control Message Transmission

An MPL Forwarder transmits MPL Control Messages using the Trickle algorithm. A MPL forwarder maintains a single Trickle timer for each MPL Domain. When CONTROL_MESSAGE_TIMER_EXPIRATIONS is 0, the MPL Forwarder does not execute the Trickle algorithm and does not transmit MPL Control Messages. In accordance with Section 5 of RFC 6206 [RFC6206], this document defines the following:

- This document defines a "consistent" transmission as receiving an MPL Control Message that indicates neither the receiving nor transmitting node has new MPL Data Message.
- This document defines an "inconsistent" transmission as receiving an MPL Control Message that indicates either the receiving or transmitting node has at least one new MPL Data Message to offer.
- This document defines an "event" as increasing MinSequence of any entry in the Seed Set or adding a message to the Buffered Message Set.

- This document defines an MPL Control Message as a Trickle message.

As specified in [RFC6206], a Trickle timer has three variables: the current interval size I, a time within the current interval t, and a counter c. MPL defines a fourth variable, e, which counts the number of Trickle timer expiration events since the Trickle timer was last reset. After CONTROL_MESSAGE_TIMER_EXPIRATIONS Trickle timer events, the MPL Forwarder MUST disable the Trickle timer.

11.3. MPL Control Message Processing

An MPL Forwarder processes each MPL Control Message that it receives to determine if it has any new MPL Data Messages to receive or offer.

An MPL Forwarder determines if a new MPL Data Message has not been received from a neighboring node if any of the following conditions hold true:

- The MPL Control Message includes an MPL Seed that does not exist in the Seed Set.

- The MPL Control Message indicates that the neighbor has an MPL Data Message in its Buffered Message Set with sequence number greater than MinSequence (i.e. the i-th bit is set to 1 and min-seqno + i > MinSequence) and is not included in the MPL Forwarder's Buffered Message Set.

When an MPL Forwarder determines that it has not yet received an MPL Data Message buffered by a neighboring device, the MPL Forwarder MUST reset its Trickle timer associated with MPL Control Message transmissions. If an MPL Control Message Trickle timer is not running, the MPL Forwarder MUST initialize and start a new Trickle timer.

An MPL Forwarder determines if an MPL Data Message in the Buffered Message Set has not yet been received by a neighboring MPL Forwarder if any of the following conditions hold true:

- The MPL Control Message does not include an MPL Seed for the MPL Data Message.

- The MPL Data Message's sequence number is greater than or equal to min-seqno and not included in the neighbor's Buffered Message Set (i.e. the MPL Data Message's sequence number does not have a corresponding bit in buffered-mpl-messages set to 1).

When an MPL Forwarder determines that it has at least one MPL Data Message in its Buffered Message Set that has not yet been received by

a neighbor, the MPL Forwarder MUST reset the MPL Control Message Trickle timer. Additionally, for each of those entries in the Buffered Message Set, the MPL Forwarder MUST reset the Trickle timer and reset e to 0. If a Trickle timer is not associated with the MPL Data Message, the MPL Forwarder MUST initialize and start a new Trickle timer.

12. Acknowledgements

The authors would like to acknowledge the helpful comments of Robert Cragie, Esko Dijk, Ralph Droms, Paul Duffy, Ulrich Herberg, Owen Kirby, Joseph Reddy, Don Sturek, Dario Tedeschi, and Peter van der Stok, which greatly improved the document.

13. IANA Considerations

This document defines one IPv6 Option, a type that must be allocated from the IPv6 "Destination Options and Hop-by-Hop Options" registry of [RFC2780].

This document defines one ICMPv6 Message, a type that must be allocated from the "ICMPv6 "type" Numbers" registry of [RFC4443].

This document registers two well-known multicast addresses from the IPv6 multicast address space.

13.1. MPL Option Type

IANA is requested to allocate an IPv6 Option Type from the IPv6 "Destination Options and Hop-by-Hop Options" registry of [RFC2780], as specified in Table 1 below:

Mnemonic	act	chg	rest	Description	Reference
MPL_OPT_TYPE	01	1	TBD (suggested value 01101)	MPL Option	This Document

Table 1: IPv6 Option Type Allocation

13.2. MPL ICMPv6 Type

IANA is requested to allocate an ICMPv6 Type from the "ICMPv6 "type" Numbers" registry of [RFC4443], as specified in Table 2 below:

Mnemonic	Type	Name	Reference
MPL_ICMP_TYPE	TBD	MPL Control Message	This Document

Table 2: IPv6 Option Type Allocation

13.3. Well-known Multicast Addresses

IANA is requested to allocate an IPv6 multicast address "ALL_MPL_FORWARDERS" from the "Variable Scope Multicast Addresses" sub-registry of the "INTERNET PROTOCOL VERSION 6 MULTICAST ADDRESSES" registry.

14. Security Considerations

MPL uses sequence numbers to maintain a total ordering of MPL Data Messages from an MPL Seed. The use of sequence numbers allows a denial-of-service attack where an attacker can spoof a message with a sufficiently large sequence number to: (i) flush messages from the Buffered Message List and (ii) increase the MinSequence value for an MPL Seed in the Seed Set. The former side effect allows an attacker to halt the forwarding process of any MPL Data Messages being disseminated. The latter side effect allows an attacker to prevent MPL Forwarders from accepting new MPL Data Messages that an MPL Seed generates while the sequence number is less than MinSequence.

More generally, the basic ability to inject messages into a Low-power and Lossy Network can be used as a denial-of-service attack regardless of what forwarding protocol is used. For these reasons, Low-power and Lossy Networks typically employ link-layer security mechanisms to disable an attacker's ability to inject messages.

To prevent attackers from injecting packets through an MPL Forwarder, the MPL Forwarder MUST NOT accept or forward MPL Data Messages from a communication interface that does not subscribe to the MPL Domain Address identified in message's destination address.

MPL uses the Trickle algorithm to manage message transmissions and the security considerations described in [RFC6206] apply.

15. Normative References

[RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, August 1996.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, March 2005.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.

[RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

Authors' Addresses

Jonathan W. Hui
Cisco
170 West Tasman Drive
San Jose, California 95134
USA

Phone: +408 424 1547
Email: jonhui@cisco.com

Richard Kelsey
Silicon Labs
25 Thomson Place
Boston, Massachusetts 02210
USA

Phone: +617 951 1225
Email: richard.kelsey@silabs.com