       Multicast Protocol for Low power and Lossy Networks (MPL)
                 draft-ietf-roll-trickle-mcast-08

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 September 29, 2014.

Table of Contents

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 (e.g.
   [RFC3973] [RFC4601]). However, maintaining such topologies in Low
   power and Lossy Networks (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 MPL. An 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 within
                         an MPL domain. An MPL Data Message contains an
                         MPL Option in the IPv6 header and has as its
                         destination address the MPL Domain Address
                         corresponding to the MPL Domain.

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.

MPL Seed Identifier   - An unsigned integer that uniquely identifies an
                         MPL Seed within an MPL Domain.

3. Applicability Statement

MPL is an IPv6 multicast forwarding protocol designed for the
communication characteristics and resource constraints of Low-Power
and Lossy Networks. By implementing controlled disseminations of
multicast messages using the Trickle algorithm, MPL is designed for
networks that communicate using low-power and lossy links with widely
varying topologies in both the space and time dimensions.

While designed specifically for Low-Power and Lossy Networks, MPL is
not limited to use over such networks. MPL may be applicable to any
network where no multicast routing state is desired. MPL may also be
used in environments where only a subset of links are considered Low-
Power and Lossy links.

A host need not be aware that their multicast is supported by MPL as
long as its attachment router forwards multicast messages between the
MPL Domain and the host. However, a host may choose to implement MPL
so that it can take advantage of the broadcast medium inherent in
many Low-Power and Lossy Networks and receive multicast messages
carried by MPL directly.

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. 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 SHOULD participate in an MPL Domain
identified by the ALL_MPL_FORWARDERS multicast address with a scope
value of 3 (Realm-Local) [I-D.ietf-6man-multicast-scopes].

When MPL is used in deployments that use administratively defined
scopes that cover, for example, multiple subnets based on different
underlying network technologies, Admin-Local scope (scop value 4) or
Site-Local scope (scop value 5) SHOULD be used.

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 assignment 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).

4.2. Information Base Overview

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

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

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

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

o A Buffered Message Set records recently received MPL Data Messages
   from an MPL Seed within an MPL Domain. Each MPL Domain has an
   associated Buffered Message Set. MPL Data Messages resident in a
   Buffered Message Set have sequence numbers that are greater than
   or equal to the minimum threshold maintained in the corresponding
   Seed Set. MPL uses Buffered Message Sets to store MPL Data
   Messages that may be transmitted by the MPL Forwarder for
   forwarding.

4.3. Protocol 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:

o When having a multicast message to forward into an MPL Domain, the
   MPL Seed generates an MPL Data Message that includes the MPL
   Domain Address as the IPv6 Destination Address, 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 an MPL Data Message, preserving the original
   IPv6 Destination Address.

o 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 MPL Domain's 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 an MPL Domain's Seed Set and Buffered Message Set.

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 via MPL Interfaces.

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.

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 an MPL Data Message, the MPL Forwarder schedules those
   MPL Data Messages for transmission using the Trickle algorithm.

Note that the use of proactive and reactive forwarding strategies
within the same MPL Domain are not mutually exclusive and may be used
simultaneously. For example, upon receiving a new MPL Data Message
when both proactive and reactive forwarding techniques are enabled,
an MPL Forwarder will proactively retransmit the MPL Data Message a

limited number of times and schedule further transmissions upon
receiving MPL Control Messages.

4.4. Signaling Overview

MPL 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 an MPL Domain's 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 Parameters and Constants

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

5.1. MPL 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
[I-D.ietf-6man-multicast-scopes].

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. MPL 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 0x6D.

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

5.3. MPL Seed Identifiers

MPL uses MPL Seed Identifiers to uniquely identify MPL Seeds within
an MPL Domain. For each MPL Domain that the MPL Forwarder serves as
an MPL Seed, the MPL Forwarder MUST have an associated MPL Seed
Identifier. An MPL Forwarder MAY use the same MPL Seed Identifier
across multiple MPL Domains, but the MPL Seed Identifier MUST be
unique within each MPL Domain. The mechanism for assigning and
verifying uniqueness of MPL Seed Identifiers is not specified in this
document.

5.4. MPL Parameters

PROACTIVE_FORWARDING A boolean value that indicates whether the MPL
   Forwarder schedules MPL Data Message transmissions after receiving
   them for the first time. PROACTIVE_FORWARDING has a default value
   of TRUE. The mechanism for setting PROACTIVE_FORWARDING is not
   specified within this document.

SEED_SET_ENTRY_LIFETIME The minimum lifetime for an entry in the
   Seed Set. SEED_SET_ENTRY_LIFETIME has a default value of 30
   minutes. It is RECOMMENDED that all MPL Forwarders use the same
   value for SEED_SET_ENTRY_LIFETIME for a given MPL Domain and use a
   default value of 30 minutes. The mechanism for setting
   SEED_SET_ENTRY_LIFETIME is not specified within this document.

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
for a given MPL Data Message or MPL Control Message.

Each MPL Interface uses the following Trickle parameters for MPL Data
Message and MPL Control Message transmissions.

DATA_MESSAGE_IMIN The minimum Trickle timer interval, as defined in
    [RFC6206], for MPL Data Message transmissions. DATA_MESSAGE_IMIN
    has a default value of 10 times the expected link-layer latency.

DATA MESSAGE_IMAX The maximum Trickle timer interval, as defined in
    [RFC6206], for MPL Data Message transmissions. DATA_MESSAGE_IMAX
    has a default value equal to DATA_MESSAGE_IMIN.

DATA_MESSAGE_K The redundancy constant, as defined in [RFC6206], for
    MPL Data Message transmissions. DATA_MESSAGE_K has a default
    value of 1.

DATA_MESSAGE_TIMER_EXPIRATIONS The number of Trickle timer
    expirations that occur before terminating the Trickle algorithm's
    retransmission of a given MPL Data Message.
    DATA_MESSAGE_TIMER_EXPIRATIONS has a default value of 3.

CONTROL_MESSAGE_IMIN The minimum Trickle timer interval, as defined
    in [RFC6206], for MPL Control Message transmissions.
    CONTROL_MESSAGE_IMIN has a default value of 10 times the worst-
    case link-layer latency.

CONTROL_MESSAGE_IMAX The maximum Trickle timer interval, as defined
    in [RFC6206], for MPL Control Message transmissions.
    CONTROL_MESSAGE_IMAX has a default value of 5 minutes.

CONTROL_MESSAGE_K The redundancy constant, as defined in [RFC6206],
    for MPL Control Message transmissions. CONTROL_MESSAGE_K has a
    default value of 1.

CONTROL_MESSAGE_TIMER_EXPIRATIONS The number of Trickle timer
    expirations that occur before terminating the Trickle algorithm
    for MPL Control Message transmissions.
    CONTROL_MESSAGE_TIMER_EXPIRATIONS has a default value of 10.

Following [RFC6206], it is RECOMMENDED that all MPL Interfaces
attached to the same link of a given MPL Domain use the same values

for the Trickle Parameters above for a given MPL Domain. The
mechanism for setting the Trickle Parameters is not specified within
this document.

The default MPL parameters specify a forwarding strategy that
utilizes both proactive and reactive techniques. Using these default
values, an MPL Forwarder proactively transmits any new MPL Data
Messages it receives then uses MPL Control Messages to trigger
additional MPL Data Message retransmissions where message drops are
detected. Setting DATA_MESSAGE_IMAX to the same as DATA_MESSAGE_IMIN
in this case is acceptable since subsequent MPL Data Message
retransmissions are triggered by MPL Control Messages, where
CONTROL_MESSAGE_IMAX is greater than CONTROL_MESSAGE_IMIN.

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:

```
 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
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                | Option Type   | Opt Data Len  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | S |M|V|  rsv  |   sequence    |      seed-id (optional)     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Type   0x6D.

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 Data
                    Messages with an MPL Option 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 an MPL Domain's Seed Set and Buffered Message Set to
neighboring MPL Forwarders. The MPL Control Message has the
following format:

```
 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      |     Code      |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                     MPL Seed Info[0..n]                       .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IP Fields:

Source Address   An IPv6 address in the AddressSet of the
                      corresponding MPL Interface and MUST be valid
                      within the MPL Domain.

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[0..n] List of zero or more MPL Seed Info entries.

The MPL Control Message indicates the sequence numbers of MPL Data
Messages that are within the MPL Domain's 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 an MPL Seed
maintained in the MPL Domain's Seed Set. The MPL Seed Info also
indicates the sequence numbers of MPL Data Messages generated by the
MPL Seed that are stored within the MPL Domain's Buffered Message
Set. The MPL Seed Info has the following format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   min-seqno   |  bm-len   | S |    seed-id (0/2/8/16 octets)  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   .              buffered-mpl-messages (variable length)          .
   .                                                               .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   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 corresponding Buffered Message Set for the
                      MPL Seed. 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.

   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

   A 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. An MPL Forwarder maintains a Seed
   Set for each MPL Domain that it participates in. A Seed Set 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 remaining lifetime of the Seed Set
      entry. An MPL Forwarder MUST NOT free a Seed Set entry before the
      remaining lifetime expires.

7.4. Buffered Message Set

   A Buffered Message Set records recently received MPL Data Messages
   from an MPL Seed within an MPL Domain. An MPL Forwarder uses a
   Buffered Message Set to buffer MPL Data Messages while the MPL
   Forwarder is forwarding the MPL Data Messages. An MPL Forwarder
   maintains a Buffered Message Set for each MPL Domain that it
   participates in. A 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 a 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
corresponding Buffered Message Set that have sequence numbers less
than MinSequence.

8. 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.

9. MPL Data Messages

9.1. MPL Data Message Generation

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

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

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

o An 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 destination address is an MPL Domain Address and the source
address is in the AddressLIst of an MPL Interface that belongs to
that 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.

9.2. MPL Data Message Transmission

An MPL Forwarder manages transmission of MPL Data Messages in its
Buffered Message Sets 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:

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

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

o This document does not define any external "events".

o This document defines MPL Data Messages as Trickle messages.

o The actions outside the Trickle algorithm that the protocol takes
   involve managing the MPL Domain's 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.

9.3. 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:

   o Accept the message and enter the MPL Data Message in the MPL
      Domain's Buffered Message Set.

   o Accept the message and update the corresponding MinSequence in the
      MPL Domain's Seed Set to 1 greater than the message's sequence
      number.

   o 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:

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

   o 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:

   o Reset the Lifetime of the corresponding Seed Set entry to
      SEED_SET_ENTRY_LIFETIME.

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

o 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.

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

10. MPL Control Messages

10.1. MPL Control Message Generation

An MPL Forwarder generates MPL Control Messages to communicate an MPL
Domain's 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 the MPL Domain's
Seed Set. Each MPL Seed Info entry has the following content:

o S set to the size of the seed-id field in the MPL Seed Info entry.

o min-seqno set to MinSequence of the MPL Seed.

o bm-len set to the size of buffered-mpl-messages in octets.

o seed-id set to the MPL seed identifier.

o 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.

10.2. MPL Control Message Transmission

An MPL Forwarder transmits MPL Control Messages using the Trickle
algorithm. An 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:

o This document defines a "consistent" transmission as receiving an
   MPL Control Message that results in a determination that neither
   the receiving nor transmitting node has any new MPL Data Messages
   to offer.

   o This document defines an "inconsistent" transmission as receiving
      an MPL Control Message that results in a determination that either
      the receiving or transmitting node has at least one new MPL Data
      Message to offer.

   o The Trickle timer is reset in response to external "events." This
      document defines an "event" as increasing MinSequence of any entry
      in the corresponding Seed Set or adding a message to the
      corresponding Buffered Message Set.

   o 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.

10.3. 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:

   o The MPL Control Message includes an MPL Seed that does not exist
      in the MPL Domain's Seed Set.

   o 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 Domain'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:

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

   o The MPL Data Message's sequence number is greater than or equal to
      min-seqno and not included in the neighbor's corresponding
      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 corresponding 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.

11. Acknowledgements

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

12. 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 a well-known multicast address from the
   Variable Scope Multicast Address registry.

12.1. 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:

```
+-----------+-----+-----+-------+------------+--------------+
| Hex Value | act | chg | rest  | Description|  Reference   |
+-----------+-----+-----+-------+------------+--------------+
|    0x6D   | 01  |  1  | 01101 | MPL Option | This Document|
+-----------+-----+-----+-------+------------+--------------+
```

Table 1: IPv6 Option Type Allocation

12.2. 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:

```
+------+---------------------+---------------+
| Type |         Name        |   Reference   |
+------+---------------------+---------------+
| TBD  | MPL Control Message | This Document |
+------+---------------------+---------------+
```

Table 2: IPv6 Option Type Allocation

In this document, the mnemonic MPL_ICMP_TYPE was used to refer to the
ICMPv6 Type above, which is TBD by IANA.

12.3. Well-known Multicast Addresses

IANA is requested to allocate an IPv6 multicast address, with Group
ID in the range [0x01,0xFF] for 6LoWPAN compression [RFC6282],
"ALL_MPL_FORWARDERS" from the "Variable Scope Multicast Addresses"
sub-registry of the "IPv6 Multicast Address Space" registry [RFC3307]
as specified in Table 3 below:

```
+---------------------+-------------------+-----------+------------+
|     Address(s)      |    Description    | Reference |    Date    |
|                     |                   |           | Registered |
+---------------------+-------------------+-----------+------------+
| FF0X:0:0:0:0:0:0:FC | ALL_MPL_FORWARDERS|    This   | 2013-04-10 |
|                     |                   | Document  |            |
+---------------------+-------------------+-----------+------------+
```

Table 3: Variable Scope Multicast Address Allocation

13. 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 corresponding 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.

14. References

14.1. Normative References

[I-D.ietf-6man-multicast-scopes]
           Droms, R., "IPv6 Multicast Address Scopes", draft-ietf-
           6man-multicast-scopes-03 (work in progress), February
           2014.

[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.

   [RFC3307] Haberman, B., "Allocation Guidelines for IPv6 Multicast
              Addresses", RFC 3307, August 2002.

   [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.

   [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6
              Datagrams over IEEE 802.15.4-Based Networks", RFC 6282,
              September 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.

14.2. Informative References

   [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol
              Independent Multicast - Dense Mode (PIM-DM): Protocol
              Specification (Revised)", RFC 3973, January 2005.

   [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas,
              "Protocol Independent Multicast - Sparse Mode (PIM-SM):
              Protocol Specification (Revised)", RFC 4601, August 2006.

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