ROLL
Internet-Draft
Intended status: Standards Track
Expires: October 13, 2011

J. Hui
Cisco
R. Kelsey
Ember Corporation

April 11, 2011

# Multicast Forwarding Using Trickle
# draft-ietf-roll-trickle-mcast-00

## Abstract

Low power and Lossy Networks (LLNs) are typically composed of resource constrained nodes communicating over links that have dynamic characteristics. Memory constraints coupled with temporal variations in link connectivity makes the use of topology maintenance to support IPv6 multicast challenging. This document describes the use of Trickle to efficiently forward multicast messages without the need for topology maintenance.

## Status of this Memo



## Copyright Notice

Table of Contents

## 1. Introduction

The resource constraints of Low power and Lossy Networks (LLNs) may preclude the use of existing IPv6 multicast forwarding mechanisms. Such networks are typically constrained in resources (limited channel capacity, processing power, energy capacity, memory). In particular memory constraints may limit nodes to maintaining state for only a small subset of neighbors. Limited channel and energy capacity require protocols to remain efficient and robust even in dense topologies.

Traditional IP multicast forwarding typically relies on topology maintenance mechanisms to efficiently forward multicast messages to the intended destinations. In some cases, topology maintenance involves maintaining multicast trees to reach all subscribers of a multicast group. Maintaining such topologies is difficult especially when memory constraints are such that nodes can only maintain a default route. Dynamic properties of wireless networks can make control traffic 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, which is more state than a LLN node may be able to handle.

This document describes the use of Trickle for IPv6 multicast forwarding in LLNs. Trickle provides a mechanism for controlled, density-aware flooding without the need to maintain a forwarding topology [I-D.levis-roll-trickle].

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. Terminology

Trickle Multicast Message
: A IPv6 multicast datagram that includes a Trickle Multicast option in the IPv6 Hop-by-Hop Options header.

Trickle Multicast Forwarder
: A IPv6 router that can process a Trickle Multicast option and follows the forwarding rules specified in this document.

Trickle Multicast Domain
: An administrative domain that defines the scope of Trickle dissemination. All routers within a Trickle Multicast Domain participate in the same dissemination process.

Seed
: The router that starts the dissemination process for a Trickle multicast message. The Seed may be different than the node identified by the IPv6 Source address of the multicast message.

## 3. Overview

Trickle multicast forwarding implements a controlled, density-aware flood to disseminate a IPv6 multicast message to all nodes within a Trickle Multicast Domain. The basic process is similar to traditional flooding - nodes forward newly received multicast messages using link-layer broadcasts. Nodes maintain state of recently received multicast messages to detect duplicates and ensure that each node receives at most one copy of each multicast message.

Each Trickle multicast message carries a Trickle Multicast option that includes a SeedID and Sequence value. The SeedID uniquely identifies the Seed that initiated the message's dissemination process within the Trickle Multicast Domain. Note that the Seed does not have to be the same node as the message's source. It is possible to tunnel a multicast message to a Seed node and start the dissemination process from a different node within the Trickle Multicast Domain.

The Sequence value establishes a total ordering of multicast messages disseminated by SeedID. Nodes maintain a sliding window of recently received multicast messages for each SeedID. The sliding window establishes what messages can be received and ensure at most one copy of each multicast message is received. Messages with sequence values lower than the lower bound of the window MUST be ignored. Messages with sequence values stored within the sliding window MUST be ignored. All other messages MUST be received, advancing the sliding window if necessary. Larger sequence values always take precedence. The sliding window can be of variable size, trading memory requirements for reliability of disseminating multiple messages simultaneously.

Trickle's density-aware properties come from its suppression mechanism. When suppression is enabled, nodes periodically advertise a summary of recently received multicast messages. These advertisements allow nodes to determine if they have any additional multicasts to offer to neighboring nodes. A multicast message is only retransmitted upon receiving positive indication that a neighbor has not yet received that multicast message.

Nodes suppress advertisement transmissions and multicast retransmissions after recently receiving "consistent" advertisements. A node determines that a neighbor's advertisement is "consistent" when neither node has new multicast messages to offer to the other. The suppression reduces the number of redundant transmissions and is what allows Trickle to maintain low channel utilization in dense environments. However, suppression trades low control overhead for longer propagation times. When using suppression, Trickle's

propagation times often have a long-tail distribution.

Trickle provides an adaptive timer, called the Trickle timer. When receiving an "inconsistent" advertisement, nodes reset the Trickle timer period to a small period so that dissemination happens quickly. The Trickle timer period doubles when the period expires and no "inconsistent" advertisements have been received, reducing control overhead when the network is in a consistent state.

This document does allow configurations that disable the suppression mechanism, reducing Trickle Multicast Forwarding to simple flooding. This can be done by setting the suppression threshold for received "consistent" advertisements to infinity. In this mode, Trickle advertisements are not sent since consistency checks are not performed. Instead, nodes simply retransmit multicast messages they are trying to forward.

## 4. Trickle Multicast Parameters

All Trickle multicast forwarders within a Trickle multicast domain MUST be configured with two sets of configurations (one for each value of the M flag). Each configuration has five parameters:

Imin — The minimum Trickle timer interval as defined in [I-D.levis-roll-trickle].

Imax — The maximum Trickle timer interval as defined in [I-D.levis-roll-trickle].

k — The redundancy constant as defined in [I-D.levis-roll-trickle].

Tactive — The duration that a multicast forwarder can attempt to forward a multicast message. Specified in units of Imax.

Tdwell — The duration that a multicast forwarder must maintain sliding window state for SeedID after receiving the last multicast message from SeedID. Specified in units of Imax.

Tactive specifies the time duration that a node may retransmit a multicast message in attempt to forward it to neighboring nodes. Larger values of Tactive increases the number of retransmissions and overall dissemination reliability.

Tdwell specifies the time duration for maintaining sliding window state to ensure that a multicast message from SeedID is received at most once. Larger values of Tdwell decreases the likelihood that a node will receive a multicast message more than once.

The specific values are left out of scope of this document as they are dependent on link-specific properties. How those parameters are configured are also left out of scope.

The Trickle multicast parameters allow both aggressive and conservative multicast forwarding strategies. For example, an aggressive strategy may specify each multicast forwarder to retransmit any newly received message 3 times on a short fixed period and maintain state for 12 retransmission periods to avoid receiving duplicate messages. This aggressive policy can be specified using a Trickle parameter set of Imin = Imax = 100ms, k = infinity, Tactive = 3, and Tdwell = 12. Setting k to infinity disables the Trickle suppression mechanism.

A conservative multicast forwarding strategy utilizes Trickle suppression and a larger Imax value to minimize redundant transmissions.  One such conservative policy is a Trickle parameter set of Imin = 100ms, Imax = 30min, k = 1, Tactive = 3, and Tdwell = 12.

## 5. Message Formats

### 5.1. Trickle Multicast Option

The Trickle Multicast option is carried in an IPv6 Hop-by-Hop Options header, immediately following the IPv6 header. The Trickle Multicast 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 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |       SeedID (optional)       |M|          Sequence           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Type — XX (to be confirmed by IANA).

Opt Data Len — Length of the Option Data field in octets. MUST be set to either 2 or 4.

SeedID — Uniquely identifies a Trickle multicast seed that initiated the dissemination process. The SeedID field is optional and only appears when Opt Data Len is set to 4. When Opt Data Len is set to 2, the SeedID is equivalent to the IPv6 Source address.

M — Mode flag. Identifies one of two Trickle parameters to use when forwarding this multicast message.

Sequence — Identifies relative ordering of multicast messages from the source identified by SeedID.

The Option Data of the Trickle Multicast option MUST NOT change en-route. Nodes that do not understand the Trickle Multicast option MUST skip over this option and continue processing the header. Thus, according to [RFC2460] the three high order bits of the Option Type must be equal set to zero. The Option Data length is variable.

The SeedID uniquely identifies a Trickle multicast seed within the Trickle multicast domain. The SeedID field may either be an IPv6 address assigned to the seed node or a managed 16-bit value. In either case, the SeedID MUST be unique within the Trickle multicast domain. Managing the SeedID namespace is left out of scope.

The M flag identifies one of two Trickle parameters to use when forwarding the message. This capability allows a Trickle Multicast Domain to support two different Trickle parameter sets that make different propagation time vs. control overhead trade-offs.

Sequence establishes a relative ordering of multicast messages from the same SeedID. The source MUST increment the Sequence value when sourcing a new Trickle multicast message. Implementations MUST follow the Serial Number Arithmetic as defined in [RFC1982].

## 5.2. Trickle ICMPv6 Message

The Trickle ICMP message is used to advertise metadata for recently received Trickle multicast messages. The Trickle ICMP 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                       Sequence List[1..n]                     .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IP Fields:

Source Address      A link-local address assigned to the sending interface.

Destination Address The link-local all-nodes (FF02::1) or link-local all-routers (FF02::2) multicast address.

Hop Limit           255

ICMP Fields:

Type                XX (to be confirmed by IANA).

Code                0

Checksum            The ICMP checksum. See [RFC4443].

Sequence List[1..n] List of zero, one, or more Sequence Lists (defined in Section 5.2.1).

The Trickle ICMP message advertises sliding windows maintained by the multicast forwarder. The advertisement serves to notify neighbors of newer messages that it can propagate or has yet to receive. Only entries for messages where Tactive has not expired are included in the ICMP message. The sliding windows are encoded using a Sequence List, defined in Section 5.2.1.

### 5.2.1. Sequence List

A Sequence List contains a list of Sequence values for a SeedID. Each Sequence List 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
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |S|M|    rsv    |    SeqLen     |    SeedID (2 or 16 octets)    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                               |
  .                      Sequence[1..SeqLen]                      .
  .                                                               .
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

S                   Indicates length of SeedID. When set to 0, SeedID is 16 octets. When set to 1, SeedID is 2 octets.

M                   Indicates one of two Trickle parameter sets used for disseminating multicast messages.

SeqLen              Number of 2-octet Sequence entries.

SeedID              Copied from a recently received Trickle Multicast option.

Sequence[1..SeqLen] List of recently received Sequence values from SeedID. Note that the Sequence value is only 15 bits and the highest order bit MUST be set to 0.

## 6. Trickle Multicast Forwarder Behavior

A Trickle Multicast Forwarder implementation needs to manage sliding windows and Trickle timers. These mechanisms are used to determine when received messages should be accepted, when ICMP messages are transmitted, and when multicast messages are retransmitted.

### 6.1. Managing Sliding Windows

Every Trickle multicast forwarder MUST maintain a sliding window of Sequence values for each SeedID that generated recently received multicast messages.

When receiving a Trickle multicast message, if no existing sliding window exists for the SeedID, a new sliding window MUST be created before accepting the message. If memory constraints are such that a new sliding window cannot be created, then the message must be ignored.

If a sliding window exists for the SeedID, the message must be ignored if the message's Sequence value falls below the lower bound of the window or appears in the list of stored Sequence values within the window. All other messages MUST be received.

When receiving a message, the sliding window MUST be updated with the message's Sequence value. If the Sequence value is larger than the upper bound of the window, the new message establishes the new upper bound.

Memory constraints may limit the total number of Sequence values that can be stored. An entry may be reclaimed before the dwell time expires if it serves as the lower bound of the window and the window has more than one entry. Note that entries can be reclaimed from sliding windows for other SeedIDs.

When only one entry for a sliding window remains, that entry MUST NOT be reclaimed until its dwell timer expires. Maintaining the largest sequence value received from a SeedID ensures that earlier messages are received at most once.

### 6.2. Trickle Timers

A Trickle multicast forwarder maintains two Trickle timers parameterized on the S flag. The Trickle timer is maintained as described in [I-D.levis-roll-trickle].

When suppression is enabled (i.e. k is finite), a Trickle transmission event consists of transmitting a Trickle ICMP message.

If an "inconsistent" advertisement was received during that period, multicast messages that caused the inconsistency are also retransmitted.

When suppression is disabled (i.e. k is infinite), a Trickle transmission event consists of transmitting multicast messages that have been received within the Tactive time window.

This document defines receiving a "consistent" transmission as receiving a Trickle ICMP message that indicates neither the receiving nor transmitting node has new multicast messages to offer.

This document defines receiving an "inconsistent" transmission as receiving a Trickle ICMP message that indicates either receiving or transmitting node has a new multicast message to offer. An "inconsistent" transmission also includes receiving a new multicast message.

### 6.3. Trickle Multicast Option Processing

All IPv6 datagrams containing a Trickle Multicast option MUST have a multicast IPv6 Destination address. If the IPv6 Destination is not a multicast address, the multicast forwarder MUST drop the datagram.

A multicast forwarder MUST drop the multicast message if it cannot ensure that the message has never been received before. This occurs when the Sequence value is below the lower bound of the sliding window for SeedID or when an entry already exists for the Sequence value.

If no sliding window state for SeedID exists, the multicast forwarder MUST allocate a new sliding window for the SeedID before accepting the message. If a sliding window cannot be allocated, the forwarder MUST drop the message.

Upon accepting the message, the forwarder MUST enter the sequence value in the sliding window and decrement the IPv6 Hop Limit. If the IPv6 Hop Limit is non-zero, the forwarder MUST buffer the message for retransmission for the duration specified by Tactive.

### 6.4. Trickle ICMP Processing

Processing a Trickle ICMP message involves determining if either the receiver or transmitter has new multicast messages to offer.

The transmitter has new multicast messages to offer if any (SeedID, Sequence) pair falls within an existing sliding window for SeedID but does not have an associated entry.

The transmitter has new multicast messages to offer if the (SeedID, Sequence) pair is great than the upper bound of an existing sliding window for SeedID.

The receiver has new multicast messages to offer if any buffered messages are not listed in the Trickle ICMP message and the Trickle ICMP message contains a (SeedID, Sequence) pair for a prior multicast message.

The receiver has a new multicast message to offer if any buffered messages does not have an associated SeedID entry in the Trickle ICMP message.

If neither receiver nor transmitter has new multicast messages to offer, the multicast forwarder logs a consistent event by incrementing c, as described in [I-D.levis-roll-trickle].

If either receiver or transmitter has new multicast messages to offer, the multicast forwarder logs an inconsistent event by resetting Trickle timer T[M], as described in [I-D.levis-roll-trickle]. All new messages that the receiver can offer MUST be scheduled for transmission at the next transmission event. Note that these transmissions may be suppressed if the transmission event is suppressed.

## 7. Acknowledgements

TODO.

## 8. IANA Considerations

The Trickle Multicast option requires an IPv6 Option Number.

| HEX | act | chg | rest |
|---|---|---|---|
| C | 00 | 0 | 01100 |

The first two bits indicate that the IPv6 node may skip over this option and continue processing the header if it doesn't recognize the option type, and the third bit indicates that the Option Data MUST NOT change en-route.

## 9. Security Considerations

TODO.

# 10. References

## 10.1. Normative References

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-19 (work in progress), March 2011.

[I-D.levis-roll-trickle]
Levis, P. and T. Clausen, "The Trickle Algorithm", draft-levis-roll-trickle-00 (work in progress), February 2010.

[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.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[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.

[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.

## 10.2. Informative References

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-05 (work in progress), March 2011.

## 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
Ember Corporation
47 Farnsworth Street
Boston, Massachusetts  02210
USA

Phone: +617 951 1225
Email: kelsey@ember.com