Multicast Forwarding Using Trickle

(draft-hui-6man-trickle-mcast-00)

Jonathan Hui
Richard Kelsey

ROLL WG Meeting
78th IETF Meeting
Maastricht, Netherlands

Outline

- Motivation
- Overview
- Applying Trickle

Motivation

- Problem: LLN resource constraints may preclude use of existing multicast forwarding mechanisms
    - Multicast trees
    - Connected dominating set
    - Uncontrolled flood

- Solution: Trickle
    - Implements a controlled, density-aware flood to disseminate IPv6 multicast message to all nodes

Overview

- Disseminate multicast message from a Seed
  - Seed does not have to be actual source of datagram
  - Source may tunnel datagram to a Seed to initiate dissemination
- Each multicast message carries in hop-by-hop:
  - SeedID: uniquely identifies node that initiates dissemination
  - Sequence: establishes a total ordering of multicast messages from SeedID

Duplicate Suppression

- Maintain sliding window for each SeedID
- Accept when:
  - Sequence is larger than largest sequence in window
  - Sequence is contained within window and not yet received
- Notes:
  - Constraining number of active Seeds reduces memory requirements
  - Sliding window size can be any positive value (implementation choice)

Applying Trickle

- Advertise recently received (SeedID, Seq)*
  - Announce what multicast messages a node can offer to neighbors
  - Indicate what messages have not yet been received
- Trickle timer drives (SeedID, Seq)* ICMP advs
  - Adaptive timer allows quick initial propagation, low-cost maintenance
  - Suppression allows adaptation to density
- Retransmit multicast message after noticing that a neighbor is not yet "up-to-date"
  - Only transmit data as needed (useful for large datagrams)

Trickle Parameters

- Parameters
  - Imin, Imax, k (as defined in draft-ietf-roll-trickle)
  - Tactive - time duration for retransmitting a multicast message
  - Tdwell - time duration for maintaining state about a multicast message
- When k is infinity (suppression disabled)
  - Send data message instead of advertising (SeedID, Sequence)

Recap

- Use Trickle to disseminate multicast messages
- Advertisements driven by Trickle timer
  - Reduces redundant transmissions without topology maintenance
- Flexible parameters
  - Conservative (k=1): Minimizes redundant transmissions but increases propagation time
  - Aggressive (k=infinity): Reduces to simple flood.

- Thoughts & comments?