Self-Learning Forwarding Strategy: Implementation Design in NFD

Version 2

Teng Liang
Oct 21, 2019

Overview

- Goal: make NFD work out of the box
- Features
  - Handle no next hop
  - Handle Interest retransmission
  - Handle NACK
  - Handle timeout
- Packet processing flows
  - Interest
  - Data
  - NACK

Interest Processing Flow

- Suppression check (using exponential backoff)
  - SUPPRESS
    - Drop
  - NEW
    - Find eligible nexthop with the lowest cost
      - No nexthop
        - » discovery Interest, broadcast it
        - » non-discovery Interest, NACK it
      - Nexthop found, forward Interest (marking non-discovery)
  - FORWARD
    - Find an unused eligible nexthop with the lowest cost
      - Not found, all faces have been tried
        - » Check if retx count reaches *RETX_TRIGGER_BROADCAST_COUNT*, and if true, reinitiate Interest flooding at consumer only (after clearing all FIB entries)
      - Nexthop found, forward Interest (marking non-discovery)

Data Processing Flow

- Check the out-record for the PIT entry
  - Empty, then use the parent Strategy class data handler (not sure why bug exits if removing this)
  - Exists
    - Check if outgoing Interest was discovery or non-discovery
      - Discovery
        - » The Data may carry a Prefix Announcement (PA); if so, use it to add routes
      - Non-discovery
        - » Check if a PA is needed to attach to the Data (when incoming interest was discovery and not from a local face); if true, do it. (Finding PA is async and conducted on the rib_io)
      - Send Data to all downstreams

NACK Processing Flow

- Only NO_ROUTE NACK is considered
  - Remove the next hop (only non-discovery Interest will be NACKed)
  - Try to find an unused eligible next hop with the lowest cost
    - No nexthop
      - Check if this NFD connects to consumer
        - » True, broadcast discovery Interest
        - » False, send NACK to downstreams
    - Nexthop found, send the Interest out