draft-ietf-pim-eckert-pim-mrh-experiment-00

PIM-WG, IETF122 Bangkok

Toerless Eckert, Futurewei USA (tte@cs.fau.de) (Editor)

Problem statement

- Extend SRv6 architecture with stateless multicast delivery
  - Same goal as SRv6 unicast, just with stateless replication in network
  - And, as SRv6 with or without steering across intermediate hops

- Needs to follow principles of RFC8200 unless clear need to go beyond it
  - Forwarding via IPv6 routing extension header

Target output

- „Architecture/Solution“ document in PIM for stateless multicast replication for SRv6 networks
  - Use case is IP Multicast service for IPv6 only networks also using SRv6 for unicast services
    - Better than static preprovisioned replication SID (tree state) in networks (RFC9524)
  - This -00 draft may be far away from that goal –
  - Work could be initial „experimental“ status so we move to Stds. track

- 6MAN documents to define MRH = Multicast Stateless Routing Header(s)
  - Alignd with RFC8200 rules (IPv6 source routing rules)
    - Routing header has information that allows to re-write IPv6 header destination address on ever (loose) hop
    - Simple or compressed list of IP-addresses in unicast
    - Compressed list of destinations (e.g.: bitstring) or tree nodes in multicast
  - Aligned with SRv6 architecture
    - intended to be used alongside Unicast traffic using SRv6 Unicast haders
    - Unicast now has 4 IPv6 unicast routing extension header „Routing Type“ types !!! (3 for SRv6, one for IoT)
    - (And multicast does not even have a single one ;-(
  - PIM must provide architecture to indicate demand for the 6MAN document(s)

Technical proposal outline MRH

- Routing Type
  - One for stateless destination based replication – aka without intermediate steering
  - One or more for steering based replication („represent compressed tree“ in header“)
- MRH Sub-Type
  - Format defined by spec for the „Routing Type“
    - Unicast: 3 = RPL, 4 = SRH, 5 = CRH-16, 6 = CRH-32 (Compressed Routing Header)
- Optional TLV objects
  - Same as in SRv6 Unicast (SRH and CRH headers). Complex encoding, only interpreted in receivers (not midpoint routers)
- MSER-Segment (Multicast Source Exit Router – Segment)
  - Indicates „Service“ - „IP Multicast ASM/SSM“ if IPv6 Multicast group address, or new group-communications semantic SIDs if unicast
  - Not part of TLV options because it needs to be processable by ancillary network services hop-by-hop
    - As is standard in stateful IP Multicast today

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  |  Routing Type | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|    MSER-Segment (128 bit IPv6 address, unicast or multicast)  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         MRH Sub-Type specific data            |
+                                                              //
//                        ...                                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//        Optional Type Length Value (TLV) objects (variable)  //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Non-Steering MRH Sub-Type specific data

proposal

- BSL, SD, SI, OAM, Entropy, Bitstring – as in RFC8279 (BIER architecture)
  - BIER architecture compliant – makes advanced functions such as FRR easy!
    - Allow re-use of HW BIER processing blocks in NPU (as much as possible – except for per-hop encap rewrite)
  - NOT using RFC8296 (BIER header), I-D.bier-bierin6 – those are not end-to-end RFC8200 processing compliant
    - RFC8200 requires per-hop destination IPv6 address rewrite, BIER header only allows per-hop MPLS label rewrite (MPLS optimized header)
  - MSER-Segment from MHR common header eliminates need for BIER overlay layer
- Ver, BFIR-id, TC, S, TTL, DSCP, Proto (used in RFC8296 BIER header)
  - Not needed: MPLS specific fields, or fields already present in IPv6 header (saves 64 bits, makes it SID semantic)
- Depths of bitstring in packet (for HW lookup) should not be an issue:
  - SRv6 routers thatcan inspect 8 * 128 bit SRH, then it can inspect > 512 bit bitstring at same packet depth.
  - Likely length of bitstring is limiting factor,not depth in packet

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  BSL  |       SD      |       SI      |OAM| Entropy           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                BitString  (first 32 bits)                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                BitString  (last 32 bits)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Steering MRH Sub-Type specific data

proposal

- TBD: Various options presented, some implemented for validation in P4-Tofino
  - Highs-speed validation in P4-Tofino on some
- Why Steering needed ?
  - Traffic engineering: „flat bitstring“ BIER-TE way too complex
    - was designed to fit steering into pre-existing BIER forwarding rules – NPU forwarding hardware today can do better
  - Also scale comparisons of different approaches (research papers)
  - Scales easier than flat bitstring (BIER – non-steering) in large networks
    - No management of addressing for bitstrings and co-location needed (SD, SI addressing space)
    - „In a very large network every random packet to N receivers using BIER needs N packet copies because each receiver is in a different bitstring“
    - Allows full self-configuration – SID, SD, bitstrings need central „controller“ management
- Encoding details TBD
  - Tree structure of Bitstrings and/or Tree of local next-hop SIDs – or both
- Would like to see a format not only feasible for „Vendor ASIC“ but also for researchers
  - Aka: public reference implementation for some researcher programmable high-speed forwarding platform
  - Good past results with this approach with BIER on P4-Tofino, including showing how easy it is to go from RFC8296 header to IPv6

MSER-Segment discussion

- MSER segment is necessary for MRH solution to fully support IP Multicast
- IP-Multicast (v4, v6) operates on Destination and Source address
  - Without the ability to carry Destination IPv6 address, stateless solution can not replicate IPv6 Multicast
    - In IPv6 source routing, destination IP address only carries intermediate-hop IPv6 unicast address until final delivery
    - For this final delivery, we need to encode one additional IPv6 address that can carry the desired semantic.
    - To support IPv6 multicast, this destination address needs to be an IPv6 multicast address
  - Counterpoint: MPLS multipoint or BIER: They can only act as a replicating underlay. They always need to encapsulate IP Multicast packets. No hop-by-hop knowledge of what IP Multicast data is carried (unless deep packet inspection is performed)
- Ancillary per-hop services common for IP multicast
  - IPfix - accounting/billing/troubleshooting. Was explicitly defined to decouple this monitoring state from any possible tree building state. So operators can introduce this added value/complexity independent of what multicast forwarding mechanism they choose. Statistics all based on (S,G) (source, destination addresses).
  - Group-range ACL based traffic policies and feature management
    - Bandwidth, Filtering, Policing, DSCP setting, Queue, discard-policy, VRF-mapping, ...
    - Firewall, IP Multicast NAT (with source,dest rewrite – of course NAT)
    - GDoI / IPsec based payload encryption/decryption (Multicast IPsec – SA bound/defined for group ranges)
- SRv6 architecture also expands semantic of IPv6 addresses – SID
  - MSER also supports this: if MSER-Segment is an IPv6 unicast address, then it is not IP Multicast group address semantic, but „Group-SID“ semantic. Semantics: wide open to explore

Summary / Technical Benefits

Operationally most preferrable option for SRv6 network operator

- Closest to SRv6 unicast:

IPv6/SRv6 based:

- RFC8200 compliant/derived end-to-end IPv6 source routing multicast replication
- Same forwarding architecture as SRv6 unicast, shared TLV space, SID definitions

Single network layer IPv6+MRH header for IPv6 multicast and SRv6 group SIDs

- No additional overlay encapsulation to forward IPv6 multicast
- No additional per-hop tunnel encapsulation – neither for strict nor loose trees.

Supports use of all ancillary per-hop forwarding plane services

- As possible for (stateful) IPv6 multicast

Destination based source routing as defined by RFC8279

- Well explored, HW investments (bitstring operations), Simple for smaller networks, FRR!

Steering based source routing based on TBD „compressed encoded tree“

- Primarily to support stateless traffic engineering („replace RSVP-TE/P2MP)
- But also to easier scale to largest networks with more efficient replication

Backup slides

Why is RFC8296+BIERin6 not sufficient

Operation / Architecture experience with multicast from the past:

- If operators have an urgent desire to do multicast because they did no plan for the requirement...
- They will take whatever you give them. Even if they do not like it.

- This is how we engineers often got first round of multicast deployed

- If operators can plan ahead, and still need multicast
– then it needs to match their unicast network architecture as much as possible
  - No additional control plane protocols
  - As little changes / enhancements for packets on the wire vs. Unicast as possible
  - Same operational ancillary functionality as also in unicast

- This is how we (IETF) had to redo a lot of already „great“ working IP Multicast functionality to better meth operators operational „similarity to unicast“ requirements

Example

- MPLS forwarding plane for multicast history:
  - Needed Multicast service for MPLS network infra offering L3VPN
  - First solution: IP multicast in parallel to MPLS unicast across core („draft-rosen“)
  - Worked perfectly fine in many customers. But later customers where adament to have actual MPLS forwarding plane for multicast - „we do not want new type of forwarding or control plane protocols (PIM).
  - Resulted in design of mLDP instead of PIM (or even PIM+labels which also existed since 1998)
  - Later, also overlay signaling (PIM) was replaced with BGP signaling to completely align with unicast protocol world as good as possible.

- BIER introduced stateless multicast
  - RFC8279 introduced „network layer agnostic“ architecture – which is great
  - RFC8296 introduced a „single header designed to suit all networks“ encapsulation/
    - Very much optimized for MPLS though: firrst 32 bit of BIER header are originally MPLS label word
    - BIERin6 does not change this. It only adds per-hop IPv6 tunneling of BIER packets
  - Not at all aligned with SRv6 unicast forwarding
  - Changes needed to make SRv6 network operators be happy much smaller than what we had to do to make MPLS network operatos happy.
    - Primarily MRH extension header instead of RFC8296 BIER header
    - Unicast source routing has already 4 headers. Clearly vendors can do more than just one!

Collaboration PIM and BIER

- Would be great to benefit from BIER experience in SRv6 stateless multicast work – and avoid any duplication of work not necesary to make SRv6 network operators happy
  - And enable more adoption of work invested into RFC8279 through more SRv6 networks willing to adopt the technology because of use of MRH

- Segment routing (unicast) seems like a good template
  - Common overall architecture
  - Different MPLS and IPv6/SRv6 extensions / encapsulations
  - Customers will be confused if we could take the same road for multicast, but don‘t doit

- Same applies to steering based stateless multicast
  - Basic encoding should be shared between BIER/MPLS encapsulation and SRv6/MRH encapsualation