Solution Overview
Stateless Traffic Engineering Multicast

or how to stuff a multicast tree into an IPv6 extension header
and be able to process it hop-by-hop

MSR6 BoF IETF 114 Philadelphia

v1.0 07/21/2022

draft-geng-msr6-traffic-engineering-01, draft-geng-msr6-rlb-segment-00,
draft-chen-pim-srv6-p2mp-path-06, draft-chen-pim-mrh6-03,
draft-eckert-msr6-rbs-00 and draft-cheng-spring-ipv6-msr-design-consideration

Operator
Waiqiang Cheng, Ysiong Lio (China Mobile), Aijun Wang (China Telecom), Zuangzhuang Qin (China Unicom)
Gyan Mishra, Mehmet Toy (Verizon)

Vendor
Xuesong Geng, Fenkai Li, Zhenbin Li, Rui Meng, Jingrong Xie, Xiuli Zheng Xuesong Zheng (Huawei)
Chi Fan (New H3C Technologies), Yanhe Fan (Casa Systems), Lei Liu (Fujitsu), Xufeng Liu (IBM Corporation/Volta)

Research
Toerless Eckert, Huaimo Chen (Futurewei)

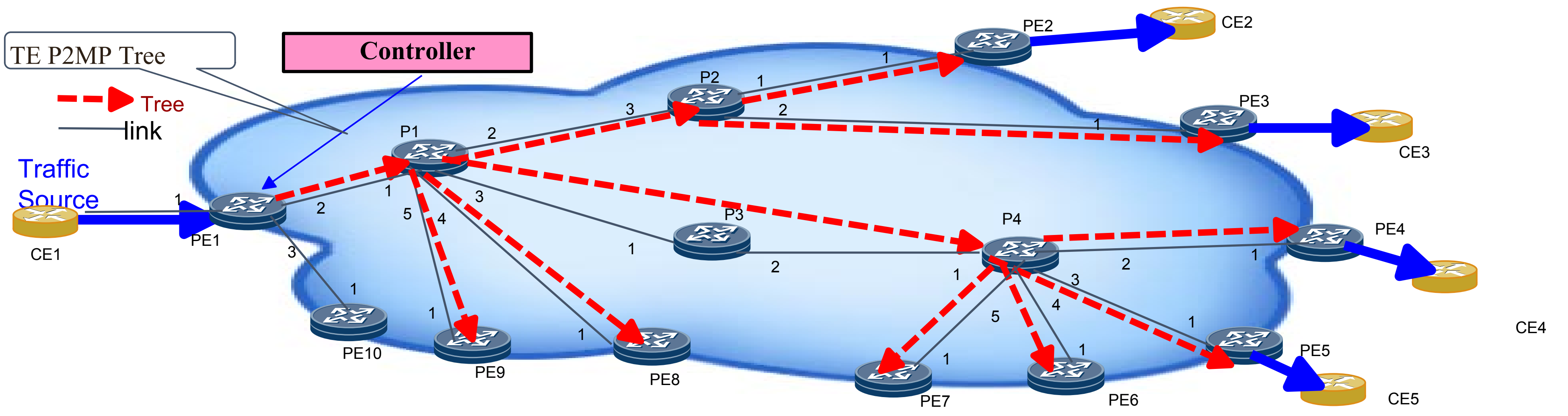
MSR6 “Traffic Engineering” Architecture Overview

Stateless native IPv6 forwarding across strict and loose hops with replication

With “Tree Engineering” (Steered Tree)

End-to-End (ingres PE to egres PE), CE-CE via usual IPv6 in IPv6 encap

Loose hops for incremental deployment and simpler, optimized trees



New IPv6 Extension Header – MRH encodes the engineered tree (here assuming routing header)

```
|<--IPv6 header -->|<--Routing header--->|
+------------------+---------------------+--------------------------+
|  Next Header =   | Next Header         | (other extension header) |
|43(Routing header)|                     | IP multicast packet/data |
| SA=IPv6 Address  |Routing Type=TBD(MRH)|                          |
| DA=IPv6 Address  |Tree/Subtree encoded |                          |
+------------------+---------------------+--------------------------+
                   |<------- MRH ------->|
```

Why encode a tree ?

- Classical unicast TE + multicast TE reasons
  - Resource guarantees, avoid unexpected re-route by IGP
    - Admission Controller: reserve bandwidth, latency, no-loss, calulate throughput, latency paths
  - Network Capacity optimization
    - Like Unicast non-ECMP multipath, but also Steiner trees
  - Path Diversity (live-live, disjoint failure domains), reduce loss
    - Combined with e.g. MoFRR, PREOF on the edge

- Better scalability than “flat bitstrings” ?! (counterintuitive !):
  - Flat-bitstrings: Packet can address only destinations (BIER) and hops (BIER-TE) that fit into single bitstring
    - Need to hard-segment destinations (BIER) and topology (BIER-TE) into different bitstrings during provisioning
  - Example: 256 bit long bitstrings, 2048 egres PE in network, and input packet for 8 of those 2048 egres PE
    - Flat-bitstrings: requires 8 packets from ingres to egres PE when all 8 egres-PE are in a different Bitstring.
    - Well compressed Tree options can always support delivery via one packet
  - Relevance: Sparse multicast (few receivers) was day 1 core IP Multicast goal (PIM-Sparse-Mode)
  - Initial large scale MSR simulation result available showing significant better than flat-bistring results!

- BIER/BIER-TE was built for assumed minimum feasible complexity - 10 years old
  - We know current/next gen hardware can do better
  - But selecting / optimizing MRH header for processing performance is one key work item !

Five MSR6 TE Solution Drafts

Too much Detail and each proposal would take more than half this time slot

- Lets focus on commonalities / aspects

But many authors invested lot of work – and hoped to present their work

- Attempt at high level comparison on next slide
- (best understood if you had seen prior side-meeting slides)

Important aspects in each draft:

- Overall MRH header encoding.
- Tree Encoding:
  - How to encode vertices (adjacencies of an MSR on the tree):
    - bitstring, SID, interface-index,...
  - How to link the vertices
  - How to (de)serialize the tree
  - Local state required on routers for MSR6 packet forwarding:
    - BIFT (e.g.: like in BIER/BIER-TE) – details of BIFT
    - or more (e.g.: Topology)
- Processing rules (explicit through pseudo-code or implied by by structure decision)

1) Stateless Traffic Engineering (TE) Multicast using MRH (draft-chen-pim-mrh6-03)

a) IPv6 extension header for TE Multicast is defined
b) TE Tree is represented by the links on the tree
c) The links are encoded by Link numbers and bitstrings
d) A link number is local to a node
e) For a portion of tree, a more efficient encoding (bitstring or link #) is used.

2) Recursive Bitstring Structure (RBS) for Multicast Source Routing over IPv6 (MSR6) (draft-eckert-msr6-rbs-00)

a) Not presented in any MRS6 side meeting yet
b) MSR6/RBS IPv6 extension header with Tree AND IP multicast destination address
c) TE Tree is represented by the adjacencies on the tree
d) The adjacencies are encoded by bit positions in bitstrings
e) A bit position is local to a node

3) Stateless SRv6 Point-to-Multipoint Path (draft-chen-pim-srv6-p2mp-path-06)

a) Multicast SIDs for the nodes on tree
b) Tree structure in SIDs' arguments by N-Branches and N-SIDs as "pointer" to start of sub-tree/branch
c) Procedure of SID duplicates packet for each branch, and sends copy to next hop

4) IPv6 Multicast Source Routing Traffic Engineering (draft-geng-msr6-traffic-engineering-01)

a) End.RL (MSR6 Endpoint Replication List) SID for each node on tree
b) Arguments in SID: "Replication number" indicating the number of replications and a "Pointer" pointing to the first child
c) Procedure of SID replicates packet for each child and sends copy to root of child

5) RLB (Replication through Local Bitstring) Segment for Multicast Source Routing over IPv6 (draft-geng-msr6-rlb-segment-00)

a) End.RLB (Replication through Local Bitstring) SID with LB
b) Local Bitstring indicating the links on tree and Pointer.
c) LB Segment is a special segment of 128-bits containing the Local Bitstring.
d) Procedure of SID replicates packet for link with bit set to 1 and sends copy to next hop

Example merged MRH Header option (and alternatives)

Single New IPv6 Routing Header for MSR

- Else every new MRH option requires separate Routing Type

MSER-Segment:

- IPv6 destination address of packet
- *Or else we need another header for native IPv6 multicast with MSR.*

MRH Sub-type:

- To support different MRH options

MRH Sub-Type specific data:

- Encoded list of egress MSR routers (BE)
- or steered tree information (TE)

Service Level Parameters (optional)

- Or else we may need another extension header for them
  - Easier if per-hop forwarding only needs to look at one header ?!
- Stateless multicast means no-per-flow state, so this may require novel DetNet queuing mechanisms.
- DSCP may not be good enough.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  |  Routing Type | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|   MSER-Segment (128 bit IPv6 address)                         |
|   (optional)                                                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
?                                                               ?
?        Service Level Parameters, e.g.: detnet                 ?
?        TBD, e.g.: latency, queuing parameters                 ?
?                                                               ?
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| MRH Sub-Type |        MRH Sub-Type specific data              |
+-+-+-+-+-+-+-+-+                                               //
//                         ...                                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                              //
//       Optional Type Length Value (TLV) objects (variable)    //
//                                                              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

More core decision/validation points

- RH (Routing Header), single, multiple, and/or DoH / HbH ?
- RFC8200 relevant header encoding/processing rules.
  - E.g.: minimum per-hop header rewrite (such as Segments-Left)
- How to best Encode/Serialize Tree
  - For fast processing and best compression of tree
- Metrics for evaluation / comparison of proposals
  - Simulation results of scale in networks ?
  - Processing Pseudocode ?
  - Amount of read/writes required into packet header ?
- How to make proposal easier comparable ?
  - Pseudocode (popular in multicast – PIM, BIER)
  - Common forwarding examples ?
  - RFC8200 language ?! (specific form of pseudocode)
  - RFC8986 language (popular with user relying on SRv6) ?

- Which design aspects would 6MAN be happy to be MSR6 responsibility ?
  - Ideally we come up with a split responsibility (e.g.: Tree encoding is MSR6).
    - Examples: (Ab)use of IPv6 addressing for IPv6 multicast done in mboned – RFC3956. Quite similar!