Solution Overview
Stateless Traffic Engineering Multicast

MSR6 BoF IETF 114 Philadelphia

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

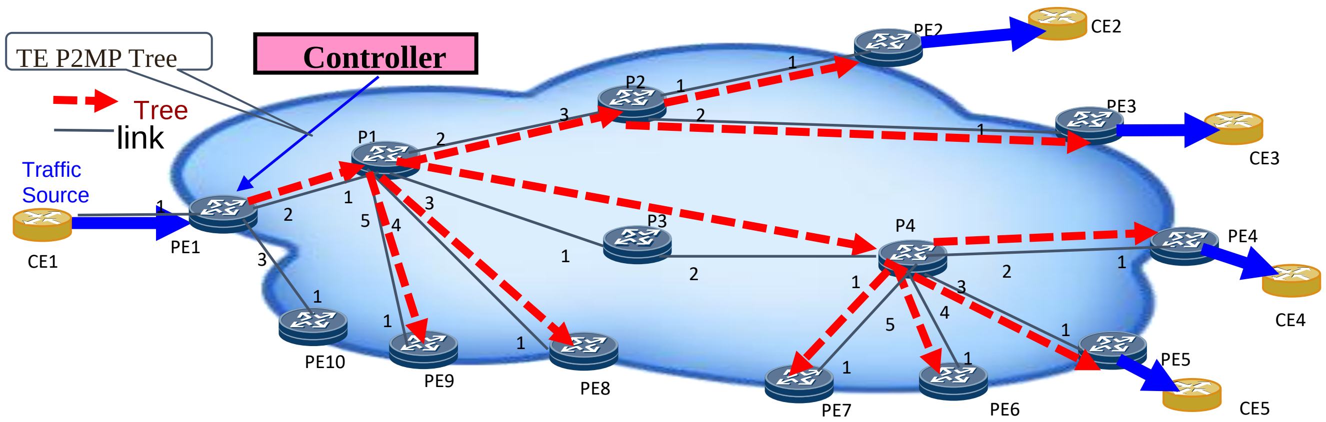
Operators
Weiqiang Cheng, Yisong Liu (China Mobile), Aijun Wang (China Telecom), Zhuangzhuang Qin (China Unicom)
Gyan Mishra, Mehmet Toy (Verizon)

Vendors
Xuesong Geng, Fengkai Li, Zhenbin Li, Rui Meng, Jingrong Xie, Xiuli 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
  - ✓ “Engineered Tree”
  - ✓ End-to-End (ingress PE to egress PE), CE-CE via usual IPv6 in IPv6 encap
  - ✓ Loose hops for incremental deployment



Encoding P2MP Tree into IPv6 Header, packet forwarded along Tree

- New IPv6 Extension Header – MRH encodes the 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 for QoS
  - ✓ Network capacity optimization
  - ✓ Path diversity (live-live) for reducing loss

Example:
- 256 bit long bitstrings, 2048 egress PE in network, and input packet for 8 of those 2048 egres PE
- Flat-bitstrings: requires 8 packets from ingress to egress PE when all 8 egress-PE are in a different Bitstring.
- Well compressed Tree options can always support delivery via one packet

- Better scalability than “flat bitstrings” ?! (counterintuitive !)
  - ✓ MSR simulation showing significant (~1 order) better!
  - ✓ Flat-bitstrings: Need to hard-segment destinations (BIER) and topology (BIER-TE) into different bitstrings during provisioning
  - ✓ Relevance: Sparse multicast was day 1 core IP multicast goal

- BIER/BIER-TE built for assumed minimum complexity - ~10 years old
  - ✓ We know current/next gen hardware can do better
  - ✓ But using MRH for performance is one key work item !

Five MSR6 TE Solution Drafts

- **Motivation of presentation**
  - ✓ Focus on common important aspects
  - ✓ Attempt at high level comparison on next slide

  (many authors invested lot of time, hoped to present their work, each proposal with detail would take half slot, best understood if you had seen prior side-meeting slides)

- **Important aspects in each draft**
  1. Overall MRH header encoding.
  2. Tree Encoding:
     - How to encode vertices/adjacencies: bitstring, SID, interface-index, ...
     - How to link the vertices
     - How to (de)serialize the tree
     - MSR6 packet forwarding table on each node
  3. Procedure (explicit through pseudo-code or implied by structure decision)

Five MSR6 TE Solution Drafts (cont.)

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.

+ Link numbers

2) Recursive Bitstring Structure (RBS) for Multicast Source Routing over IPv6 (MSR6) (draft-eckert-msr6-rbs-00) [Not presented in any MRS6 side meeting]
   a) MSR6/RBS IPv6 extension header with Tree AND IP multicast destination address
   b) TE Tree is represented by the adjacencies on the tree
   c) The adjacencies are encoded by bit positions in bitstrings
   d) A bit position is local to a node

+ Multicast Dest

Using Routing Header w/ Local bitstring

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) Args 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

Using Local bitstring

SRv6 based (SRH)

Example merged MRH Header option (and alternatives)

- **Single New IPv6 Routing Header for MSR**
  - Or else every new MRH option requires separate Routing Type
- **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 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 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
?                                                               ?
?        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 into packet header ?
- How to make proposal easier comparable ?
  - Pseudocode (popular in multicast – PIM, BIER)
  - Common forwarding examples ?
  - Unified form of pseudocode ?
- Which aspects are MSR6's responsibility (6MAN happy)?
  - Ideally split responsibility (e.g., Tree encoding is MSR6's).

    Examples: Use of IPv6 addressing for IPv6 multicast done in mboned – RFC3956. Quite similar!