RBS (Recursive BitString Structure) for Multicast Source Routing over IPv6

draft-xu-msr6-rbs-00

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

Bing Xu ((bing.xu@huawei.com)
Xuesong Geng (gengxuesong@huawei.com)

PIM WG, 03/24/2022

What it is

Native IPv6 / SRv6 solution for P2MP

Stateless (no per-hop multicast tree state) = source-routed

Intended to be part of SRv6 architecture for IPv6/SRv6 only networks

- Goal: support/Interop with all possible IPv6 functions / Extension Headers, SRv6 functions.
- Goal: No desire for separate L3 forwarding planes beside IPv6 to support multicast

No such solution today in IETF standards work.

- Stateless SRv6 is unicast only today.
- but several draft proposals exist (competing/complementary ?)

Core property of this proposal “Recursive Bitstring Structure” (RBS) addressing

How it works (high level)

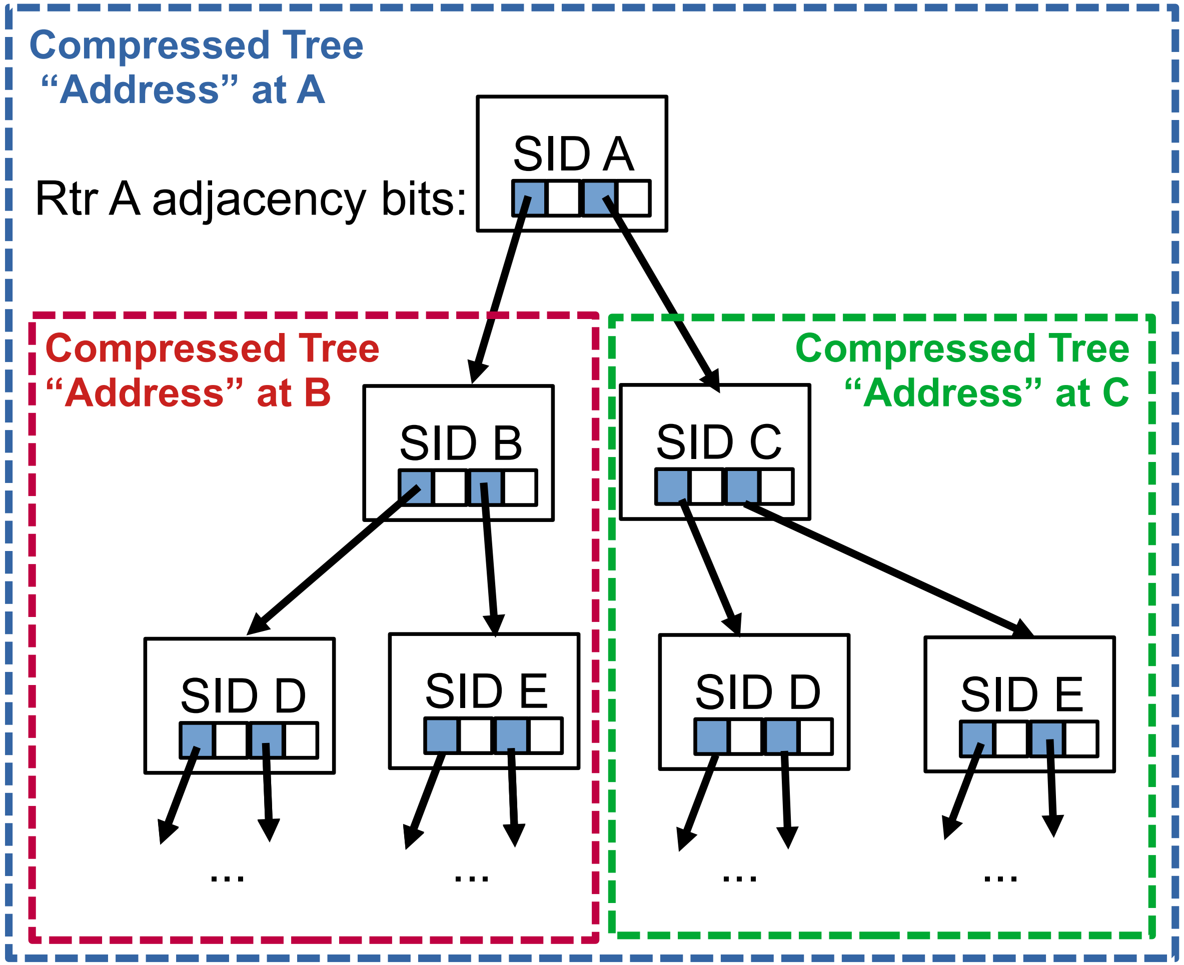
Compressed Tree "Address" at A

1) Describes whole tree!!
2) Rtr A examines its adjacency bits
3) Sees two bits are set. Creates two packet copies
4) For each copy, (active) address needs to adjust address

(active) Compressed Tree "Address" at B
(active) Compressed Tree "Address" at C

Per-hop "rewrite" operation options:

A) rewrite compressed tree to only sub-tree
Not desired by RFC8200 (but not prohibited?)
B) Adjust "Segment Offset" equivalent
C) Adjust "Segment Pointer / Length"
More compact compression than B)



Why RBS

Traffic steering

- **Enables guaranteed resources paths** (bandwidth/latency) with PCE / Admission controller
- DetNet/PREOF **path diversity** (**zero loss multicast**), **Steiner trees** (minimum total cost)
- SP network **capacity optimization** through non-ECMP load splitting (95% of what SR is being used for)
- **Predictable, simple convergence**: no micro-loops (as in hop-by-hop FIB reconvergence)

RBS vs. "flat bitstrings" and scalability/performance

- *PIM "Sparse Mode": "sparse set of receivers from large set of (possible) receivers" (core PIM goal)*

- *Simple thought example*:
  - 2560 destinations (PE), max 256 flat bitstring length = 10 bitstrings required.
  - Multicast packet to 10 destinations
  - Flat bitstring: may need to send 1…10 packets depending which bitstring the 10 destinations are in
  - RBS: always only need to send one packet *(in validated topologies)*.

- Aka: even though RBS needs to encode steering hops, it could be even more efficient in large networks

Scaling issues with flat bitstrings significantly increases when traffic steering is needed!

RBS – Encode and Process (high level)

Components length are in bits (most compact). Padding at the end

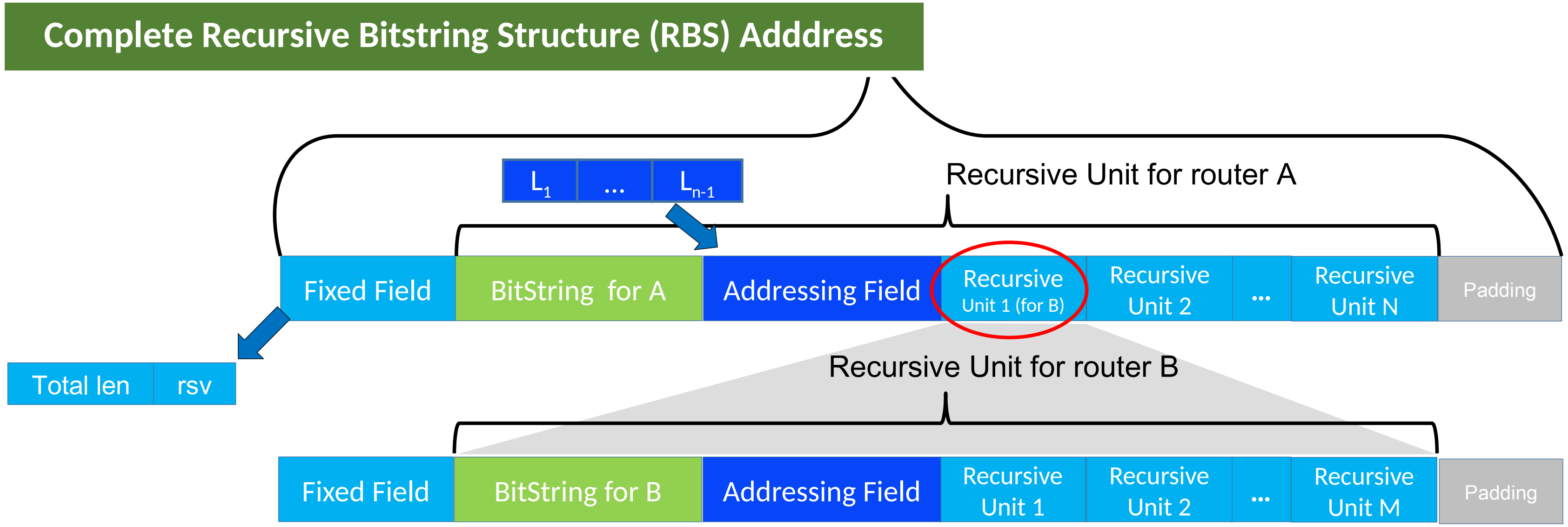
If router A has N neighbors, its bitstring is N bits long.

Bitstring replication logic same as for flat bitstring traffic steering

- bit-i set => send copy to neighbor(i) (generalized: adjacency(i)).

Additional forwarding plane work beyond flat bitstring:

- For each copy to an adjacency: select the recursive unit and "activate" it
  - A) Extract and adjust "pointers" (s/Recursive Unit A/Recursive Unit B/)
  - B) (not shown) Adjust (segment) "pointers" in fixed field to point to new Recursive Unit



RBS FIB for router A (assuming N = 4)

Index	Recursive	Segment IPv6addr
1	1	B
2	...	...
3	1	C
4	...	...

RBS address for B is of course much shorter than the one for A with extract/replace

IPv6 Multicast Routing Header for RBS

compare with SRH (RFC8754, unicast)

Architecturally/operationally the same, except for replication" ?!	
SRH: RH carries segment path	RBS: RH carries segment tree
One uncompressed solution 8 years later: work on one compressed option	RBS: One proposed compressed option Design extensibility (for alternatives ?)
Segment-by-Segment forwarding by swapping IPv6-Dest from next-segment	
Only one next-segment (unicast)	One or more next-segments (replication)
Same set of egress TLV functionality (as applicable) (TBD)	
Per transit-hop programmability (only when not compressed ?!)	Not considered (yet) no identified use-cases

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Last Entry   |     Flags     |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Segment List[0] (128-bit IPv6 address)             |
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//            ....                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//         Optional Type Length Value objects (variable)       //
//         (Segment Routing Header TLVs)                       //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MRH Sub-type? |  Addrlen      | Reserved                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      RBS Address(variable)                   //
//                                                             //
//                                                             //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//         Optional Type Length Value objects (variable)       //
//         (Segment Routing Header TLVs)                       //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next steps / IETF process

Assuming there is an SRv6 (SPRING/PIM) community that wants this (and vote to adopt)

How to split the work between “use-case WGs” (SPRING/PIM) and 6MAN

What to do so this will be easier / faster / Better 2nd time around than with SRH ?

- Would we need an additional “arch” document in PIM ?
  - Similar to RFC 8986/SRv6-arch ?
- Assume the EH spec would go to 6MAN
  - Can we bake extensibility / modification by “use-case” WG
    better/easier into EH specification than with RFC8754 ?
  - E.g.: Maybe we want multiple multicast-tree address compression option ?
  - E.g.: Define permissible per-segment modifications
    (shorten, rewrite one/two fields , rewrite more ?)

The End

Please chime in on PIM-WG for the solution overall