

Adaptive Stateless TE Multicast

draft-chen-pim-adaptive-te-00

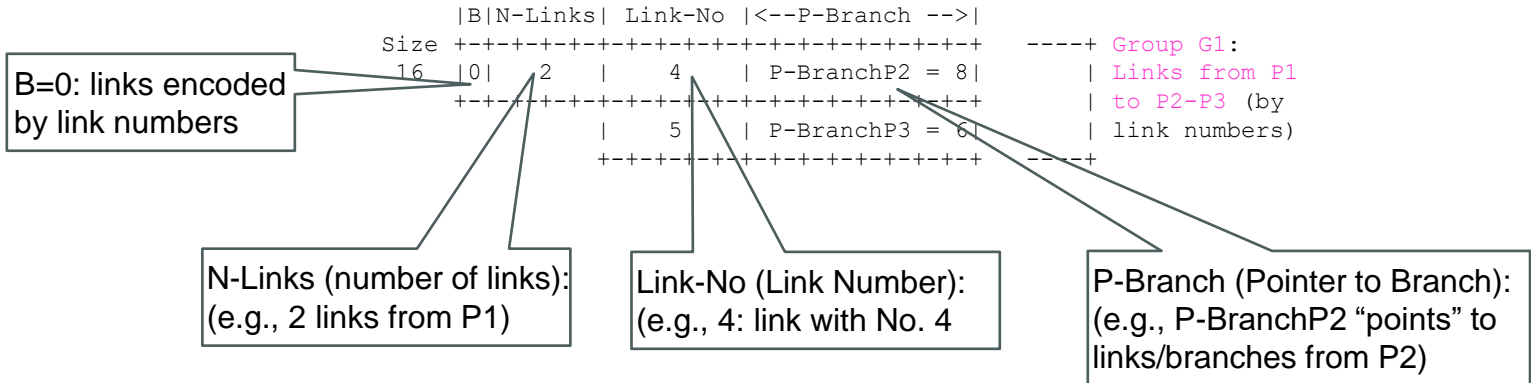
Huaimo Chen, Mike McBride (Futurewei)
Yanhe Fan (Casa Systems)
Robin Li, Xuesong Geng (Huawei)
Mehmet Toy, Gyan Mishra (Verizon)
Yisong Liu (China Mobile)
Aijun Wang (China Telecom)
Lei Liu (Fujitsu)
Xufeng Liu (IBM Corporation)

IETF 117

A Few Basic Encoding Methods (1/2)

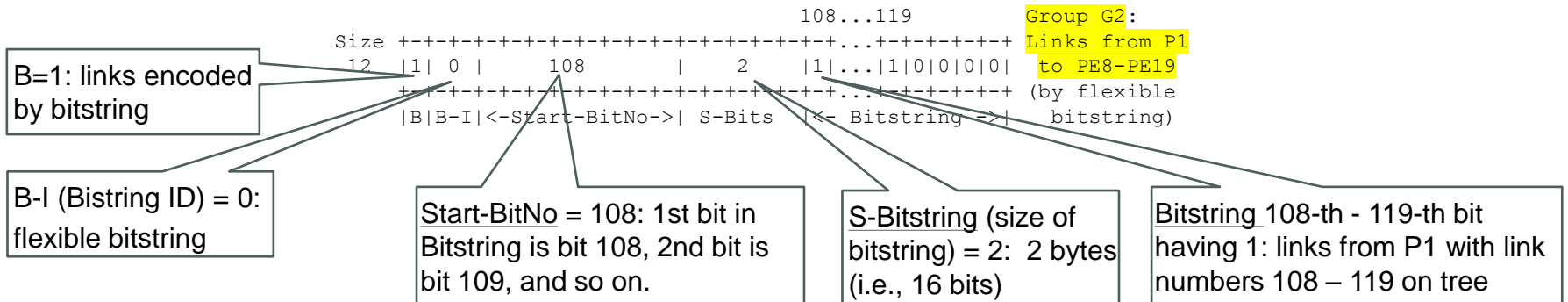
Link Numbers

Encoding 2 links from P1 to P2 – P3 by link numbers



Flexible Bitstring (Both start and end flexible)

Encoding links from P1 to PE8 – PE19 by flexible bitstring:



A Few Basic Encoding Methods (2/2)

Half Flexible Bitstring (Start fixed at 1st bit, end flexible)

Encoding 2 links from P2 by half flexible bitstring.

B=1: links encoded by bitstring

B-I (Bistring ID) = 1:
Half flexible bitstring

```

Size +-----+-----+-----+-----+-----+-----+-----+-----+
      8 |1| 1 |      1      |0|0|0|1|0|1|0|0| Links from P2
      +-----+-----+-----+-----+-----+-----+
      |B|B-I|S-Bitstring|<- Bitstring ->| bitstring)
  
```

S-Bitstring (size of bitstring) = 1: 1 byte (i.e., 8 bits)

Bitstring's 4-th, 6-th bit having 1: links from P2 with link numbers 4 and 6 on tree

Fix Bitstring (Start fixed at 1st bit, end fixed at last bit), but number of links on a node is not fixed

Encoding links from P1 by fix bitstring.

B=1: links encoded by bitstring

B-I (Bistring ID) = 3:
fixed bitstring

```

      1 2 3 4 5 6 ... 108 ... 119
+-----+-----+-----+-----+-----+-----+-----+-----+
|1|1|1|1|0|0|0|1|1|0| ... |0|1| ... |1|0| P-BranchP2 | P-BranchP3 | P1 to P2,P3,PE8-PE19
+-----+-----+-----+-----+-----+-----+-----+-----+
|B|B-I|<--- Bitstring (120 bits) ---->|<- P-Branch ->|<- P-Branch ->|
  
```

Bitstring's 4-th bit = 1: link from P1 with link number 4 on tree

Bitstring's 119-th bit = 1: link with link number 119 on tree

Neighbor table

Table 1 is Neighbor table of P1.

16 rows of link number, link type and IPv6 address of next hop.

Row 1 has link number 3 and link type SB (Split Branch) for split branch link of P1.

Row 2 has link number 4, link type P2P for link from P1 to next hop P2 and IPv6 address of P2.

Row 4 has link number 108, link type P2P2E (P2P to Egress) for link from P1 to next hop egress PE8 and PE8's IPv6 address.

...

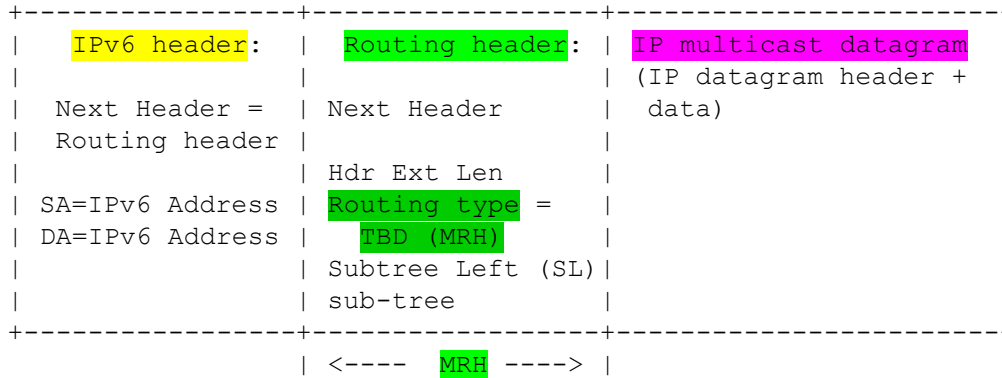
Using link number 4, P1 gets P2's IPv6 address from table;

Using link number 3, P1 knows it is a split branch (SB) link.

Link Number	Link Type	IPv6 Address of next hop
3	SB	NULL
4	P2P	IPv6 address of P2
5	P2P	IPv6 address of P3
108	P2P2E	IPv6 address of PE8
:	:	:
119	P2P2E	IPv6 address of PE19
120	P2P	IPv6 address of PE1

Table 1. Neighbor table of P1

Multicast Routing Header (MRH)



IPv6 packet with Multicast Routing Header (MRH):

- **IPv6 header** with DA and SA of IPv6,
- **routing header** with Routing type (TBD): MRH and
- **IP multicast datagram**.

Figure 5A. Routing Header (MRH) in IPv6 packet

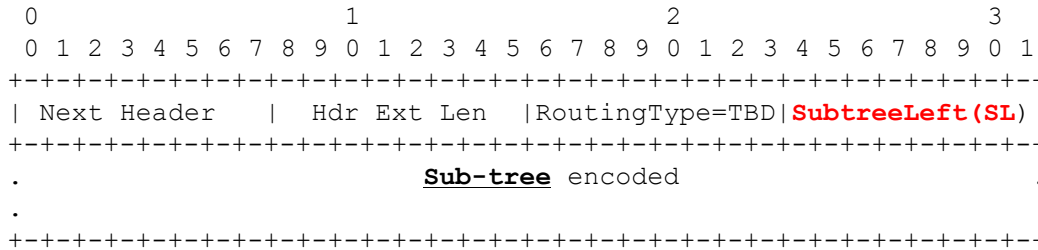


Figure 5B. Multicast Routing Header (MRH)

MRH:

Next Header: Header after MRH

Hdr Ext Len: Length of MRH

Routing Type: TBD means MRH for multicast

Sub-tree Left (SL): points to sub-tree

Sub-tree: encodes sub-tree/tree

Procedure/Behavior on Ingress

Ingress

duplicates packet for each link/sub-tree,
encapsulates copy in MRH containing sub-tree and
sends copy to next hop of link.

E.g., one sub-tree/link from PE1, packet sent to P1 is shown.

```

| IPv6 Header | <----- MRH -----> |
+-----+
| DA=P1's IPv6 | Routing Type = TBD, SL = 20 | IP multicast |
| SA=PE1's IPv6 | sub-tree from P1 to PE2-PE19 | datagram |
+-----+
Size                               Links
+-----+
20 | 0 | 2 | 3 | P-BranchG1=16 | 2 SB links split
   +-----+ | links from P1
       | 3 | P-BranchG2=12 | into 2 groups:
       +-----+ | G1 and G2
|B|N-Links| Link-No |<--P-Branch -->|
+-----+
16 | 0 | 2 | 4 | P-BranchP2 = 8 | Group G1:
   +-----+ | Links from P1
       | 5 | P-BranchP3 = 6 | | to P2-P3 (by
       +-----+ | link numbers)
                               108...119
                               +-----+
                               |B|N-Links| Link-No |<--P-Branch -->|
                               +-----+
                               12 | 1 | 0 | 108 | 2 | 1 | ... | 1 | 0 | 0 | 0 | 0 | Group G2:
                               +-----+ | Links from P1
                               |B|B-I|<-Start-BitNo->| S-Bits |<- Bitstring ->| | to PE8-PE19
                               +-----+ | (by flexible
                               |B|N-Links| Link-No |<--P-Branch -->| | bitstring)
                               +-----+ | Links from P2
                               8 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | | to PE2-PE3
                               +-----+ | (by half flex
                               |B|B-I| S-Bits |<- Bitstring ->| | bitstring)
                               |B|N-Links| Link-No |<--P-Branch -->|
                               +-----+ | link from P3 to P4
                               6 | 0 | 1 | 5 | P-BranchP4 = 3 | | (by link number)
                               +-----+ | Links from P4
                               3 | 1 | 0 | 54 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | to PE4-PE7
                               +-----+ | (by flexible
                               |B|B-I|<-Start-BitNo->| S-Bits |<- Bitstring ->| | bitstring)

```

Figure 6: IPv6 packet with MRH received by P1

Procedure /Behavior on Transit, Egress Node

Transit node :

duplicates packet with MRH for each link of node,
sets SL accordingly and sends copy to next hop of link.

SB link: sets SL to P-Branch for link and processes links pointed by SL.

Link to transit: sets SL to P-Branch for link; Link to egress: sets SL to 0;

gets IPv6 of next hop, sets DA to IPv6 and sends packet to DA (i.e., next hop)

E.g., P1 duplicates packet for link to P2, set SL to P-BranchP2 = 8, and sends packet to P2.

```

| IPv6 Header | <----- MRH -----> |
+-----+
| DA=P2's IPv6 | Routing Type = TBD, SL = 8 | IP multicast |
| SA=PE1's IPv6 | sub-tree from P2 to PE2-PE3 | datagram |
+-----+
Size                               Links
8 | 1 | 1 | 1 | 0|0|0|1|0|1|0|0| | Links from P2
  | 1 | 1 | 1 | 0|0|0|1|0|1|0|0| | to PE2-PE3
  +-----+ (by half flex
  |B|B-I| S-Bits |<- Bitstring ->| bitstring)
  |B|N-Links| Link-No |<--P-Branch -->|
  +-----+ link from P3 to P4
6 | 0 | 1 | 5 | P-BranchP4 = 3| (by link number)
  +-----+ Links from P4
3 | 1 | 0 | 54 | 1 | 1|1|1|1|0|0|0|0| | to PE4-PE7
  +-----+ (by flexible
  |B|B-I|<-Start-BitNo->| S-Bits |<- Bitstring ->| bitstring)
  
```

Figure 6: IPv6 packet with MRH received by P2

Egress :

decapsulates packet (MRH with SL = 0) and
sends the IP multicast datagram to the IP multicast forwarding module.

Next

Comments