Multicast using Multicast Routing Header

draft-chen-pim-mrh6

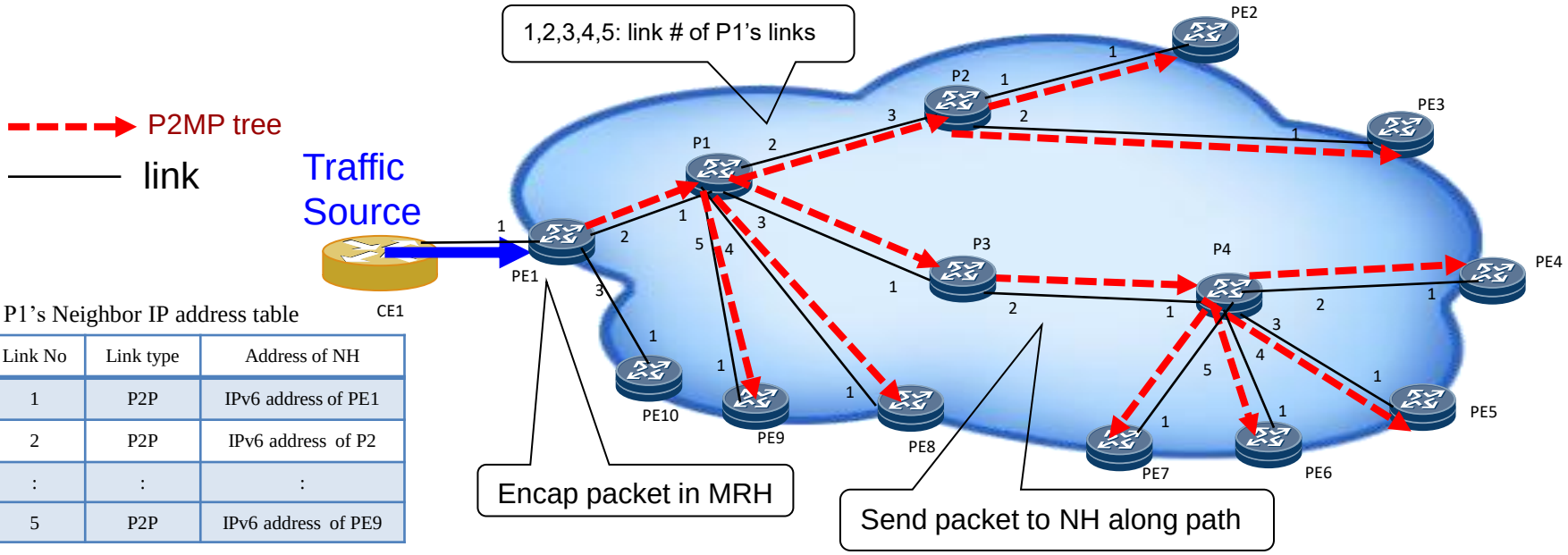
Huaimo Chen, Mike McBride (Futurewei)
Yanhe Fan (Casa Systems)
Robin Li, Xuesong Geng(Huawei)
Mehmet Toy, Gyan Mishra (Verizon)
Yisong Liu (China Mobile
Aijun Wang (China Telecom)
Lei Liu (Fujitsu)
Xufeng Liu (Volta Networks)

IETF 113

Introduction

- Existing solutions
  - Ietf-sr-p2mp-policy
  - chen-pim-srv6-p2mp-path (comments received from WG)
- But have weaknesses
- This MRH: a good alternative
  - Taking those comments into account
  - More scalable

Brief Description



P1's Neighbor IP address table

Link No	Link type	Address of NH
1	P2P	IPv6 address of PE1
2	P2P	IPv6 address of P2
:	:	:
5	P2P	IPv6 address of PE9

- P2MP path/tree is represented by the link numbers
- ➢ Ingress (e.g., PE1) encapsulates the packet in a MRH with sub-tree from NH
- The packet is transmitted along the tree to the egresses.
- ➢ After receiving the packet, a transit node (e.g., P1)
  - ✓gets/pops each of its link numbers from MRH,
  - ✓finds the NH address from a neighbor table using the link number,
  - ✓sends the packet to the next hop (such as P3)
- ➢ Egress (e.g., PE2) decapsulates the packet in a MRH and sends it to multicast forwarding module

Encoding of P2MP Path/Tree: basic idea

Link U→D on tree is encoded by 3 fields:

a). Link-No, b). N-Branches, c). S-Branches+

a). link # on U, b). # of branches from D, c). "pointer" to sub-tree from D

E.g., for link PE1→P1,

a). Link-No = 2, b). N-Branches = 4, c). S-Branches+ = 22

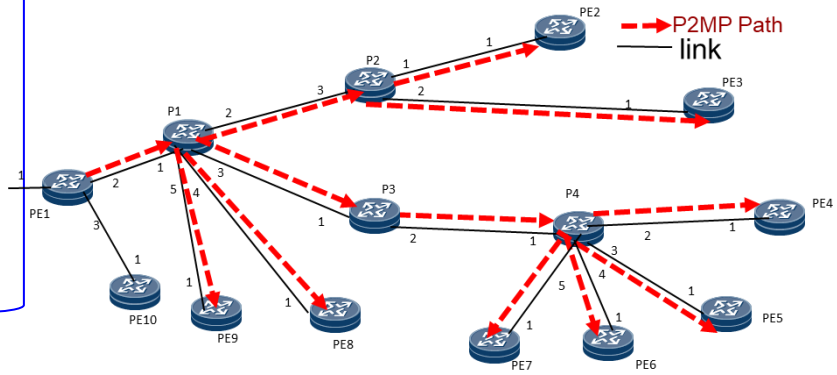
a). link # on PE1 is 2, b). 4 branches from P1, c). 22 → sub-tree from P1 ("starting from first link P1→x: P!→P2)

size	Link-No	N-Branches	S-Branches+	link	
24	2	4	22	PE1 to P1	
22	2	2	14	P1 to P2	sub-tree from P1 to PE2-PE9
20	3	1	10	P1 to P3	
18	4	0	0	P1 to PE8	
16	5	0	0	P1 to PE9	
14	1	0	0	P2 to PE2	sub-tree from P2
12	2	0	0	P2 to PE3	
10	2	4	8	P3 to P4	sub-tree from P3
8	2	0	0	P4 to PE4	
6	3	0	0	P4 to PE5	
4	4	0	0	P4 to PE6	
2	5	0	0	P4 to PE7	

For link P1→P2,

a). Link-No = 2, b). N-Branches = 2, c). S-Branches+ = 14

a). link # on P1 is 2, b). 2 branches from P2, c). 14 "points" to sub-tree from P2, starting from first link P2→x: P2→PE2



Encoding of tree from PE1 via P1 to PE2, PE3, ..., PE9

Encoding of P2MP Path/Tree: L flag

L = 1 for link U→D, D is leaf. "N-Branches" and "S-Branches+" are removed.
U→D: L + Link-No (1 byte)
L = 0 for link U→D, D is not leaf.
U→D: L, Link-No, N-Branches and S-Branches+ (2 bytes)

E.g., for link PE1→P1,
Link-No = 2, N-Branches = 4, S-Branches+ = 14
link # on PE1 is 2, 4 branches from P1, 14 "points" to sub-tree from P1

size	L	Link-No	N-Branches	S-Branches+	link
16	0	2	4	14	PE1 to P1
14	0	2	2	8	P1 to P2
12	0	3	1	6	P1 to P3
10	1	4			P1 to PE8
9	1	5			P1 to PE9
8	1	1			P2 to PE2
7	1	2			P2 to PE3
6	0	2	4	4	P3 to P4
4	1	2			P4 to PE4
3	1	3			P4 to PE5
2	1	4			P4 to PE6
1	1	5			P4 to PE7

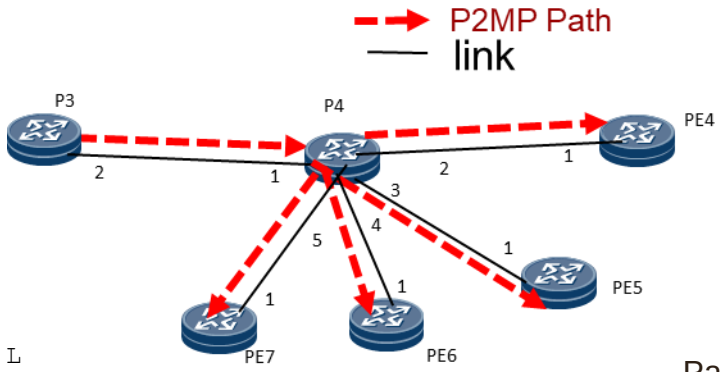
sub-tree from P1

sub-tree from P2

sub-tree from P3

For link P1→PE8,
L = 1, Link-No = 4 (link # on P1)

Encoding tree without L uses 24 bytes.
Encoding tree with L uses 16.
8 bytes are saved/reduced.



Encoding of tree from PE1 via P1 to PE2, PE3, ..., PE9 with L

Encoding of P2MP Path/Tree: B flag (P=1)

B=1 for U→D (e.g., P3→P4): Bits+ used for links from D

- i-th bit = 1 in Bits: link # i from D; Link-No removed
- # of bits with 1: # of branches from D (e.g., P4);
- P = 1: NHs are leaves (P: Plus leaves)

E.g., encoding sub-tree from P3 via P4 to PE4 - PE7 with B=1

B=1 for P3→P4,
Link-No = 2, S-Branches+ = 2
link # on P3 is 2, 2 "points" to sub-tree from P4

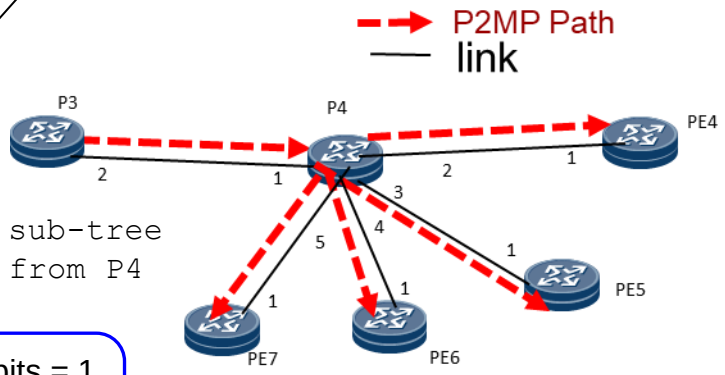
B=1: Bits+
for P4's links

```
      +-+-+-------+----------+----------+
size  |L|B|Link-No|N-Branches|S-Branches+|   link
      +-+-+-------+----------+----------+
  4   |0|1|   2   |          |    2     | P3 to P4
      +-+-+-------+-----+---+----------+
  2   |1|        1       |0 1 1 1 0 0 0| P4 to PE4   sub-tree
      +-+----------------+-------------+     - PE7   from P4
      |P|     S-Bits     |     Bits    |
```

P=1: NHs
are leaves

S-Bits=1:
Bits of 1 byte

4 links to leaves from P4 by 4 bits = 1
2-th bit = 1: link # 2 (P4→PE4);
3-th bit = 1: link # 3 (P4→PE5); …

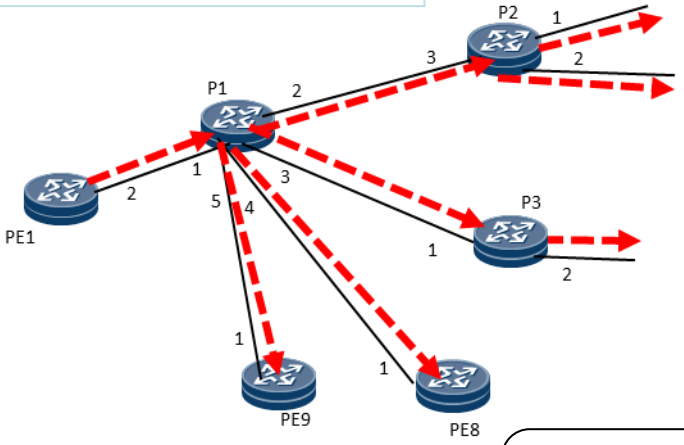


sub-tree from P3
Encoded in 4 bytes.

Encoding of sub-tree from P3 via P4 to PE4, PE5, PE6, PE7 with B=1

Encoding of P2MP Path/Tree: B flag (P=0)

E.g., Branch from PE1 via P1 to P2, P3, PE8 and PE9 with B = 1



B=1: Bits+ for P1's links

```
      +=+=+=======+==========+===========+
size  |L|B|Link-No|N-Branches|S-Branches+|   link
      +=+=+=======+==========+===========+
 13   |0|1|   2   |          |    11     | PE1 to P1
      +-+-+-------+----------+-----------+
 11   |0|        1          |0 1 1 1 1 0 0 0| P1 to P2,
      +-+-------------------+---------------+ P3,PE8,PE9
      |P|     S-Bits        |     Bits      |
```

B = 1 for PE1→P1,
Link-No = 2, S-Branches+ = 11
link # on PE1 is 2, 11 points to sub-tree from P1

4 links from P1 by 4 bits = 1:
2-th bit = 1: link # 2 (P1→P2);
3-th bit = 1: link # 3 (P1→P3);
…

P=0: link from P1 **w/o Link-No**

```
      |L|B|N-Branches|S-Branches+|
      +=+=+==========+===========+
      |0|0|     2    |     6     | P1 to P2   links
      +-+-+----------+-----------+            from P1
      |0|0|     1    |     4     | P1 to P3
      +-+-+----------+-----------+
      |1|                        | P1 to PE8
      +-+
      |1|                        | P1 to PE9
      +-+
```

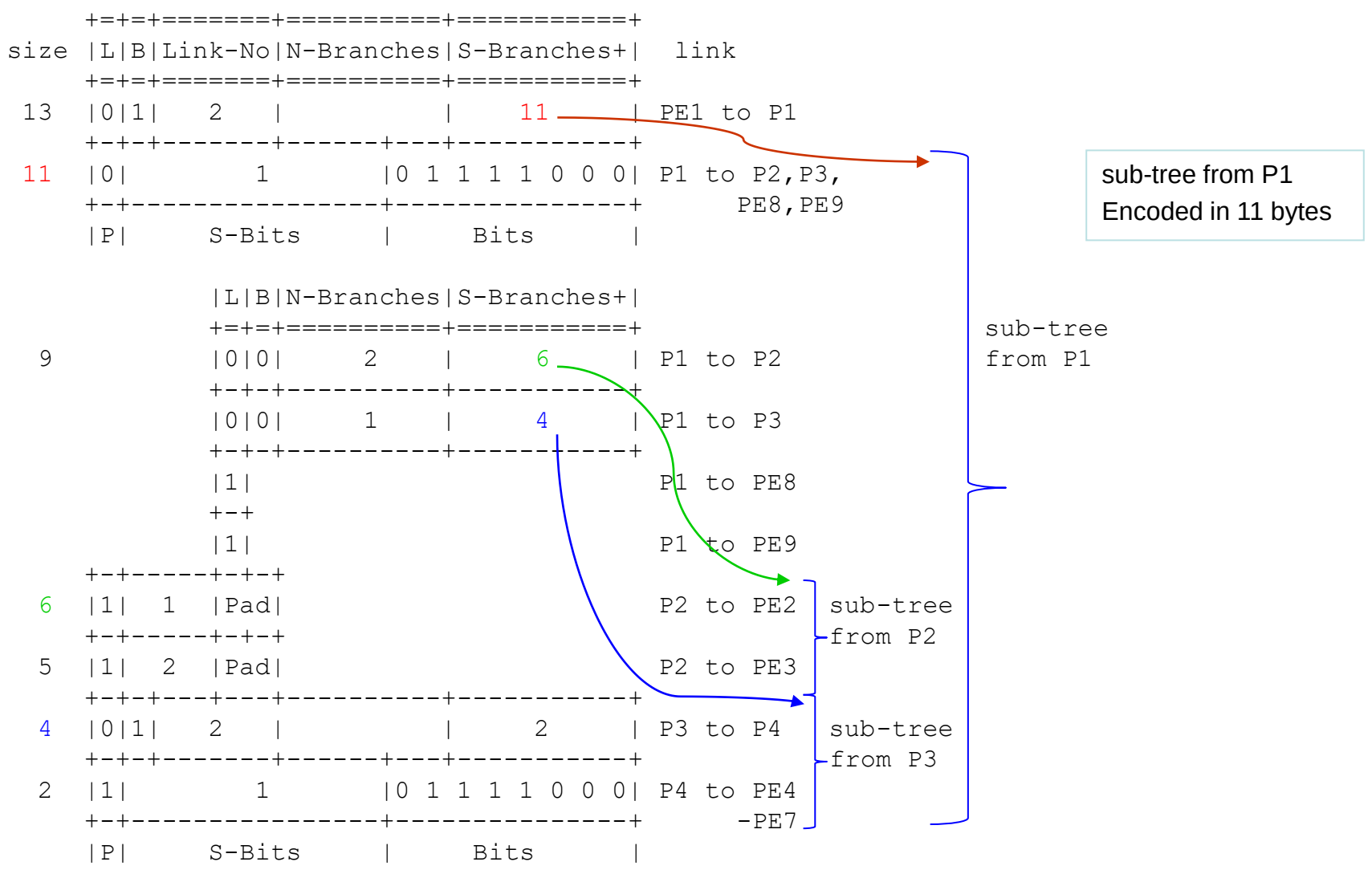
Reduced fields for P1→P2 (link # 2)
L=0, B=0, N-Branches=2, S-Branches+=6

Reduced field for P1→PE8 (link # 4)
L=1 (PE8 is leaf)

Branch part from PE1 via P1 to P2, P3, PE8, PE9 with B = 1

Encoding of P2MP Path/Tree: L and B

E.g., Encoding tree from PE1 via P1 to PE2 - PE9
Link U→D: Link-No on U, N-Branches from D, pointer to sub-tree from D; L and B for improvements



sub-tree from P1
Encoded in 11 bytes

Encoding of tree from PE1 via P1 to PE2 - PE9 with L and B

Multicast Routing Header (MRH): Format, Ingress

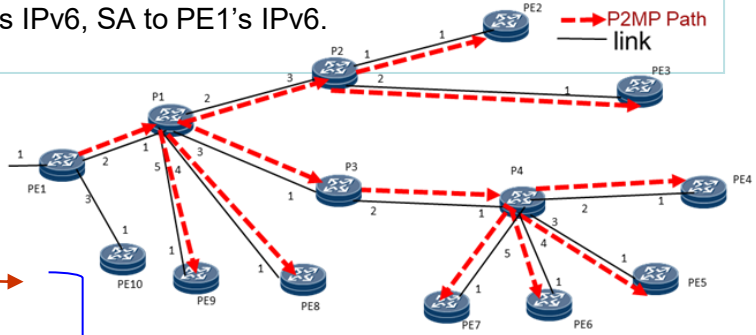
```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   | Hdr Ext Len |RoutingType=TBD|     SL        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|b| Rsv |   nB  |                                               |
+-+-+-+-+-+-+-+-+  Sub-tree from NH encoded by link numbers :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SL: points to sub-tree from NH

nB: # branches/links from NH
b : bits used for links from NH

Ingress (e.g., PE1) encaps packet in MRH for each NH and sends it to NH.

MRH includes sub-tree from NH (e.g., P1); SL, nB, b in MRH are set to values for link to NH (i.e., b=1, nB, SL= 11); PE1 finds P1's IPv6 by link # 2 for PE1 → P1, sets DA to P1's IPv6, SA to PE1's IPv6.



size	L	B	Link-No	N-Branches	S-Branches		link
13	0	1	**2**		11		PE1 to P1
11	0		1	0 1 1 1 1 0 0 0			P1 to P2,P3, PE8,PE9
	P		S-Bits	Bits			
	L	B	N-Branches	S-Branches			
9	0	0	2	6			P1 to P2
	0	0	1	4			P1 to P3
	1						P1 to PE8
	1						P1 to PE9
6	1	1	Pad				P2 to PE2

sub-tree from P1

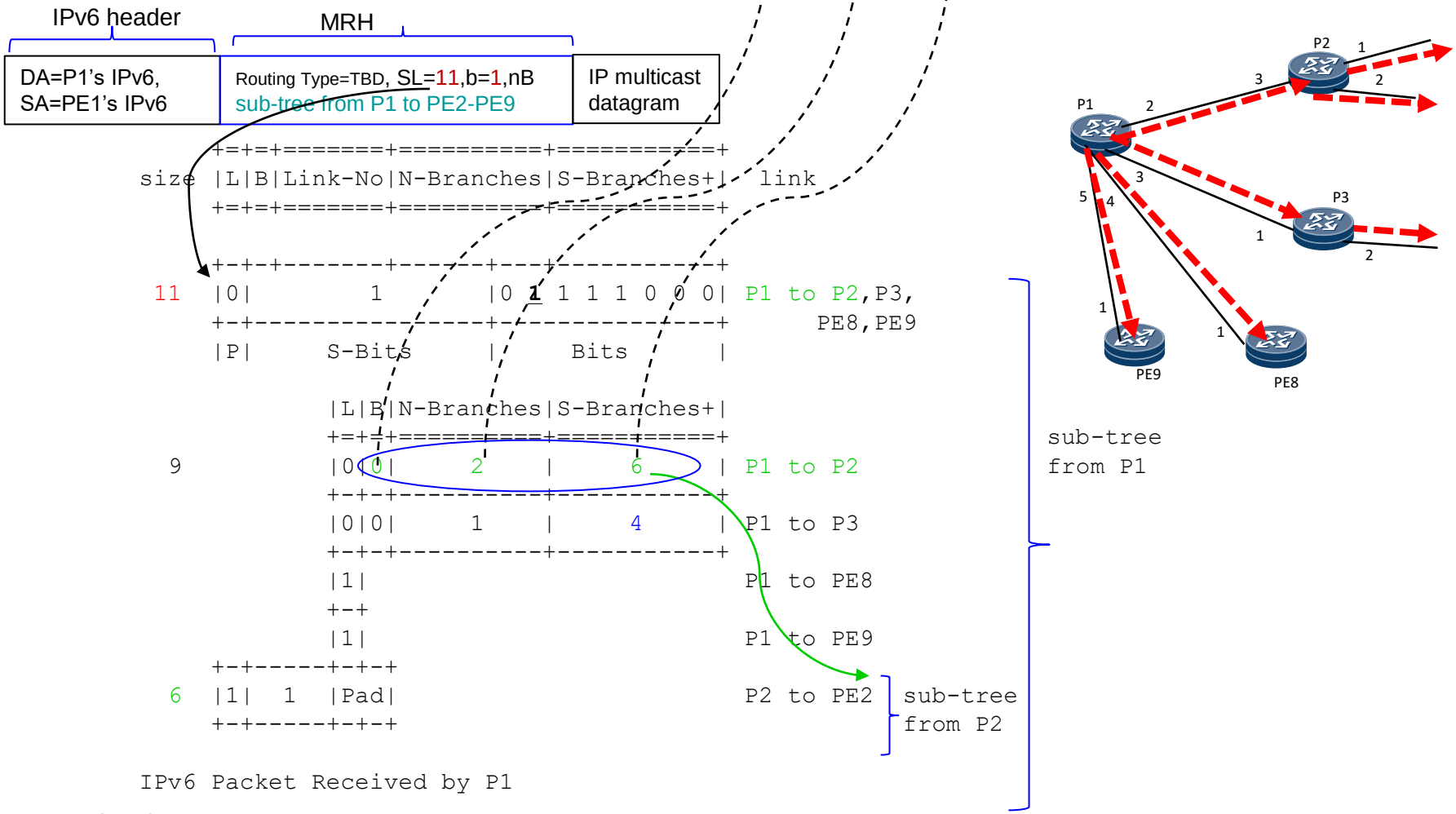
sub-tree from P2

Multicast Routing Header (MRH): Transit, Egress

Packet received by P1: 4 branches/links from P1→P2, P3, PE8, PE9 P1 sends a copy to each NH

P1→P2: b, nB, SL in MRH are set to values for P1→P2 (i.e., b=0, nB = 2, SL = 6), DA to P2's IPv6, SA to P1's IPv6;

P1→PE8 (Egress): SL = 0, DA to PE8's IPv6, SA to P1's IPv6.



IPv6 header | MRH | Packet to PE8

DA=PE8's IPv6, SA=P1's IPv6 | Routing Type=TBD, SL=0, b, nB sub-tree/leaf PE8 | IP multicast datagram

Egress PE8: (SL == 0):

Decaps, sends it to IP multicast forwarding

Next

Comments

Request for Adoption