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
**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

<table>
<thead>
<tr>
<th>size</th>
<th>Link-No</th>
<th>N-Branches</th>
<th>S-Branches+</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>2</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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")

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 \( U \rightarrow D \), \( D \) is leaf. “N-Branches” and “S-Branches+” are removed.

\( U \rightarrow D: L + \text{Link-No} \) (1 byte)

\( L = 0 \) for link \( U \rightarrow D \), \( D \) is not leaf.

\( U \rightarrow D: L, \text{Link-No}, \text{N-Branches} \) and \( \text{S-Branches}+ \) (2 bytes)

E.g., for link \( \text{PE1} \rightarrow \text{P1} \),

\( \text{Link-No} = 2, \text{N-Branches} = 4, \text{S-Branches}+ = 14 \)

link # on \( \text{PE1} \) is 2, 4 branches from \( \text{P1} \), 14 “points” to sub-tree from \( \text{P1} \)

For link \( \text{P1} \rightarrow \text{PE8} \),

\( L = 1, \text{Link-No} = 4 \) (link # on \( \text{P1} \))

Encoding tree without \( L \) uses 24 bytes.

Encoding tree with \( L \) uses 16.

8 bytes are saved/reduced.
Encoding of P2MP Path/Tree: \( B \) flag (\( P=1 \))

\( B=1 \) for \( U \rightarrow D \) (e.g., \( P3 \rightarrow P4 \)): 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 \( B=1 \)

<table>
<thead>
<tr>
<th>( L )</th>
<th>( B )</th>
<th>( Link-No )</th>
<th>( N) ( -) ( Branches )</th>
<th>( S) ( -) ( Branches+ )</th>
<th>( link )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td>( P3 ) to ( P4 )</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>( P4 ) to ( PE4 )</td>
</tr>
</tbody>
</table>

\( B=1 \) for \( P3 \rightarrow P4 \), Link-No = 2, \( S\)-Branches+ = 2
link \# on \( P3 \) is 2, 2 “points” to sub-tree from \( P4 \)

\( P=1 \): 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 \( # \ 2 \) (\( P4 \rightarrow PE4 \));
3-th bit = 1: link \( # \ 3 \) (\( P4 \rightarrow PE5 \)); …

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 \( B = 1 \)

- \( B = 1 \) 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

- 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: \textbf{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

<table>
<thead>
<tr>
<th>size</th>
<th>L</th>
<th>B</th>
<th>Link-No</th>
<th>N-Branches</th>
<th>S-Branches+</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td>PE1 to P1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td></td>
<td>1</td>
<td>0 1 1 1 1 0 0 0</td>
<td></td>
<td>P1 to P2, P3, PE8, PE9</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td></td>
<td>S-Bits</td>
<td></td>
<td>Bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>B</td>
<td>N-Branches</td>
<td>S-Branches+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td>P1 to P2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>Pad</td>
<td></td>
<td></td>
<td>P2 to PE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>Pad</td>
<td></td>
<td></td>
<td>P2 to PE3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td>P3 to P4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td>1</td>
<td>0 1 1 1 1 0 0 0</td>
<td></td>
<td>P4 to PE4</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td></td>
<td>S-Bits</td>
<td></td>
<td>Bits</td>
<td></td>
</tr>
</tbody>
</table>

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

sub-tree from P1
Encoded in 11 bytes

sub-tree from P1

sub-tree from P2

sub-tree from P3

sub-tree from P2

sub-tree from P3

PE7
Multicast Routing Header (MRH): Format, Ingress

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=11); PE1 finds P1’s IPv6 by link #2 for PE1 → P1, sets DA to P1’s IPv6, SA to PE1’s IPv6.

SL: points to sub-tree from NH
nB: # branches/links from NH
b: bits used for links from NH

---

SL: points to sub-tree from NH
nB: # branches/links from NH
b: bits used for links from NH

---

I/Ogress (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=11); PE1 finds P1’s IPv6 by link #2 for PE1 → P1, sets DA to P1’s IPv6, SA to PE1’s IPv6.
Multicast Routing Header (MRH): Transit, Egress

Packet received by P1: 4 branches/links from P1→P2, P3, PE8, PE9
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
DA=P1’s IPv6,
SA=PE1’s IPv6
Routing Type=TBD, SL=11,b=1,nB
sub-tree from P1 to PE2-PE9
IP multicast datagram

IPv6 Packet Received by P1
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

IPv6 header
MRH
P1 to P2, P3,
PE8, PE9

Egress PE8: (SL == 0):
Decaps, sends it to IP multicast forwarding
Comments
Request for Adoption