Designed Routing in BIER Forwarding
draft-zhang-bier-designed-routing-00

BIER WG
IETF94# Yokohama

Sandy Zhang
Bo Wu
Problem Statement

- BIER forwarding (BIER-Arch)

Suppose a multicast flow, ingress node is A, egress nodes are D, F and K. According to the shortest path forwarding, the multicast flow will be forwarded through paths: A→B→C→D; A→E→F, A→G→H→K. Obviously, it is not an optimal route.
Problem Statement

- BIER forwarding (Optimal)

The optimal path, A—E—F—C/H—D/K

Problem: How to forward the packet through the optimal path?
The packet will be encapsulated by path-list.
There is a flag in the BIER header, it indicate that there is a path-list behind the BIER header.
The path-list format is a new type TLV.

The packet will be forwarded by the nodes according to the BIER forwarding table and the path-list.
Every node on the path-list will treat the path-list in special method.
Solution Statement

- **Forwarding packet**

  - For the ingress node A: Level1, E; Level2, F; Level3, C/H; Level4, D/K.
  - The nodes are classified by the distance from the ingress node.
  - Use a flag in Reserved field to indicate that there is a path-list behind the BIER header.

The path-list is encapsulated:
```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|       Current level :1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--
|        level : 1     |                 Bit String : E |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--
|        level : 2     |                 Bit String : F |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--
|        level : 3     |                 Bit String : C/H|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--
|        level : 4     |                 Bit String : D/K|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--

Path-list:
A
  E
  level1
  F
  level2
  C/H
  level3
  D/K
  level4
```
A checks the path list, forwards the packet to level 1 node E, then, changes the current level to 2. The destination is D/F/K.

E checks the path list, forwards the packet to level 2 node F, then, changes the current level to 3. The destination is D/F/K.

F decapsulates the packet and forwards out.
F checks the path list, forwards the packet to level 3 node C, then, changes the current level to 4. The destination is D.
F checks the path list, forwards the packet to level 3 node H, then, changes the current level to 4. The destination is K.

C checks the path list, forwards the packet to level 4 node D, then, changes the current level to 5. The destination is D.
H checks the path list, forwards the packet to level 4 node K, then, changes the current level to 5. The destination is K.
Solution Statement

• Forwarding algorithm

• A finds that it should forward the packet to E according to the path-list, A picks the item “E/F” which next-hop is E out from the forwarding table;
• A uses the destination ”D/F/K” AND the item “E/F”. Gets the result and INVERSE it, then AND the destination, gets the “Resv-nodes” which includes “D/K”.
• A uses the destination ”D/F/K” AND the item “E/F”, and OR the “Resv-nodes””D/K”, then forwards the packet to E.
• The process in E is similar.

PS, What are “Resv-nodes”?
• When node forwards packet according to the picked items in BIER forwarding table, some of the destination may be lost. We should get them back. These lost nodes named “Resv-nodes”.
When the packet reaches F, because F is one of the egress nodes, F decapsulates the packet and forwards out. The destination changes to “D/K”;

- F picks out two items, “C/D” which next-hop is C; “H/K” which next-hop is H.
- F uses the destination “D/K” AND the item “C/D”, gets “D”; F uses the “D/F/K” AND the item “H/K”, get “K”. Mix the two result, gets “D/K”. Then INVERSE the mixed result, AND the destination “D/K”, then the “Resv-nodes” is NULL.
- F uses the destination D AND the item “C/D”, gets the result, OR the “Resv-nodes”, gets “D”, then forwards the packet to D.
• The packet reaches C, C picks out the items “D” which next-hop is D.
• C uses the destination “D” AND the item “D”, gets “D”; Then INVERSE the result, AND the destination “D”, then the “Resv-nodes” is NULL.
• C uses the destination “D” AND the item ”D”, gets the result, OR the “Resv-nodes”, get “D”, then forwards the packet to D.
• The process in H is similar.
• **Advantage**

• The forwarding table is the same as the table that is defined in BIER-Arch.

• The bitstring in path-list is the same as the bitstring that is defined in BIER-Arch.

• When the crossover node is as close as possible to the egress nodes, the forwarding will be most efficient.
Any comments are welcome
Thanks!