Proposed Extensions to Tunnel Encapsulation Attribute

draft-ietf-bess-bgp-multicast-controller

Presented by Z. Zhang for IDR, IETF108

Tunnel Encapsulation Attribute (TEA)

- A TEA can include a list of tunnels of various types
- When attached to a unicast route, matching traffic is sent out of one of the listed tunnels
- Proposed extensions to TEA in draft-ietf-bess-bgp-multicast-controller
  - The draft is about controllers signaling multicast state onto a router
    - How traffic for a multicast tree/tunnel is replicated
  - When a TEA is attached to a multicast route, matching traffic is replicated out of listed tunnels
  - A few new tunnel types and sub-TLVs for multicast purpose are proposed

Any-encapsulation Tunnel

- Existing tunnel types are all associated with an encap type
- The new Any-encapsulation tunnel means any encapsulation can be used
  - Only need a remote endpoint address sub-TLV
- Examples
  - Native IP multicast forwarding – IP traffic from an upstream node replicated to a bunch of directly connected downstream node
    - TEA lists a bunch of any-encapsulation tunnels, each with the interface address of the downstream node for the remote endpoint address sub-TLV
  - Native or labeled multicast forwarding – traffic needs to be tunneled from an upstream node to a non-adjacent downstream node via ***any*** available tunnel
    - Any type/instance of tunnel to the listed remote endpoint address can be used

Load-balancing Tunnel

- Consider that there are M ways to reach a downstream node from an upstream node, and the controller wants to specify any of the N (N < M) specific ways to be used
  - If the any-encap tunnel was used – all M ways would be used
  - Introduced Load-balancing tunnel for this purpose
- A load-balancing tunnel lists a few tunnels, and itself is a member tunnel of a TEA attached to a multicast route
  - Traffic is replicated out of all member tunnels for the TEA
  - For the load-balancing tunnel in the TEA, only one of its member tunnels is used to send traffic

RPF sub-TLV

- Unidirectional multicast forwarding state includes an upstream (incoming) and a bunch of downstreams (outgoing)
  - Encoding downstreams in TEA is quite natural
  - Encoding upstream in TEA is quite reasonable/convenient as well
    - just add a RPF sub-TLV
- The RPF sub-TLV indicates the tunnel is for ***upstream/incoming***
- This is actually applicable to bidirectional/MP2MP as well
  - Forwarding state includes an upstream and a bunch of downstreams
    - Each for both incoming and outgoing traffic

MP2MP MPLS Support

- An MPLS tunnel can include a label stack sub-TLV
  - For sending outgoing traffic
- In case of MP2MP, another label stack is needed
  - For receiving incoming traffic
  - So we define a new sub-TLV “incoming label stack”
    - Like the existing label stack sub-TLV, just a different type
    - A tunnel includes both types of label stack, one for incoming and one for outgoing traffic
- The new “incoming label stack” sub-TLV is also used for P2MP
  - For the upstream tunnel (with the RPF sub-TLV)
    - It won’t have the regular label stack (that is for outgoing traffic)
  - Downstream tunnels (w/o the RPF sub-TLV) only has the regular label stack

Tree Label sub-TLV

- In case of MPLS tunnel, the label stack sub-TLV for the outgoing traffic can include both the tree-identifying label and the transport labels
- However one may want to use any-encap tunnel
  - w/o specifying transport label stack, yet still need to specify the tree label
  - Tree Label sub-TLV is proposed for that purpose
- The final outgoing label stack for a tunnel is obtained as following
  - First push the tree label (if the sub-TLV is present)
  - The push the transport label stack, which is obtained as following
    - As specified in the label stack sub-TLV (if present), or,
    - As returned by the lookup of the remote endpoint address

Summary

- Two new tunnel types
  - Any-encap tunnel and Load-balancing tunnel
- Three new sub-TLVs
  - RPF sub-TLV
  - Incoming Label Stack sub-TLV
  - Tree Label sub-TLV
- Draft's home is BESS but obvious the TEA changes need to be blessed here
  - Comments appreciated
  - Early allocation of codepoints requested