draft-xu-mpls-unified-source-routing-instruction

xuxiaohu@huawei.com, stewart.bryant@gmail.com, robert@raszuk.net, uma.chunduri@gmail.com, luismiguel.contrerasmurillo@telefonica.com, luay.jalil@verizon.com, hamid.assarpour@broadcom.com
Goals and MUST NOTs

• Goal: Carry MPLS-SR packets across network segments that do not support MPLS.
• Goal: A minimalist approach to SFC.
• Goal: Use existing hardware and IETF Specifications.
• Goal: Provide a common approach to all of the above.
• MUST NOT require MPLS control protocols outside the MPLS domain.
A Common encapsulation

+----------------------+
|      IP header       |
+----------------------+
|       Optional       |
|         UDP          |
+----------------------+
|       MPLS Label     |
|                      |
|        Stack         |
|                      |
+----------------------+
|       Payload        |

- The payload is outside the scope of this proposal.

- The KEY part of the MPLS Label Stack is that it is hardware friendly, existing way of carrying a series of 20 bit instructions (SFid, SID, etc).

- The Optional UDP header is to provide an ECMP method that works with existing IP forwarders.

- The IP header can be IPv4 or IPv6.
Tunnelling MPLS-SR over an IP Network

B, F & D NOT MPLS-SR Capable
L(E)-> Next Hop E
L(G)-> Next Hop G
L(H)-> Next Hop H
Detail

• Tunnelling of MPLS-SR has previously been described at IETF
• There is a bunch of detail that is an exercise for the reader authors.
• It is conceptually simple and we believe that there are no showstoppers.
Building an Service Function Chain

B, F & D are IP transit nodes

E, G & H are nodes hosting SF

L(E) -> Execute SF E
L(G) -> Execute SF G
L(H) -> Execute SF H
Detail

• Building an SFC using MPLS has previously been described at IETF
• There is a bunch of detail that is an exercise for the authors.
• It is conceptually simple and we believe that there are no showstoppers.
• IMPORTANT – It is not necessary to turn on any MPLS control function to make this – SDN for example can be used.
• Multiple SFs can be served via the same node – just put the labels in.
• This requires the mapping of SF to 20 bit label and SF host address (simple) see the next slide for an alternative approach.
Building a More Complex SF Chain

B, F & D are IP transit nodes
E, G & H are nodes hosting SF

L(E1)-> Execute SF E1
L(E2)-> Execute SF E2
L(G)-> Goto G
L(G1)-> Execute SF G1
L(H)-> Goto H
L(H1)-> Execute SF H1
Detail

• In this example the host identity is explicitly encoded as a label, preferably a domain wide aka SR Nodal Label.
• We could of course use the same technique in a pure MPLS network.
• There is lots of detail to work through but the principle is clear.
SR in an IP Network

B, F & D Simply forward IP packets

E & G Interpret the 20 labels as:

L(G)→ Next Hop G
L(H)→ Next Hop H
Detail

- Does not require a new encapsulation definition
  - MPLS over IP [RFC4023]
  - MPLS-over-UDP [RFC7510]
- Compact Instruction format 20 bits per SID.
- Compact format means much shorter reach into packet by forwarder.
- IMPORTANT – It is not necessary to turn on any MPLS control function to make this work.
- Can be deployed as in interim until full featured SRv6 is available on more platforms and where IPv4 support is required.
- Again there is a bunch of detail that is an exercise for the reader authors.
Conclusion

• A single compact data plane format can support
  • Interconnection of disjoint MPLS-SR islands
  • Service Function Chaining
  • Segment Routing version X.

• The required data-plane specifications mostly exist.

• It is important to focus on the 20 bit instructions, not the packaging of those instructions into a RFC3032 format. This packaging is just a convenience.

• It is also important to remember that the use of RFC3032 format DOES NOT imply that we always use the MPLS control protocols.

• This unification approach has many benefits, and is worthy of further development.
Questions?