Building blocks for Network Slice Realization in SR

Zafar Ali, Cisco Systems (zali@cisco.com) - Presenter
Clarence Filsfils, Cisco Systems (cfilsfil@cisco.com)
Pablo Camarillo, Cisco Systems (pcamaril@cisco.com)
Francois Clad, Cisco Systems (fclad@cisco.com)
Daniel Voyer, Bell Canada (daniel.voyer@bell.ca)
Satoru Matsushima, Softbank, (satoru.matsushima@g.softbank.co.jp)
Reza Rokui, Nokia, (reza.rokui@nokia.com)
Scope of the Draft

- **Scope**
  - Informational Draft
  - Lists essential building blocks needed for network slicing
  - Explains how these building blocks interact, seamlessly

- **Goals**
  - Scaling
  - Incremental deployments
History of the Draft

• History
  – Rev 0 was published in July 2018
  – Rev 2 was presented at IETF106 (in Spring WG)
  – Rev 4 was presented at IETF110 (in Spring WG)
    > Addressed comments received
    > Published as TEAS-Spring draft
Draft Summary

• Building Blocks
  – SR Policy - with or without Flexible Algorithm – Flexible Algorithm
  – TI-LFA with O(50 msec) protection
  – SR VPN
  – SR Service Programming (NFV, SFC)
  – OAM and Performance Management (PM)
  – QoS
  – Orchestration at the Controller
  – Stateless Slice identification (SLID)

• These building blocks need to work together, seamlessly
SLID Attributes

• SLID enables the differentiate treatment
  – QoS/ DiffServ policy on a per SLID

• SLID construct is like QoS
  – Independent of Routing and Topology

• Stateless

• Backward compatible
  – Incremental deployments
Seamless Building Blocks Example

- **Flex-Algo and TI-LFA**
  - The backup path is optimized per Flex-Algo

- **Flex-Algo, TI-LFA and SLID**
  - Slid does not create a new instance of Flex-Algo
    > Scalable
  - TI-LFA works seamlessly for each SLID
    > The slid is stateless
    > Backup paths provide differentiated treatment

Underlay building blocks shared among SLIDs
Seamless Building Blocks (cont’ed)

• SLID work seamlessly with other building blocks for scaling
  – Flex Algo
    > Like in previous slide
    • Orange & red Flex Algo and Green & Blue SLID works seamlessly
  – VPN
  – SR Policy (with or without flex algo)
  – QoS/ DiffServ policy, etc.

• SLID is a differentiated behavior at a node
  – Not too many SLIDs are needed
    > Scaling
**SLID for SRv6**

- **Reference**
  - draft-filsfils-spring-srv6-stateless-slice-id

- **Ingress PE**
  - Encapsulates an outer IPv6 header and optional SRH
  - MAY classify the traffic to a slice and sets the following in the outer IPv6 header
    - SPI bit (SLID Presence Indicator) in the TC
    - SLID in the 8 MSB of the Flow Label

- **Per Slice Differential Treatment**
  - The SLID is used to apply per-slice policies

- **Backward Compatible**
  - Node not supporting SLID provides slicing using non-SLID building blocks (default SLID)
SLID for MPLS

• Reference
  – draft-decaenempls-slid-encoded-entropy-label-id

• Similar to draft-filsfils-spring-srv6-stateless-slice-id for MPLS networks

• Ingress PE
  – MAY classify the traffic to a slice and sets the following in the entropy label of MPLS label stack:
    > SLID in the x MSB of the entropy label
    > SPI bit (SLID Presence Indicator) in the one bit of TTL field (ELC)

• Per Slice Differential Treatment
  – The SLID is used to apply per-slice policies

• Backward Compatible
  – Node not supporting SLID provides slicing using non-SLID building blocks (default SLID)
Next Steps

• The authors would like the WG provide comments
• The authors would like the WG to adopt the document