Realizing Network Slices in IP/MPLS Networks

draft-bestbar-teas-ns-packet-00

Tarek Saad Juniper Networks

Vishnu Pavan Beeram Juniper Networks

Contributors: Colby Barth, Srihari Sangli, Chandra Ramachandran

IETF-109, November 2020, Virtual

Agenda

- Introduction
- Slice Per Hop Definition
- Solution approaches
- Next Steps

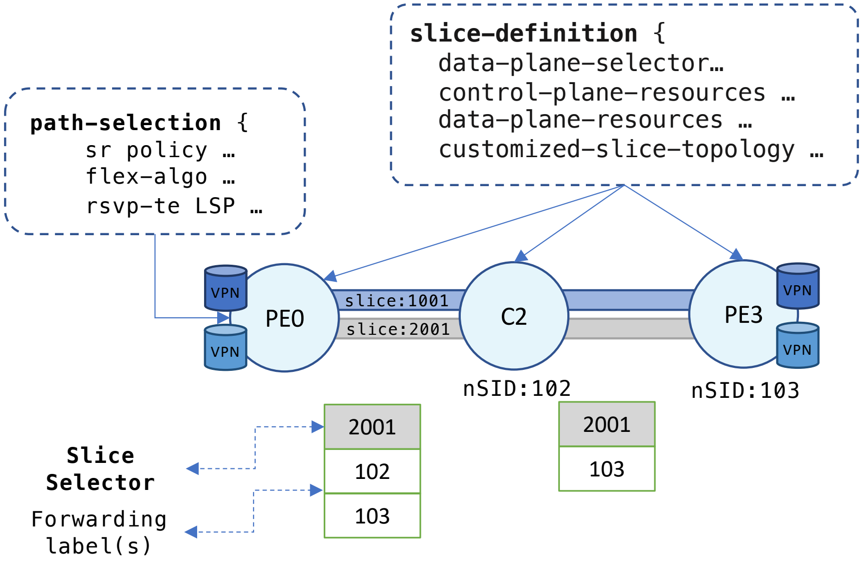
Introduction

- Solution to realize network slicing in IP/MPLS networks
  - Based on DiffServ principles
    - Slicing from a device and network resource level perspective
  - Ensure proper placement of paths and respective treatment of traffic traversing network slice resources
    - Agnostic to the path control technique used in the network slicing domain

- Multiple network slices can be realized on top of a shared physical IP/MPLS infrastructure network
  - Dynamically created and managed
- Traffic traversing shared network resources may require specific treatment to meet target SLOs
  - A Slice Selector within a packet identifies slice traffic
  - Device specific slice resources instantiated on-demand: Slice Queues and Slice PHB
- Placement of traffic on slice resources optimized based on network slice resource utilization
  - Slice-aware TE
- Differentiation of traffic within same network slice is also possible
  - Diffserv Class Selector (CS) in the packet distinguishes traffic within same slice

Solution Overview

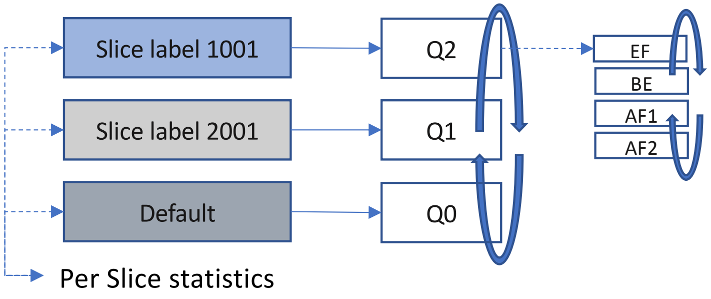
- Approaches to network resource slicing:
  - Control plane slicing, Data plane slicing, mix of Control and Data plane slicing
  - Hybrid options possible
- An integral Slice Per Hop Definition, encompassing:
  - Data plane Slice Selector
  - Data plane resources (H-QoS PHBs)
  - Control plane resources (bandwidths, priorities, shared-resource groups)
  - Slice specific topology facilitating Slice aware TE
- Multiple options for Slice Per Hop Definition dissemination
  - Programmed via Netconf/gRPC interface(s)
  - Exchanged in IGP and/or BGP
  - Configured directly on device(s)
- Multiple options for path selection and control techniques within a Slice (SR Policy, Flex-algo, RSVP-TE LSP, …)



```
path-selection {
    sr policy …
    flex-algo …
    rsvp-te LSP …
```

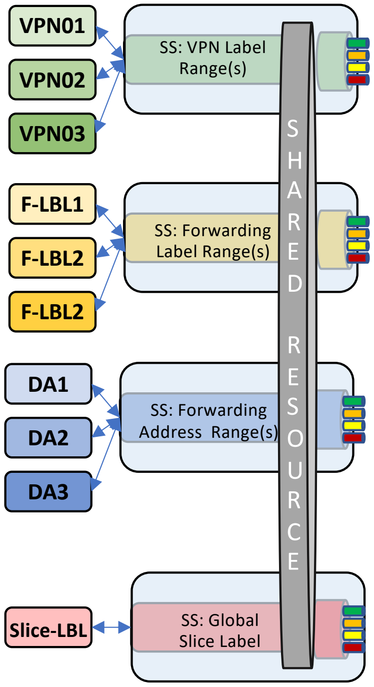
```
slice-definition {
  data-plane-selector…
  control-plane-resources …
  data-plane-resources …
  customized-slice-topology …
```

Automatic Slice Resource Allocations & per Slice QoS isolation



Slice Per-Hop Definition (1/2)

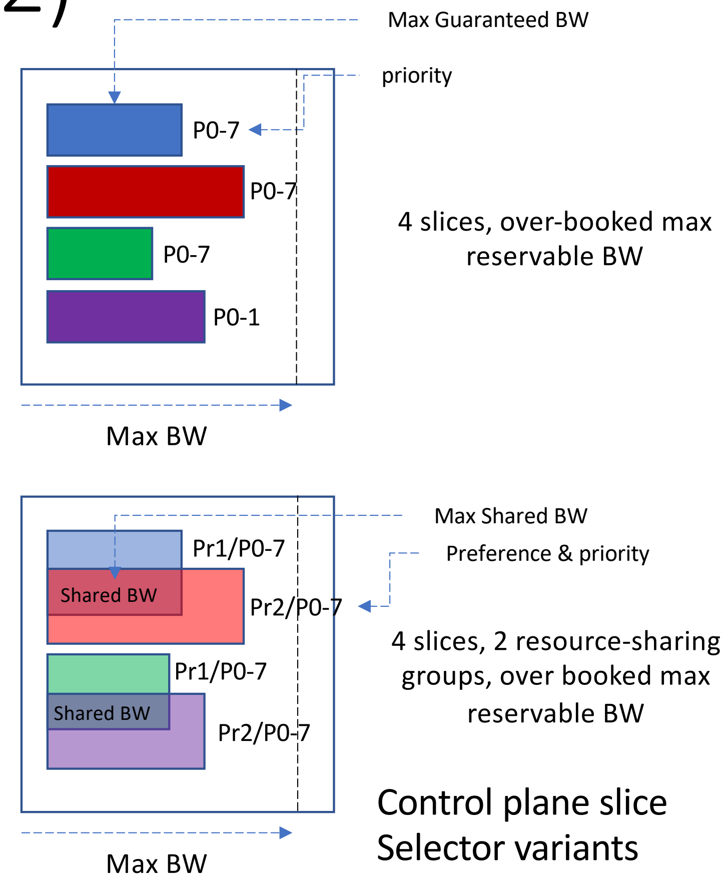
- Data plane Slice Selector
  - A range of MPLS forwarding labels (for each destination) mapping to a Slice
  - A range of MPLS VPN service labels mapping to a single Slice
  - A single Global label mapping to a Slice
  - A set of IP destinations mapped to a slice
  - A multi-field packet selector
- Data plane resources (H-QoS PHBs)
  - A Slice QoS profile or a reference to device local profile
  - Different profiles to multiple interfaces possible



Data plane Slice Selector variants

Slice Per-Hop Definition (2/2)

- Control-plane resources management
  - Slice-aware Bandwidth Engineering
  - Preference-based preemption of Slice-aware TE paths
  - Resource sharing
  - Protection
- Slice topology for Slice aware TE
  - Slice Membership: Topological elements (link/node) participating in a slice
  - Single Topology can cater to multiple slices

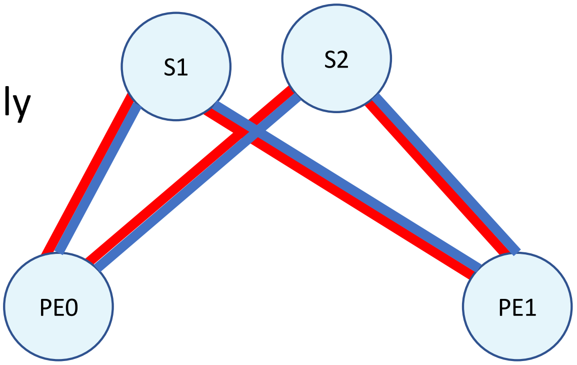


Control plane slice Selector variants

Slicing uses transport or Slice label inferred PHB

Data plane only network slicing

- Data plane network resource slicing
  - Slice Selector – identifies packets belonging to a specific slice
  - Per slice CoS profile is applied on participating links and nodes
  - Transit nodes classify incoming traffic (e.g. using Slice label) and apply per slice scheduling
- Control plane
  - No control plane awareness of slice resources
  - No slice-aware path placement/TE
- Use-case
  - Suitable when no BW engineering is required and ECMP is leveraged between endpoints (e.g. Spine/Leaf deployment)
  - Does not address all network slice SLOs being standardized at IETF

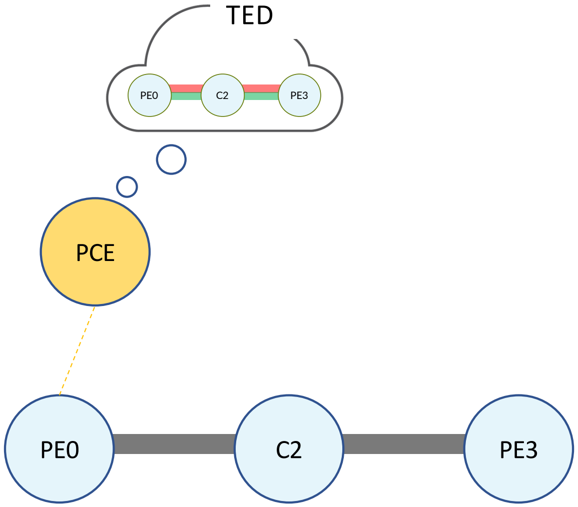


🟥 RED slice is 50% of each link
🟦 GREEN slice is 50% of each link

Control plane only network slicing

Compute & Resource management are Slice aware

- Control plane network resource slicing
  - Link resources are sliced in control plane
    - Per slice link maximum and available BW
    - Ingress router/PCE forms per slice TED using the link-state
  - Slice-aware path computation and path placement
    - CSPF uses the slice aware TED to select optimal TE path
  - Control plane preemption in case of contention on a specific link resource
    - In case of degradation of LAG, control plane can preempt LSP(s) to avoid congestion
- Data plane
  - No per slice classification of traffic or per slice PHB on transit routers
  - Policing can happen on slice incoming traffic

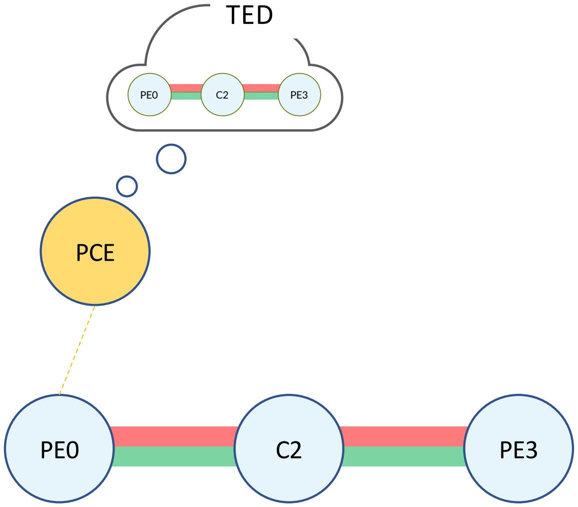


Simple, consistent, 4-8Q PHBs may still be deployed

Slice label inferred PHB
Compute & resource
management are Slice aware

Control and data plane network slicing

- Control plane and data plane network slicing
  - Combination of the previous two
  - Slice-aware TE enables ingress/PCE to do proper placement of LSPs based on per slice link available BW
  - Data plane Per-Hop Behavior on transit nodes provides guarantees in case of congestion on a link
  - Covers strict and shared resource slice isolation requirements



Next Steps

- Slice-aware TE
  - Protocol Extensions -- Coordination with relevant drafts
- Request review and feedback

YANG Data Model for Network Slice Per-Hop Definition

draft-bestbar-teas-yang-ns-phd-00

Tarek Saad Juniper Networks

Vishnu Pavan Beeram Juniper Networks

Contributors: Colby Barth, Srihari Sangli, Chandra Ramachandran

IETF-109, November 2020, Virtual

Overview

- YANG data model for programming Network Slice Per Hop Definition (Slice-PHD) on IP/MPLS devices
  - Multiple Slice Selector options
  - Flexible and hierarchical Slice PHB(s)
  - Covers resource management in control plane and data plane
  - Mapping of a slice to a logical topology

Network Slice Per Hop Definition Model

Model Structure

```
module: ietf-network-slice-phd
      +--rw network-slicing!
        +--rw network-slice-phbs
        |  +--rw network-slice-phb* [id]
        |     ............
        +--rw network-slices
           +--rw network-slice* [name]
              |  ............
              +--rw slice-resource-reservation
              |  ............
              +--rw slice-selectors
              |  +--rw slice-selector* [id]
              |     ............
              +--rw slice-phb?                    ns-phb-ref
              |  ............
              +--rw slice-membership
                 ............
```

Slice-PHDs

network-slices container

- Key elements
  - Slice Resource Reservation
  - Slice Selectors
  - Slice PHB
  - Slice Membership

Slice-PHBs

network-slice-phbs container

- Referenced by Slice-PHDs

Network Slice Per Hop Definition Model

Slice Per-Hop-Behaviors

```
+--rw network-slice-phbs
|  +--rw network-slice-phb* [id]
|     +--rw id                                    uint16
|     +--rw (profile-type)?
|        +--:(profile)
|        |  +--rw profile?                        string
|        +--:(custom-profile)
|           ............
```

Slice-PHBs container (network-slice-phbs)

- Carries a list of Slice-PHB entries
- Slice-PHB entry
  - Referenced by one or more Slice-PHD
  - Options:
    - Reference to a generic PHB profile
    - Custom PHB profile

Network Slice Per Hop Definition Model

Slice Resource Reservation

```
+--rw slice-resource-reservation
|  +--rw preference?                        uint16
|  +--rw (max-bw-type)?
|  |  +--:(bw-value)
|  |  |  +--rw maximum-bandwidth?           uint64
|  |  +--:(bw-percentage)
|  |     +--rw maximum-bandwidth-percent?
|  |             rt-types:percentage
|  +--rw shared-resource-groups*            uint32
|  +--rw protection
|     +--rw backup-slice-id?                  uint32
|     +--rw (backup-bw-type)?
|        +--:(backup-bw-value)
|        |  +--rw backup-bandwidth?           uint64
|        +--:(backup-bw-percentage)
|           +--rw backup-bandwidth-percent?
|                   rt-types:percentage
```

slice-resource-reservation Container

- Slice-aware Bandwidth Engineering
- Preference-based preemption of Slice-aware TE paths
- Sharing of resources amongst a group of slices
- Slice Protection

Network Slice Per Hop Definition Model

Slice Selectors

```
+--rw slice-selectors
|  +--rw slice-selector* [id]
|     +--rw id         uint16
|     +--rw mpls
|     |  +--rw (ss-mpls-type)?
|     |     +--:(label-value)
|     |     |  +--rw label?
|     |     |  |       rt-types:mpls-label
|     |     |  +--rw label-position?          identityref
|     |     |  +--rw label-position-offset?   uint8
|     |     +--:(label-ranges)
|     |        +--rw label-range* [index]
|     |           +--rw index                    string
|     |           +--rw start-label?
|     |           |       rt-types:mpls-label
|     |           +--rw end-label?
|     |           |       rt-types:mpls-label
|     |           +--rw label-position?
|     |           |       identityref
|     |           +--rw label-position-offset?   uint8
|     +--rw ipv4
|     |  +--rw destination-prefix*   inet:ipv4-prefix
|     +--rw ipv6
|     |  +--rw (ss-ipv6-type)?
|     |     +--:(ipv6-destination)
|     |     |  +--rw destination-prefix*
|     |     |          inet:ipv6-prefix
|     |     +--:(ipv6-flow-label)
|     |        +--rw slid-flow-labels
|     |           +--rw slid-flow-label* [slid]
|     |              +--rw slid       inet:ipv6-flow-label
|     |              +--rw bitmask?   uint32
|     +--rw acl-ref*   ns-acl-ref
```

slice-selectors Container

- Set of data plane field selectors
- Slice Selector (SS)
  - Identify packets belonging to the given network slice
  - 16-bit ID
    - SS with the lowest ID is the default used by all the topological elements that are members of the given network slice
      - Other entries are used to override the default on select topological elements

Network Slice Per Hop Definition Model

Slice Membership

```
+--rw slice-membership
   +--rw filter-policies
      +--rw filter-policy* [id]
      +--rw id
      |       uint16
      +--rw (filter-type)?
      |  +--:(topology-ref)
      |  |  +--rw (topo-ref-type)?
      |  |     +--:(algo-id)
      |  |     |  +--rw algo-id?                 uint8
      |  |     +--:(te-topo-id)
      |  |        +--rw te-topology-identifier
      |  |           +--rw provider-id?   te-global-id
      |  |           +--rw client-id?     te-global-id
      |  |           +--rw topology-id?
      |  |                   te-topology-id
      |  +--:(custom-topology)
      |     +--rw include
      |     |  +--rw link-affinity*   string
      |     |  +--rw link-name*       string
      |     |  +--rw node-prefix*     inet:ip-prefix
      |     |  +--rw as*              inet:as-number
      |     +--rw exclude
      |        +--rw link-affinity*   string
      |        +--rw link-name*       string
      |        +--rw node-prefix*     inet:ip-prefix
      |        +--rw as*              inet:as-number
      +--rw slice-selector?
      |       ns-ss-ref
      +--rw slice-phb?
              ns-phb-ref
```

slice-membership Container

- Set of filtering policies
  - Determine which topological elements belong the specific network slice
- Filtering Policy
  - Reference a predefined topology (or)
  - Specify rules to construct customized topology
- Slice members can optionally override the default Slice-PHB and/or the default slice selector.

Next Steps

- Request review and feedback