

YANG Data Model for Slice Policy

draft-bestbar-teas-yang-slice-policy-00

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

Tarek Saad, Vishnu Pavan Beeram ^{Presenter} Juniper Networks

Bin Wen

Comcast

Daniele Ceccarelli

Ericsson

Shaofu Peng, Ran Chen

ZTE Corporation

Luis M. Contreras

Telefonica

Xufeng Liu

Volta Networks

Contributors: Colby Barth, Srihari Sangli, Chandra Ramachandran

Overview

- Slice policy enables instantiation of mechanisms in support of IETF network slices.
- Enforcement of the slice policy results in the creation of a slice aggregate.
- Slice policy dictates how a slice aggregate can be realized in IP/ MPLS networks using one of three modes.
 - Partitioning of the shared network resources can be achieved in:
 - (a) Just the data plane or in
 - (b) Just the control plane or in
 - (c) Both the control and data planes

Slice policy modes (a) and (c):

- Require the forwarding engine to identify the traffic belonging to a specific slice aggregate and to apply the corresponding Per-Hop Behavior (PHB)
 - Identification of the slice aggregate that the packet belongs to and the corresponding forwarding treatment that needs to be applied to the packet are dictated by the slice policy.

▪ Draft defines a YANG data model for the management of slice policies on slice policy capable nodes and controllers in IP/MPLS networks.

- The latest (renamed) version aligns with the terminology used in <draft-ietf-teas-ietf-network-slice-definition>, <draft-nsdt-teas-ns-framework> and <draft-bestbar-teas-ns-packet>.
- Addresses all comments received so far.

▪ Slice policy modes (b) and (c):

- Require the distributed/ centralized resource reservation manager in the control plane to manage slice aggregate resource reservation.
 - The provisions for enabling slice aggregate aware traffic engineering are dictated by the slice policy.

▪ Slice policy modes (a), (b) and (c):

- Require the topology associated with the slice aggregate to be specified.
 - The rules for determining which topological elements cater to the slice aggregate are dictated by the slice policy.

Model Structure

```
module: ietf-slice-policy
  +--rw network-slicing!
    +--rw phbs
      | +--rw phb* [id]
      | .....
    +--rw topology-filters
      | +--rw topology-filter* [name]
      | .....
    +--rw slice-policies
      +--rw slice-policy* [name]
        + .....
      +--rw resource-reservation
        | .....
      +--rw slice-selectors
        | +--rw slice-selector* [index]
        | .....
      +--rw phb? ..... slice-policy-phb-ref
      +--rw member-topologies
        +--rw member-topology* [topology-filter]
          .....
      .....
```

- Top-level container includes placeholders for:
 - Set of Slice Policies
 - Set of PHBs that are referenced by the slice policies
 - Set of topology filters that are referenced by the slice policies.

Per-Hop-Behaviors

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

- 'phbs' container:
 - Carries a list of PHB entries.
 - Each of these entries can be referenced by one or more slice policies.
 - A PHB entry can either
 - Carry a reference to a generic PHB profile available on the node or
 - Carry a custom PHB profile.
 - Custom PHB profile includes attributes to construct a slice aggregate specific QoS profile and any classes within it.

Topology Filters

```
+--rw topology-filters
|   +--rw topology-filter* [name]
|       +--rw name string
|       +--rw (topology-filter-type)?
|           +--:(standard-topology)
|               |   +--rw (standard-topo-type)?
|               |       +--:(flex-algo)
|               |           |   +--rw algo-id? uint8
|               |           |   +--rw mt-id? uint16
|               |           +--:(te-topo)
|               |               +--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-any
|                   |   +--rw link-affinity* string
|                   |   +--rw link-name* string
|                   |   +--rw node-prefix* inet:ip-prefix
|                   |   +--rw as* inet:as-number
|               +--rw include-all
|                   |   +--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
```

- 'topology-filters' container:
 - Carries a list of topology filters.
 - Each topology filter entry could either
 - Reference a predefined topology (or)
 - Specify the rules to construct a customized topology using a set of include-any, include-all and exclude filters.

Slice Policies

```
+--rw slice-policies
  +--rw slice-policy* [name]
    +--rw name                string
    +--rw sa-id?              uint32
    +--rw resource-reservation
    | .....
  +--rw slice-selectors
    | +--rw slice-selector* [index]
    | .....
  +--rw phb?                  slice-policy-phb-ref
  +--rw member-topologies
    +--rw member-topology* [topology-filter]
    .....
```

- 'slice-policies' container:
 - Carries a list of slice policies.
 - Each slice-policy entry is identified by a name and holds the set of attributes needed to instantiate a slice aggregate.
 - Key elements of slice-policy entry:
 - Resource Reservation
 - Slice Selectors
 - Per-Hop-Behavior
 - Member Topologies

Slice Policy – Resource Reservation

```
+--rw 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-sa-id?                           uint32
|   |   +--rw (backup-bw-type)?
|   |   |   +--:(backup-bw-value)
|   |   |   |   +--rw backup-bandwidth?              uint64
|   |   |   +--:(backup-bw-percentage)
|   |   |       +--rw backup-bandwidth-percent?
|   |   |           rt-types:percentage
```

- 'resource-reservation' container:
 - Carries data nodes that are used to support slice aggregate aware bandwidth engineering.
 - Facilitates preference-based preemption of slice aggregate aware TE paths, sharing of resources amongst a group of slice aggregates and backup slice aggregate path bandwidth protection.

Slice Policy – Slice Selectors

```
+--rw slice-selectors
|   +--rw slice-selector* [index]
|       +--rw index          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*    slice-policy-acl-ref
```

- 'slice-selectors' container:
 - Carries a set of data plane field selectors which are used to identify the packets belonging to the given slice aggregate.
 - Each slice-selector entry in the list has an index associated with it.
 - The slice selector with the lowest index is the default slice selector used by all the topological elements that are members of the given slice policy.
 - The other entries are used only when there is a need to override the default slice selector on some select topological elements.

Slice Policy – Per-Hop-Behavior

```
+ - -rw phb? slice-policy-phb-ref
```

- 'phb' leaf:
 - Carries a reference to the PHB that needs to be applied for the given slice aggregate.
 - Unless specified otherwise, this is the default phb to be used by all the topological elements that are members of the given slice policy.

Slice Policy – Member Topologies

```
+--rw member-topologies
  +--rw member-topology* [topology-filter]
    +--rw topology-filter
      |      slice-policy-topo-filter-ref
    +--rw slice-selector-override?  slice-policy-ss-ref
    +--rw phb-override?
      slice-policy-phb-ref
```

- 'member-topologies' container:
 - Consists of a set of member topologies.
 - Each member topology references a topology filter.
 - The topological elements that satisfy the membership criteria can optionally override the default PHB and/or the default slice selector.

Next Steps

- Request further review and feedback.
- Draft provides the foundation for realizing slice aggregates
 - Authors believe it is sufficiently baked (ready to be considered for WG adoption).