Private MPLS Namespaces

Signaled using BGP VPNs

Kaliraj Vairavakkalai
Juniper Networks

(https://www.ietf.org/id/draft-kaliraj-bess-bgp-sig-private-mpls-labels-00.txt)
Problem

- How to achieve “Predictable MPLS label allocation”

  - Install route with desired “label-value” and “forwarding-behavior” in a router’s MPLS forwarding-plane

  - API to mpls-router’s forwarding-plane, for use by external label-allocators (another router or a controller)

- Possible solutions:
  - Label range reservations
  - Label contexts, RFC 5331 (chosen here; why? slide15)
MPLS context-table, enables “off-box” lbl-allocator

- Router
  - Nhop1
  - VRF1
  - ge-0/0/0
- On-box Label-Allocator
  - ge-0/0/0
- Off-box Label-Allocator
  - Lo0
  - ge-2/0/0

 MPLS.0
- mpls.0
- a.mpls.0

Payload
- CL1
- PL1

Pfx1, PNH: CPNH1, Label PL1
CPNH1, PNH: Lo0, Label CL1

a.mpls.0
- PL1 -> VRF1
- PL2 -> Nhop1

mpls.0
- L1 -> VRF1
- L2 -> Nhop1
- CL1 -> a.mpls.0 (context-lbl)
Private MPLS namespace (mpls-plane)

- Router creates MPLS name-space (context-table) for an Application, and propagates in the network using BGP.

- App’s handle to the name-space:
  - “context-label” or “private-interface” : for forwarding-control
  - “context-route-target” : for routing-control.

- The network thus provides a ‘private MPLS-plane’ abstraction to Apps, over a shared MPLS-network.

- A ‘mpls-plane’ is identified by a “Context-PNH”. Service routes bind to this PNH, instead of PE-lo0, to use a mpls-plane.
Protocol extensions

• New RFC-4364 style AFI-SAFIs to exchange Private-Labels
  (https://www.ietf.org/id/draft-kaliraj-bess-bgp-sig-private-mpls-labels-00.txt)

• A new BGP “Multi-Nexthop” attribute to carry Forwarding-nexthop information for the Private-Labels
  (https://www.ietf.org/id/draft-kaliraj-idr-multinexthop-attribute-00.txt)
  • Ability to carry multiple nexthops in one bgp-route
  • Each nexthop-leg expressed with “FwdSemantic” of (Forward, Push, Pop, Swap) with qualifiers such as preference, load-balance-factor etc.
  • Facilitates the API to express forwarding-behavior for upstream-allocated labels.
Details, Route-Types

• **Route-Type-1** “Context-PNH advertisement”

  Prefix: **RD:CPNH-Address**

  Attributes:
  
  • **MultiNexthopAttr:**
    
    Push(CL1), ForwardTo(Lo0)
  
  • Route-Target identifying the mpls-plane

• Lo0 resolves over some transport tunnel (LDP, RSVP, GRE, ...)


Details, Route-Types ...

- Route-Type-2 "Private Label advertisement"

Prefix: RD:Label

Attributes:

- Multi Nexthop Attr:
  - e.g. - ForwardTo(RD:CPNH-addr)
    - Pop-n-Lookup(VRF-RD)
    - Pop-n-Fwd(10.1.0.2)

- Route-Target identifying the mpls-plane

- RD:CPNH-addr resolves over the Type-1 route in previous slide
Put it together

(PE1->PE2) Service-route with PNH: **CPNH1**, Label **PL1** (AF: 1/128)

**R1**
- bgp.mpls vpn.0
- RD1:CPNH1
- mpls.0

**Type 1** Pfx: RD1:CPNH1.

**Type 2** Pfx: RD1:**PL1**.

**PNH:** R1, Label **CL1**

**R3**
- bgp.mpls vpn.0
- PL1
- mpls.0

**Type 1** Pfx: RD1:CPNH1.

**Type 2** Pfx: RD1:**PL1**.

**PNH:** R3, Label **CL3**

**R5**
- bgp.mpls vpn.0
- RD5:CPNH1
- mpls.0
- a.mpls.0

**Type 1** Pfx: RD1:CPNH1.

**Type 2** Pfx: RD1:**PL1**.

**PNH:** RD1:CPNH1

**PE1**
- PL1
- Payload

**PE2**
- PL1
- Payload

**R3:** mpls.0
- CL3 -> Swap(CL1), ForwardTo(R1)

**R3:** bgp.mpls vpn.0
- RD1:PL1 -> ForwardTo(RD1:CPNH1)
- RD1:CPNH1 -> Push(CL1), ForwardTo(R1)

**R3:** inet.3:
- R1 -> Push(TL1), ge-x/y/z

**R5:** a.mpls.0
- PL1 -> ForwardTo(RD1:CPNH1)

**R5:** bgp.mpls vpn.0
- RD1:PL1 -> ForwardTo(RD1:CPNH1)
- RD1:CPNH1 -> Push(CL3), ForwardTo(R3)

**R5:** inet.3:
- R3 -> Push(TL3), ge-x/y/z
Use case 1: optimal traffic-fwd in virtualized environment

- "service-label mirroring" to Service-Forwarding-Helper
- SFH optimally forwards mpls-traffic from core to correct vFP
Backup slides.
Summary, nomenclature

• **Context-PNH**: IP-address identifying a private-mpls-plane. (Network-wide Unique like a Lo0)

• **Context-Label**: Locally-significant label pointing to label-context. App sends mpls-traffic with context-label to get to the right context-table.

• **Private-interface**: ge-x/y/z interface at router owned by application. App doesn’t use context-label if it is connected via private-interface

• **Context-Route-Target**: Route-target identifying the mpls-plane
(PE1->PE2) Service-route with PNH: CPNH1, Label PL1 (AF: 1/128)

**PE1**
- **Pfx:** PL1
- **PNH:** IntfAddr (AF: 3/1 Type2)

**R1**
- **Type1 Pfx:** RD1:CPNH1
- **PNH:** R1, Label L3

**R3**
- **Type1 Pfx:** RD1:CPNH1
- **PNH:** R3, Label L4

**R5**
- **PNH:** R5, Label L5

**PE2**
- **Pfx:** CPNH1
- **PNH:** IntfAddr (AF: 3/1 Type1 or 1/4)

**Bkp-slide: Show traffic detour via R7**

**R7**
- **mpls.0**
  - CL7 -> Pop, Lookup(a.mpls)

**R7**
- **a.mpls**
  - PL7 -> ForwardTo(RD7:CPNH1)

**R5**
- **mpls.0**
  - PL7 -> ForwardTo(RD7:CPNH1)

**R5**
- **bgp.mplsVpn.0**
  - RD7:CPNH1 -> Push(CL7), ForwardTo(R7)
Usecase 2: Static-bindings for Control-plane routes

- Take as example L3VPN network. Use “Context-PNH” as the PNH for service-routes, and “Private-Labels” as the service-label.
- State on end vPEs get reduced, they don’t need all PE lo0 state
- POP of service can be taken into M/W by just tweaking Private-Label-route Local-preference.
- Single-homing, multihoming of service works
- Features like PIC-edge/Egress-Protection/EPE can be provided at private-mpls layer, and be actually service agnostic.
- Inet family can be equated with L3VPN, by using Multi-next-hop attribute to advertise a “push label” for inet-uni family routes as-well.
Seamless MPLS – private mpls-planes
## Comparison

(why you should be interested)

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Label range reservation</th>
<th>Label contexts (private namespace)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaling</td>
<td>Shrinks per application usable namespace</td>
<td>Gives each application a full mpls label namespace</td>
</tr>
<tr>
<td>Coordination with existing users of MPLS global-namespace</td>
<td>Tightly coupled, range reservation and coordination required</td>
<td>Flexible. No overlap with global/other MPLS namespace users.</td>
</tr>
<tr>
<td>Security</td>
<td>nope</td>
<td>Yes. Fwd-context based spoof-check</td>
</tr>
<tr>
<td>Platform dependency (label-context forwarding support RFC5331)</td>
<td>Not required</td>
<td>Lbl-ctx Forwarding support required, only on Private-LERs</td>
</tr>
<tr>
<td>Label-stack overhead</td>
<td>No overhead</td>
<td>One extra label (Context-label)</td>
</tr>
<tr>
<td>Applicability</td>
<td>Limited (e.g. cannot be used for label-mirroring applications)</td>
<td>More flexible. As it gives full control to external allocator on label-value.</td>
</tr>
</tbody>
</table>

(Focus of this talk)
Pointers to new drafts:

- [https://www.ietf.org/id/draft-kaliraj-bess-bgp-sig-private-mpls-labels-00.txt](https://www.ietf.org/id/draft-kaliraj-bess-bgp-sig-private-mpls-labels-00.txt)
- [https://www.ietf.org/id/draft-kaliraj-idr-multinexthop-attribute-00.txt](https://www.ietf.org/id/draft-kaliraj-idr-multinexthop-attribute-00.txt)

References to existing specs:

- [RFC-4364] BGP/MPLS IP Virtual Private Networks (VPNs)
- [RFC-5331] MPLS Upstream Label Assignment and Context-Specific Label Space