Using BGP-Labeled Unicast for Segment Routing Traffic Engineering

Application Note

October 2016
Version 1.0
Contents
Managing Segment Routing Overview ................................................................. 1
Original Routing State ...................................................................................... 1
Reconfigured Routing State ............................................................................. 2
Managing Segment Routing Overview

Support for basic Source Packet Routing in Networking (SPRING) functionality has been added to the Junos OS Release 15.1F5. With this release, you can also assign a stack of labels to manage segment routing by using Border Gateway Protocol (BGP)-Labeled Unicast. This document describes how to leverage those capabilities to build a configuration for SPRING using Juniper Networks virtual MX (vMX) Series routers and an external controller.

Figure 1 shows the configuration topology. Node segment identifiers (SIDs), which are MPLS labels that uniquely identify the node, are shown in the diagram. Traffic should be forwarded between the two provider edge (PE) routers: vMX-3 and vMX-8.

Original Routing State

In its original state, vMX-3 sends traffic to vMX-8. Equal-cost multipath (ECMP) routing is used to balance traffic between vMX-4 and vMX-5 as shown in the CLI output:

```bash
user1@vMX-3> show route table VRF-1.inet.0 protocol bgp
VRF-1.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.3.0/30        *[BGP/13] 00:15:13, localpref 100, from 10.255.255.8
                AS path: I, validation-state: unverified
                to 10.0.0.2 via ge-0/0/0.0, Push 16, Push 800008(top)
                > to 10.0.0.6 via ge-0/0/1.0, Push 16, Push 800008(top)
```

![Figure 1 – SPRING Configuration Topology](image-url)
According to the traceroute in the CLI output, vMX8 is reachable from vMX3 with only three hops:

```
user1@vMX-3> traceroute routing-instance VRF-1 10.0.3.1
traceroute to 10.0.3.1 (10.0.3.1), 30 hops max, 52 byte packets
1 10.0.0.2 (10.0.0.2) 16.498 ms 10.0.0.6 (10.0.0.6) 13.613 ms 12.687 ms
   MPLS Label=800008 CoS=0 TTL=1 S=0
   MPLS Label=16 CoS=0 TTL=1 S=1
2 10.0.0.22 (10.0.0.22) 18.804 ms 9.387 ms 10.0.0.18 (10.0.0.18) 12.590 ms
   MPLS Label=800008 CoS=0 TTL=1 S=0
   MPLS Label=16 CoS=0 TTL=2 S=1
3 10.0.3.1 (10.0.3.1) 11.939 ms 13.701 ms 8.515 ms
```

Reconfigured Routing State

However, by using a segment routing technique, you can configure traffic to reach its destination point, vMX8, by taking a less obvious route.

As described previously, BGP-Labeled Unicast can announce a label stack for segment routing. In the topology example, an external controller, ExaBGP\(^1\), is used and updated with a label stack so that the new route takes the following path: vMX-5 → vMX-7 → vMX-6 → vMX-8.

```
user1@CentOS-1 ~/exabgp-3.4.16/sbin>cat ~user1/config/exabgp
neighbor 192.168.255.12 {
   group-updates true;
   local-address 192.168.255.2;
   peer-as 65000;
   local-as 65000;
   family {
      ipv4 nlri-mpls;
   }
   static {
      route 10.255.255.8/32 {
         next-hop 10.0.0.2;
         label [ 800005 800007 800006 800008 ];
      }
   }
}
```

To ensure that the first hop is taken towards vMX4, explicitly add the vMX4 interface address into the updated route’s next-hop 10.0.0.2 address. To make this address resolvable, add a labeled path to the label-switching routers (LSRs) configuration:

1. Copy the interfaces’ routes to the inet.3 table using routing information base (RIB) groups.
2. Change the priority of the BGP routes by making it higher than the ISIS/OSPF priority to ensure that your new traffic engineering (TE) route is more preferable to routes announced though ISIS/OSPF routing protocols.
using bgp-labeled unicast for segment routing traffic engineering

```
routing-options {
  interface-routes {
    rib-group inet INTERFACE;
  }
  rib-groups {
    INTERFACE {
      import-rib [ inet.0 inet.3 ];
    }
  }
  protocols {
    bgp {
      group EXABGP {
        type internal;
        preference 13;
        local-address 192.168.255.12;
        family inet {
          labeled-unicast {
            rib {
              inet.3;
            }
          }
        }
        neighbor 192.168.255.2;
      }
    }
  }
}
```

After the configuration changes, the expected full forwarding path from vMX3 to vMX8 is now:

vMX-4 → vMX-5 → vMX-7 → vMX-6 → vMX-8.

The following output shows how the new route sent using BGP- Labeled Unicast from the external controller appears in the CLI configuration:

```
user1@vMX-3> show route receive-protocol bgp 192.168.255.2 extensive
inet.3: 15 destinations, 16 routes (15 active, 0 holddown, 0 hidden)
  * 10.255.255.8/32 (2 entries, 1 announced)
    Accepted
    Route Labels: 800005(top) 800007 800006 800008
    Nexthop: 10.0.0.2
    Localpref: 100
    AS path: I

user1@vMX-3> show route table VRF-1.inet.0 protocol bgp
VRF-1.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.3.0/30        *[BGP/170] 00:00:27, localpref 100, from 10.255.255.8
  AS path: I, validation-state: unverified
  > to 10.0.0.2 via ge-0/0/0.0, Push 16, Push 800008, Push 800006, Push 800007, Push 800005(top)
```
Run the `traceroute` command to confirm that the actual data path has been changed accordingly:

```
user1@vMX-3> traceroute routing-instance VRF-1 10.0.3.1
traceroute to 10.0.3.1 (10.0.3.1), 30 hops max, 52 byte packets
1  10.0.0.2 (10.0.0.2) 18.447 ms 5.458 ms 28.701 ms
   MPLS Label=800005 CoS=0 TTL=1 S=0
   MPLS Label=800007 CoS=0 TTL=1 S=0
   MPLS Label=800006 CoS=0 TTL=1 S=0
   MPLS Label=800008 CoS=0 TTL=1 S=0
   MPLS Label=16 CoS=0 TTL=1 S=1
2  10.0.0.14 (10.0.0.14) 15.079 ms 8.238 ms 6.238 ms
   MPLS Label=800007 CoS=0 TTL=1 S=0
   MPLS Label=800006 CoS=0 TTL=2 S=0
   MPLS Label=800008 CoS=0 TTL=2 S=0
   MPLS Label=16 CoS=0 TTL=2 S=1
3  10.0.0.22 (10.0.0.22) 10.053 ms 9.496 ms 13.334 ms
   MPLS Label=800007 CoS=0 TTL=1 S=0
   MPLS Label=800006 CoS=0 TTL=3 S=0
   MPLS Label=800008 CoS=0 TTL=3 S=0
   MPLS Label=16 CoS=0 TTL=3 S=1
4  10.0.0.25 (10.0.0.25) 13.828 ms 25.271 ms 13.312 ms
   MPLS Label=800008 CoS=0 TTL=1 S=0
   MPLS Label=16 CoS=0 TTL=4 S=1
5  10.0.3.1 (10.0.3.1) 11.511 ms 42.425 ms 32.366 ms
```

For details about how to use the ExaBGP controller, click: https://github.com/Exa-Networks/exabgp

For ExaBGP configuration examples, click: https://github.com/Exa-Networks/exabgp/tree/master/qa/conf