Segment Routing

Cumulus Linux supports segment routing, also known as source routing, which provides the ability for a source node to specify the path a packet should take (traffic engineering). In some more advanced cases, you can use segment routing to have offline multiprotocol label switching (MPLS) controllers program labels into the network for traffic engineering.

Cumulus Linux provides full label-based forwarding, relying on BGP for label exchange. However, Cumulus Linux does not provide LDP interoperability for MPLS and it does not support VRFs for tenant isolation.

Contents

- Features
- Example Configuration
- Configure Segment Routing
- View the Configuration

Features

Segment routing is MPLS for the data plane only. In this EA release, Cumulus Linux does not impose the labels, the host does. The MTUs should be large enough to accommodate the MPLS shim header and label stack. Segment routing supports the following features:

- MPLS label edge router (LER) functionality for IPv4 and IPv6 routing with ECMP. An ingress LER first adds an MPLS label to an IP packet. An egress LER removes the outermost MPLS label (also called popping the label).
- MPLS label switch router (LSR) functionality with ECMP. The LSR receives a packet with a label and forwards it based on that label.
- FRRouting support for MPLS transit label switched paths (LSPs) and labeled routes (LER), both static routes and routes using BGP labeled-unicast (LU).
- FRR support for BGP/MPLS segment routing based on draft-ietf-idr-bgp-prefix-sid-06.

Example Configuration

Consider the following topology. Typically, host1 sends traffic to host2 through r1, r2 and r3. However, you can use segment routing to route traffic through a specific path. In the examples below, HTTP traffic is routed from host1 to host2 via r1, r4, r5 then r3. In addition, FTP traffic is routed via r5 without worrying what path it takes to get there.

For HTTP traffic to be routed from host1 to host2 via r1, r4, r5 then r3, the MPLS controller tells host1 to push label stack 103,105,104 on all HTTP traffic destined for host2; 104 is the outside label and 103 is the inside label. Switch r1 sees label 104, then pops that outermost label and forwards the payload towards switch r4. Switch r4 sees label 105, then pops that label and forwards the payload towards switch r5. Switch r5 sees label 103, then pops that label and forwards the payload towards switch r3. Switch r3 sees just an IP packet, and routes it as usual.
For FTP traffic to be routed from host1 to host2 through r5, the MPLS controller tells host1 to push label stack 105 on all FTP traffic destined for host2. Switch r1 sees label 105, then uses ECMP using swap with label 105 and forwards the payload towards switches r4 and r2. Switches r2 and r4 see label 105, then they pop the label and forward the payload towards switches r5 and r3. Switches r5 and r3 both see just an IP packet and route it as usual.

Switches r1 through r5 announce their loopbacks (the 10.1.1.* addresses above) in BGP with a label-index.

The table below contains the configuration for all five nodes.

<table>
<thead>
<tr>
<th>Node</th>
<th>/etc/network/interfaces</th>
<th>/etc/frr/frr.conf</th>
</tr>
</thead>
</table>
### /etc/network/interfaces

<table>
<thead>
<tr>
<th>auto lo</th>
<th>iface lo inet</th>
<th>loopback</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>address 10.1.1.2/32</td>
</tr>
<tr>
<td>auto swp1</td>
<td>iface swp1</td>
<td>mpls-enable yes</td>
</tr>
<tr>
<td>auto swp3</td>
<td>iface swp3</td>
<td>mpls-enable yes</td>
</tr>
<tr>
<td>auto swp5</td>
<td>iface swp5</td>
<td>mpls-enable yes</td>
</tr>
<tr>
<td>auto vagrant</td>
<td>iface vagrant</td>
<td>inet dhcp</td>
</tr>
<tr>
<td>auto eth0</td>
<td>iface eth0</td>
<td>inet dhcp vrf mgmt</td>
</tr>
<tr>
<td>auto mgmt</td>
<td>iface mgmt</td>
<td>address 127.0.0.1/8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vrf-table auto</td>
</tr>
</tbody>
</table>

### /etc/frr/frr.conf

<table>
<thead>
<tr>
<th>frr version 4.0+cl3u9</th>
</tr>
</thead>
<tbody>
<tr>
<td>frr defaults datacenter</td>
</tr>
<tr>
<td>hostname r2</td>
</tr>
<tr>
<td>username cumulus nopassword</td>
</tr>
<tr>
<td>service integrated-vtysh-config</td>
</tr>
<tr>
<td>log syslog informational</td>
</tr>
<tr>
<td>router bgp 65222</td>
</tr>
<tr>
<td>bgp router-id 10.1.1.2</td>
</tr>
<tr>
<td>no bgp default ipv4-unicast</td>
</tr>
<tr>
<td>neighbor EBGP peer-group</td>
</tr>
<tr>
<td>neighbor EBGP remote-as external</td>
</tr>
<tr>
<td>neighbor swp1 interface peer-group EBGP</td>
</tr>
<tr>
<td>neighbor swp3 interface peer-group EBGP</td>
</tr>
<tr>
<td>neighbor swp5 interface peer-group EBGP</td>
</tr>
<tr>
<td>address-family ipv4 unicast</td>
</tr>
<tr>
<td>network 10.1.1.1/32 label-index 1</td>
</tr>
<tr>
<td>network 10.1.1.2/32 label-index 2</td>
</tr>
<tr>
<td>network 10.1.1.3/32 label-index 3</td>
</tr>
<tr>
<td>network 10.1.1.4/32 label-index 4</td>
</tr>
<tr>
<td>network 10.1.1.5/32 label-index 5</td>
</tr>
<tr>
<td>exit-address-family</td>
</tr>
<tr>
<td>address-family ipv4 labeled-unicast</td>
</tr>
<tr>
<td>neighbor EBGP activate</td>
</tr>
<tr>
<td>exit-address-family</td>
</tr>
<tr>
<td>mpls label global-block 100 200</td>
</tr>
<tr>
<td>line vty</td>
</tr>
<tr>
<td>!</td>
</tr>
</tbody>
</table>
/etc/network/interfaces

auto lo
iface lo inet
loopback
  address
  10.1.1.3/32

auto swp2
iface swp2
  mpls-enable yes

auto swp5
iface swp5
  mpls-enable yes

auto swp10
iface swp10
  address
  192.168.22.1/24
  mpls-enable yes

auto vagrant
iface vagrant inet
dhcp

auto eth0
iface eth0 inet dhcp
  vrf mgmt

auto mgmt
iface mgmt
  address
  127.0.0.1/8
  vrf-table auto

/etc/frr/frr.conf

frr version 4.0+cl3u9
frr defaults datacenter
hostname r3
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
router bgp 65333
  bgp router-id 10.1.1.3
  no bgp default ipv4-unicast
  neighbor EBGP peer-group
  neighbor EBGP remote-as external
  neighbor swp2 interface peer-group EBGP
  neighbor swp5 interface peer-group EBGP
    address-family ipv4 unicast
      network 10.1.1.1/32
      label-index 1
      network 10.1.1.2/32
      label-index 2
      network 10.1.1.3/32
      label-index 3
      network 10.1.1.4/32
      label-index 4
      network 10.1.1.5/32
      label-index 5
      exit-address-family
    !
    address-family ipv4 labeled-unicast
      neighbor EBGP activate
      exit-address-family
    !
    mpls label global-block 100 200
! line vty
### /etc/network/interfaces

```plaintext
auto lo
iface lo inet loopback
    address 10.1.1.4/32
auto swp1
iface swp1 mpls-enable yes
auto swp5
iface swp5 mpls-enable yes
auto vagrant
iface vagrant inet dhcp
auto eth0
iface eth0 inet dhcp
    vrf mgmt
auto mgmt
iface mgmt
    address 127.0.0.1/8
    vrf-table auto
```
```
/etc/network/interfaces

auto lo
iface lo inet
loopback
  address
  10.1.1.5/32

auto swp2
iface swp2
  mpls-enable yes

auto swp5
iface swp5
  mpls-enable yes

auto swp10
iface swp10
  address
  192.168.22.1/24
  mpls-enable yes

auto vagrant
iface vagrant inet
dhcp

auto eth0
iface eth0 inet dhcp
  vrf mgmt

auto mgmt
iface mgmt
  address
  127.0.0.1/8
  vrf-table auto
```

```
/etc/frr/frr.conf

frr version 4.0+cl3u9
frr defaults datacenter
hostname r5
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
router bgp 65555
  bgp router-id 10.1.1.5
  no bgp default ipv4-unicast
  neighbor EBGP peer-group
  neighbor EBGP remote-as external
  neighbor swp2 interface
  peer-group EBGP
  neighbor swp3 interface
  peer-group EBGP
  neighbor swp5 interface
  peer-group EBGP
  !
  address-family ipv4 unicast
    network 10.1.1.1/32
    label-index 1
      network 10.1.1.2/32
    label-index 2
      network 10.1.1.3/32
    label-index 3
      network 10.1.1.4/32
    label-index 4
      network 10.1.1.5/32
    label-index 5
    exit-address-family
    !
    address-family ipv4 unlabeled-unicast
    neighbor EBGP activate
    exit-address-family
    !
    mpls label global-block 100 200
    !
    line vty
    !
```
Configure Segment Routing

To configure the segment routing example above, use the `label-index` option in NCLU. Configure the following on each node:

```
cumulus@switch:~$ net add bgp network 10.1.1.1/32 label-index 1
cumulus@switch:~$ net add bgp network 10.1.1.2/32 label-index 2
cumulus@switch:~$ net add bgp network 10.1.1.3/32 label-index 3
cumulus@switch:~$ net add bgp network 10.1.1.4/32 label-index 4
cumulus@switch:~$ net add bgp network 10.1.1.5/32 label-index 5
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

Then, for each switch in the topology, define the `global-block` of labels to use for segment routing in FRR. The default global-block is 16000-23999. The example configuration uses `global-block 100 200`. The `local label` is the MPLS label `global-block` plus the `label-index`.

```
cumulus@switch:~$ cat /etc/frr/frr.conf
router bgp 400
  bgp router-id 10.1.1.4
  no bgp default ipv4-unicast
  neighbor EBGP peer-group
  neighbor EBGP remote-as external
  neighbor swp1 interface peer-group EBGP
  neighbor swp2 interface peer-group EBGP
  !
  address-family ipv4 unicast
    network 10.1.1.4/32 label-index 4
  exit-address-family
  !
  address-family ipv4 labeled-unicast
    neighbor EBGP activate
  exit-address-family
  !
  mpls label global-block 100 200
```

View the Configuration

You can see the `label-index` when you show the BGP configuration on a router.

```
cumulus@r4:~$ net show configuration bgp
...
router bgp 400
  bgp router-id 10.1.1.4
  address-family ipv4 unicast
    network 10.1.1.4/32 label-index 4
```

Or from another node in the network:
You can see the FRR MPLS table in the output below, where r1 receives a packet with label 104. Its outbound label is 3, which appears as implicit-null below, so it pops then the payload is forwarded out of swp3, the interface to r4:

```
cumulus@r1:~$ net show mpls table
Inbound                                      Outbound
Label         Type              Nexthop     Label
--------  -------  -------------------  --------
102      BGP  fe80::202:ff:fe00:6         3
103      BGP  fe80::202:ff:fe00:6       103
104      BGP  fe80::202:ff:fe00:c         3
105      BGP  fe80::202:ff:fe00:c       105
106      BGP  fe80::202:ff:fe00:1         3
107      BGP  fe80::202:ff:fe00:6       107
cumulus@r1:~$
```

You can see the MPLS routing table that is installed in the kernel as well:

```
cumulus@r1:~$ net show mpls table 104
Local label: 104 (installed)
 type: BGP remote label: implicit-null distance: 150
  via fe80::202:ff:fe00:c dev swp3 (installed)
cumulus@r1:~$
```
cumulus@r1:~$ ip -f mpls route show
102 via inet6 fe80::202:ff:fe00:6 dev swp2 proto zebra
103 as to 103 via inet6 fe80::202:ff:fe00:6 dev swp2 proto zebra
104 via inet6 fe80::202:ff:fe00:c dev swp3 proto zebra
105 proto zebra
    nexthop as to 105 via inet6 fe80::202:ff:fe00:6 dev swp2
    nexthop as to 105 via inet6 fe80::202:ff:fe00:c dev swp3
106 via inet6 fe80::202:ff:fe00:1 dev swp1 proto zebra
107 as to 107 via inet6 fe80::202:ff:fe00:6 dev swp2 proto zebra
cumulus@r1:~$
cumulus@r1:~$ ip -f mpls route show 104
104 via inet6 fe80::202:ff:fe00:c dev swp3 proto zebra
cumulus@r1:~$