CCIE Service Provider Lab Workbook v4.0 (http://labs.ine.com/workbook/toc/service-provider-v4) » CCIE SP v4 Advanced Technology Labs - Layer 3 VPN

MPLS L3VPN Inter-AS Option C with iBGP + Label

« MPLS L3VPN Inter-AS Option C - Multihop VPNv4 EBGP Exchange (/workbook/view/service-provider-v4/task/mpls-l3vpn-inter-as-option-c-multihop-vpnv4-ebgp-exchange-Mjg2OQ%3D%3D) | MPLS L3VPN Carrier Supporting Carrier (/workbook/view/service-provider-v4/task/mpls-l3vpn-carrier-supporting-carrier-Mjg3MQ%3D%3D) »

Last updated: April 22, 2016

Note:

Initial Configuration & Diagrams: Load the initial configuration files for the section named **Inter AS L3VPN Option C**, which can be found in CCIE SPv4 Topology Diagrams & Initial Configurations (http://labs.ine.com/workbook/view/service-provider-v4/task/ccie-spv4-topology-diagrams-initial-configs). Refer to the **Inter AS L3VPN Option C Diagram** in order to complete this task.

Task

- Configure IGP routing and LDP in the first AS, which consists of R1, R2, R3, and R5 as follows:
  - Enable OSPF Area 0 on the links between R1 & R3, R1 & R5, R2 & R3, and R3 & R5.
  - Enable OSPF Area 0 on the Loopback0 interfaces of R1, R2, R3, & R5 as passive interfaces.
  - Enable LDP on the links between R1 & R3, R1 & R5, R2 & R3, and R3 & R5.
- Configure IGP routing and LDP in the second AS, which consists of R4, R6, XR1, and XR2 as follows:
  - Use the following IS-IS NET addressing:
    - R4 – 49.0001.0000.0000.0004.00
    - R6 – 49.0001.0000.0000.0006.00
    - XR1 – 49.0001.0000.0000.0019.00
    - XR1 – 49.0001.0000.0000.0020.00
  - Enable IS-IS Level 2 on the links between R4 & R6, R4 & XR1, R4 & XR2, and XR1 & XR2.
  - Advertise the Loopback0 interfaces of R4, R6, XR1, and XR2 into IS-IS Level 2 as passive interfaces.
  - Enable LDP on the links between R4 & R6, R4 & XR1, R4 & XR2, and XR1 & XR2.
- Configure the following VRFs on PE routers R2 and R6 follows:
  - VRF VPN_A:
    - Route Distinguisher: 101:201
    - Route Target Import: 101:201
    - Route Target Export: 101:201
    - Assign this VRF on the links in the 30.0.0.0 network on R2 and R6.
  - VRF VPN_B:
    - Route Distinguisher: 102:202
    - Route Target Import: 102:202
    - Route Target Export: 102:202
    - Assign this VRF on the links in the 40.0.0.0 network on R2 and R6.

- Configure RIPv2 routing for VRF VPN_A as follows:
  - Enable RIP between R9 & R2.
  - Enable RIP between R10 & R6.
  - Advertise the Loopback0 networks of R9 & R10 into RIP.
- Configure EIGRP routing for VRF VPN_B as follows:
  - Use EIGRP Autonomous System 1.
  - Enable EIGRP between R7 & R2.
  - Enable EIGRP between R8 & R6.
  - Advertise the Loopback0 networks of R7 & R8 into EIGRP.
- Configure IPv4 Labeled Unicast BGP peerings as follows:
  - R1, R2, R3, and R5 are in AS 100.
  - R4, R6, XR1, and XR2 are in AS 200.
  - R1 should be an IPv4 Unicast iBGP Route Reflector for R2, R3, and R5.
  - XR1 should be an IPv4 Unicast iBGP Route Reflector for R4, R6, and XR2.
  - R1 and XR1 should be IPv4 Unicast EBGP peers.
  - Advertise the Loopback0 networks of R2 and R5 into BGP on R1.
  - Advertise the Loopback0 networks of R6 and XR2 into BGP on XR1.
  - Include BGP MPLS Labels advertisements for these networks.
- Configure VPNv4 BGP peerings as follows:
  - R2 and R5 should peer in AS 100 using each other’s Loopback0 interfaces.
  - R5 should be a VPNv4 Route Reflector for R2.
  - R6 and XR2 should peer in AS 200 using each other’s Loopback0 interfaces.
  - XR2 should be a VPNv4 Route Reflector for R6.
  - R5 and XR2 should peer multihop EBGP with each other’s Loopback0 interfaces.
  - Do not change the next-hop value of VPNv4 routes advertised from R5 to XR2 and vice versa.
- Redistribute between VPNv4 BGP and the VRF aware IGP processes on R2 and R6.
- Once complete the following reachability should be achieved:
  - Customer routers R9 and R10 should have full IP reachability to each other’s networks.
  - Customer routers R7 and R8 should have full IP reachability to each other’s networks.
  - Traceroutes between these networks should indicate that a single end-to-end Label Switch Path is used.
  - Traceroutes should also indicate that traffic between VPN_A and VPN_B sites does not transit through the VPNv4 Route Reflectors R5 and XR2.

Configuration

Click to *collapse*

```
R1:
interface Loopback0
 ip ospf 1 area 0
!
interface GigabitEthernet1.119
 mpls bgp forwarding
!
router ospf 1
 passive-interface Loopback0
 network 10.0.0.0 0.255.255.255 area 0
 mpls ldp autoconfig area 0
!
router bgp 100
 template peer-policy iBGP_LABEL_POLICY
  route-reflector-client
  next-hop-self
  send-label
 exit-peer-policy
 !
 template peer-session iBGP_LABEL_SESSION
  remote-as 100
  update-source Loopback0
 exit-peer-session
 !
 bgp log-neighbor-changes
 network 2.2.2.2 mask 255.255.255.255
 network 5.5.5.5 mask 255.255.255.255
 neighbor 2.2.2.2 inherit peer-session iBGP_LABEL_SESSION
 neighbor 2.2.2.2 inherit peer-policy iBGP_LABEL_POLICY
 neighbor 3.3.3.3 inherit peer-session iBGP_LABEL_SESSION
 neighbor 3.3.3.3 inherit peer-policy iBGP_LABEL_POLICY
 neighbor 5.5.5.5 inherit peer-session iBGP_LABEL_SESSION
 neighbor 5.5.5.5 inherit peer-policy iBGP_LABEL_POLICY
 neighbor 12.1.19.19 remote-as 200
 neighbor 12.1.19.19 send-label
!
mpls ldp router-id Loopback0


R2:
vrf definition VPN_A
 rd 101:201
 !
 address-family ipv4
 route-target export 101:201
 route-target import 101:201
 exit-address-family
!
vrf definition VPN_B
 rd 102:202
 !
 address-family ipv4
 route-target export 102:202
 route-target import 102:202
```

```
 exit-address-family
!
interface Loopback0
 ip ospf 1 area 0
!
interface GigabitEthernet1.27
 vrf forwarding VPN_B
 ip address 40.2.7.2 255.255.255.0
!
interface GigabitEthernet1.29
 vrf forwarding VPN_A
 ip address 30.2.9.2 255.255.255.0
!
router eigrp 65535
 !
address-family ipv4 vrf VPN_B
 redistribute bgp 100
 network 40.0.0.0
 autonomous-system 1
 exit-address-family
!
router ospf 1
 passive-interface Loopback0
 network 10.0.0.0 0.255.255.255 area 0
 mpls ldp autoconfig area 0
!
router rip
 !
 address-family ipv4 vrf VPN_A
  redistribute bgp 100 metric 1
  network 30.0.0.0
  no auto-summary
  version 2
 exit-address-family
!
router bgp 100
 no bgp default ipv4-unicast
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 5.5.5.5 remote-as 100
 neighbor 5.5.5.5 update-source Loopback0
!
address-family ipv4
 neighbor 1.1.1.1 activate
 neighbor 1.1.1.1 send-label
exit-address-family
!
address-family vpnv4
 neighbor 5.5.5.5 activate
 neighbor 5.5.5.5 send-community extended
exit-address-family
!
address-family ipv4 vrf VPN_A
 redistribute rip
```

```
exit-address-family
!
address-family ipv4 vrf VPN_B
 redistribute eigrp 1
exit-address-family
!
mpls ldp router-id Loopback0

R3:
interface Loopback0
 ip ospf 1 area 0
!
router ospf 1
 passive-interface Loopback0
 network 10.0.0.0 0.255.255.255 area 0
 mpls ldp autoconfig area 0
!
router bgp 100
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback0
!
address-family ipv4
 neighbor 1.1.1.1 activate
 neighbor 1.1.1.1 send-label
 exit-address-family
!
mpls ldp router-id Loopback0


R4:
interface GigabitEthernet1.46
 ip router isis
!
interface GigabitEthernet1.419
 ip router isis
!
interface GigabitEthernet1.420
 ip router isis
!
router isis
 net 49.0001.0000.0000.0004.00
 is-type level-2-only
 passive-interface Loopback0
 mpls ldp autoconfig
!
router bgp 200
 neighbor 19.19.19.19 remote-as 200
 neighbor 19.19.19.19 update-source Loopback0
 !
 address-family ipv4
  neighbor 19.19.19.19 activate
  neighbor 19.19.19.19 send-label
 exit-address-family
 !
```

```
 mpls ldp router-id Loopback0


R5:
interface Loopback0
 ip ospf 1 area 0
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 mpls ldp autoconfig area 0
!
router bgp 100
 no bgp default ipv4-unicast
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 20.20.20.20 remote-as 200
 neighbor 20.20.20.20 ebgp-multihop 255
 neighbor 20.20.20.20 update-source Loopback0
 !
 address-family ipv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-label
 exit-address-family
 !
 address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
  neighbor 2.2.2.2 route-reflector-client
  neighbor 20.20.20.20 activate
  neighbor 20.20.20.20 send-community extended
  neighbor 20.20.20.20 next-hop-unchanged
 exit-address-family
!
mpls ldp router-id Loopback0


R6:
vrf definition VPN_A
 rd 101:201
 !
 address-family ipv4
 route-target export 101:201
 route-target import 101:201
 exit-address-family
!
vrf definition VPN_B
 rd 102:202
 !
 address-family ipv4
 route-target export 102:202
 route-target import 102:202
 exit-address-family
!
interface GigabitEthernet1.46
```

```
 ip router isis
!
interface GigabitEthernet1.68
 vrf forwarding VPN_B
 ip address 40.6.8.6 255.255.255.0
!
interface GigabitEthernet1.610
 vrf forwarding VPN_A
 ip address 30.6.10.6 255.255.255.0
!
router eigrp 65535
 !
 address-family ipv4 vrf VPN_B
  redistribute bgp 200
  network 40.0.0.0
  autonomous-system 1
 exit-address-family
!
router isis
 net 49.0001.0000.0000.0006.00
 is-type level-2-only
 passive-interface Loopback0
 mpls ldp autoconfig
!
router rip
 !
 address-family ipv4 vrf VPN_A
  redistribute bgp 200 metric 1
  network 30.0.0.0
  no auto-summary
  version 2
 exit-address-family
!
router bgp 200
 no bgp default ipv4-unicast
 neighbor 19.19.19.19 remote-as 200
 neighbor 19.19.19.19 update-source Loopback0
 neighbor 20.20.20.20 remote-as 200
 neighbor 20.20.20.20 update-source Loopback0
 !
 address-family ipv4
  neighbor 19.19.19.19 activate
  neighbor 19.19.19.19 send-label
 exit-address-family
 !
 address-family vpnv4
  neighbor 20.20.20.20 activate
  neighbor 20.20.20.20 send-community
 exit-address-family
 !
 address-family ipv4 vrf VPN_A
  redistribute rip
 exit-address-family
 !
```

```
 address-family ipv4 vrf VPN_B
  redistribute eigrp 1
 exit-address-family
!
mpls ldp router-id Loopback0

R7:
router eigrp 1
 network 0.0.0.0
 no auto-summary

R8:
router eigrp 1
 network 0.0.0.0
 no auto-summary

R9:
router rip
 version 2
 network 9.0.0.0
 network 30.0.0.0
 no auto-summary

R10:
router rip
 version 2
 network 10.0.0.0
 network 30.0.0.0
 no auto-summary

XR1:
route-policy PASS
  pass
end-policy
!
router static
 address-family ipv4 unicast
  12.1.19.1/32 GigabitEthernet0/0/0/0.119
 !
!
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0019.00
 address-family ipv4 unicast
 mpls ldp auto-config
!
interface Loopback0
 passive
 address-family ipv4 unicast
 !
!
interface GigabitEthernet0/0/0/0.419
 address-family ipv4 unicast
 !
```

CONTENTS

```
!
interface GigabitEthernet0/0/0/0.1920
  address-family ipv4 unicast
  !
 !
!
router bgp 200
 address-family ipv4 unicast
  network 6.6.6.6/32
  network 20.20.20.20/32
  allocate-label all
 !
 neighbor-group iBGP_LABEL_GROUP
  remote-as 200
  update-source Loopback0
  address-family ipv4 labeled-unicast
   route-reflector-client
   next-hop-self
  !
 !
 neighbor 4.4.4.4
  use neighbor-group iBGP_LABEL_GROUP
 !
 neighbor 6.6.6.6
  use neighbor-group iBGP_LABEL_GROUP
 !
 neighbor 12.1.19.1
  remote-as 100
  address-family ipv4 labeled-unicast
   route-policy PASS in
   route-policy PASS out
  !
 !
 neighbor 20.20.20.20
  use neighbor-group iBGP_LABEL_GROUP
  address-family ipv4 unicast
  !
 !
!
mpls ldp
 router-id 19.19.19.19

XR2:
route-policy PASS
  pass
end-policy
!
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0020.00
 address-family ipv4 unicast
  mpls ldp auto-config
 !
 interface Loopback0
```

CONTENTS

```
  passive
  address-family ipv4 unicast
 !
!
interface GigabitEthernet0/0/0/0.420
 address-family ipv4 unicast
 !
!
interface GigabitEthernet0/0/0/0.1920
  address-family ipv4 unicast
  !
 !
!
router bgp 200
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 5.5.5.5
  remote-as 100
  ebgp-multihop 255
  update-source Loopback0
  address-family vpnv4 unicast
   next-hop-unchanged
   route-policy PASS in
   route-policy PASS out
 !
!
neighbor 6.6.6.6
 remote-as 200
 update-source Loopback0
 address-family vpnv4 unicast
  route-reflector-client
 !
!
neighbor 19.19.19.19
  remote-as 200
  update-source Loopback0
  address-family ipv4 labeled-unicast
  !
 !
!
mpls ldp
 router-id 20.20.20.20
```

Verification

This example is similar to the previous MPLS L3VPN Inter-AS Option C design, with the exception that now BGP + Label is used everywhere to build Transport Labels between the Autonomous Systems as opposed to redistributing BGP into IGP and using LDP derived labels. This design is also used to help dispel a common misconception about where you do or do not need to add the **send-label** command in regular IPv4 Unicast BGP. The key question to always ask yourself is “how is the next-hop of the

VPNv4 route being learned?” If the next-hop is being learned via IGP, you need to use an IGP based label (e.g. LDP or MPLS TE) to reach it, but if the next-hop is being learned via BGP, you need to use a BGP based label.

In this case all routers in AS 100 are peering IPv4 Unicast BGP with R1, and all routers in AS 200 are peering with XR1. R1 and XR1 are originating the Loopback0 networks of R2/R5 and R6/XR2 into BGP respectively. No redistribution of regular IPv4 Unicast BGP is occurring, so all devices should be learning these routes through regular BGP, as seen below.

CONTENTS

```
R1#show ip route bgp | begin ^Gateway
Gateway of last resort is not set

    6.0.0.0/32 is subnetted, 1 subnets
B       6.6.6.6 [20/20] via 12.1.19.19, 00:33:33
    20.0.0.0/32 is subnetted, 1 subnets
B       20.20.20.20 [20/10] via 12.1.19.19, 00:33:33

R2#show ip route bgp | begin ^Gateway
Gateway of last resort is not set

    6.0.0.0/32 is subnetted, 1 subnets
B       6.6.6.6 [200/20] via 1.1.1.1, 00:32:17
    20.0.0.0/32 is subnetted, 1 subnets
B       20.20.20.20 [200/10] via 1.1.1.1, 00:32:17

R3#show ip route bgp | begin ^Gateway
Gateway of last resort is not set

    6.0.0.0/32 is subnetted, 1 subnets
B       6.6.6.6 [200/20] via 1.1.1.1, 00:32:45
    20.0.0.0/32 is subnetted, 1 subnets
B       20.20.20.20 [200/10] via 1.1.1.1, 00:32:45

R4#show ip route bgp | begin ^Gateway
Gateway of last resort is not set

    2.0.0.0/32 is subnetted, 1 subnets
B       2.2.2.2 [200/3] via 19.19.19.19, 00:28:19
    5.0.0.0/32 is subnetted, 1 subnets
B       5.5.5.5 [200/2] via 19.19.19.19, 00:28:19
R4#


R5#show ip route bgp | begin ^Gateway
Gateway of last resort is not set

    6.0.0.0/32 is subnetted, 1 subnets
B       6.6.6.6 [200/20] via 1.1.1.1, 00:32:17
    20.0.0.0/32 is subnetted, 1 subnets
B       20.20.20.20 [200/10] via 1.1.1.1, 00:32:17

R6#show ip route bgp | begin ^Gateway
Gateway of last resort is not set

    2.0.0.0/32 is subnetted, 1 subnets
B       2.2.2.2 [200/3] via 19.19.19.19, 00:28:19
    5.0.0.0/32 is subnetted, 1 subnets
B       5.5.5.5 [200/2] via 19.19.19.19, 00:28:19

RP/0/0/CPU0:XR1#show route bgp
Thu May 14 22:55:58.882 UTC
```

```
B   2.2.2.2/32 [20/3] via 12.1.19.1, 00:34:13

B   5.5.5.5/32 [20/2] via 12.1.19.1, 00:34:13


RP/0/3/CPU0:XR2#show route bgp

Thu May 14 22:58:58.900 UTC


B   2.2.2.2/32 [200/3] via 19.19.19.19, 00:29:05

B   5.5.5.5/32 [200/2] via 19.19.19.19, 00:29:05
```

Note that in the routing tables only the BGP routes from the remote AS are installed, and not the BGP routes from the local AS. This is because the Loopbacks of R2, R5, R6, and XR2 are advertised into both IGP and iBGP, and IGP has a lower administrative distance than iBGP. The routes will still be in the BGP table, but won't be installed in the routing table. This is what the RIB Failure indicates in the **show ip bgp** output.

```
R3#show bgp ipv4 unicast

BGP table version is 9, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,

              x best-external, a additional-path, c RIB-compressed,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found


 Network          Next Hop            Metric LocPrf Weight Path

r>i2.2.2.2/32       1.1.1.1         3   100         0   i

r>i5.5.5.5/32       1.1.1.1         2   100         0   i

*>i6.6.6.6/32       1.1.1.1         20  100         0   200 i

*>i20.20.20.20/32   1.1.1.1         10  100         0   200 i
```

Now that we know at least that the routes to the remote AS are being learned via BGP, let's look at how this affects the Label Switch Path of traffic going between the VPN_A and VPN_B customer sites.

```
R9#traceroute 10.10.10.10 source loopback 0

Type escape sequence to abort.

Tracing the route to 10.10.10.10

VRF info: (vrf in name/id, vrf out name/id)

  1 30.2.9.2 5 msec 1 msec 1 msec

  2 10.2.3.3 [MPLS: Labels 16/22/19 Exp 0] 13 msec 11 msec 127 msec

  3 10.1.3.1 [MPLS: Labels 22/19 Exp 0] 32 msec 29 msec 34 msec

  4 12.1.19.19 [MPLS: Labels 16008/19 Exp 0] 34 msec 28 msec 30 msec

  5 20.4.19.4 [MPLS: Labels 16/19 Exp 0] 32 msec 29 msec 35 msec

  6 30.6.10.6 [MPLS: Label 19 Exp 0] 13 msec 16 msec 17 msec

  7 30.6.10.10 19 msec *  15 msec
```

R9's traceroute from the local VPN_A site to the remote VPN_A site on R10 indicates that the traffic goes from R9 > R2 > R3 > R1 > XR1 > R4 > R6 > R10. This is as expected, because like in the last example, the VPNv4 Route Reflectors (R5 and XR2) are not updating the VPNv4 next-hop value. This means that although they are in the control plane for the VPNv4 route advertisement, they are not actually in the data plane.

What is different about the above traceroute in this example vs. the last one though is that when R2 sends traffic to the first labeled hop of R3, a three label stack of 16/22/19 is used instead of a normal two label stack commonly seen in L3VPN. To see why this is happening we need to look at the VPNv4 route recursion process in more detail for this destination.

```
R2#show bgp vpnv4 unicast vrf VPN_A 10.10.10.10/32

BGP routing table entry for 101:201:10.10.10.10/32, version 18

Paths: (1 available, best #1, table VPN_A)

  Not advertised to any peer

  Refresh Epoch 1

  200

     6.6.6.6 (metric 3) (via default) from 5.5.5.5 (5.5.5.5)

      Origin incomplete, metric 0, localpref 100, valid, internal, best

      Extended Community: RT:101:201

       mpls labels in/out nolabel/19

      rx pathid: 0, tx pathid: 0x0
```

The first step is to look at the VPNv4 route itself. Like in the last exampl,e this route is being learned from the VPNv4 Route Reflector R5, but the next-hop value points at 6.6.6.6 (R6's Loopback0). We already know the VPN Label will be 19 from this output, but to find a transport label for 6.6.6.6 we next need to look in the global routing table.

```
R2#show ip route 6.6.6.6

Routing entry for 6.6.6.6/32

   Known via "bgp 100" , distance 200, metric 20

  Tag 200, type internal

  Last update from 1.1.1.1 00:13:12 ago

  Routing Descriptor Blocks:

  * 1.1.1.1 , from 1.1.1.1, 00:13:12 ago

      Route metric is 20, traffic share count is 1

      AS Hops 1

      Route tag 200

       MPLS label: 22
```

R2 sees the next-hop value 6.6.6.6 learned via iBGP from R1. In the previous example this next-hop was learned from OSPF due to the BGP to IGP redistribution on the Inter-AS edge routers. Since this is a BGP learned route, we have to use a BGP derived label in the LSP. This is seen in the output above as label number 22. Again note that this is a *transport label* and not a *VPN label*. Even though we have found a BGP derived label for the next-hop, the route recursion process is not complete, because we haven't found the outgoing interface. We now need to do a lookup on the next-hop of 1.1.1.1 until route recursion eventually points at a physical interface.

```
R2#show ip route 1.1.1.1

Routing entry for 1.1.1.1/32

   Known via "ospf 1" , distance 110, metric 3, type intra area

  Last update from 10.2.3.3 on GigabitEthernet1.23, 00:32:23 ago

  Routing Descriptor Blocks:

  * 10.2.3.3, from 1.1.1.1, 00:32:23 ago, via GigabitEthernet1.23

      Route metric is 3, traffic share count is 1
```

This was the last step needed in this particular recursion lookup, as the next-hop 1.1.1.1 is learned from the IGP peer 10.2.3.3 on the connected link GigabitEthernet1.23. Due to this two step recursion process to find the transport label, R2 will encapsulate traffic with two transport labels instead of one. The topmost label will be for the last recursion of 1.1.1.1 towards R3 (LDP label), the next label will be for the BGP route of 6.6.6.6 towards 1.1.1.1, and the bottom VPN label will be for the final destination. The output below neatly shows the three steps taken to build the label stack.

```
R2#show ip cef vrf VPN_A 10.10.10.10/32 detail
10.10.10.10/32, epoch 0, flags [rib defined all labels]
  recursive via 6.6.6.6 label 19
    recursive via 1.1.1.1 label 22
      nexthop 10.2.3.3 GigabitEthernet1.23 label 16
```

CONTENTS

- Label 16: LDP Label for 1.1.1.1, the next-hop towards 6.6.6.6.
- Label 22: BGP Label for 6.6.6.6, the next-hop towards VPN_A route 10.10.10.10/32
- Label 19: VPN Label for 10.10.10.10/32.

Further verification of the control plane can be done as follows:

```
R2#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  3.3.3.3/32       0             Gi1.23     10.2.3.3
17         16         1.1.1.1/32       0             Gi1.23     10.2.3.3
18         Pop Label  10.3.5.0/24      0             Gi1.23     10.2.3.3
19         17         10.1.5.0/24      0             Gi1.23     10.2.3.3
20         Pop Label  10.1.3.0/24      0             Gi1.23     10.2.3.3
21         No Label   30.2.9.0/24[V]   0             aggregate/VPN_A
22         No Label   40.2.7.0/24[V]   0             aggregate/VPN_B
23         19         5.5.5.5/32       0             Gi1.23     10.2.3.3
24         No Label   7.7.7.7/32[V]    0             Gi1.27     40.2.7.7
25         No Label   9.9.9.9/32[V]    2998          Gi1.29     30.2.9.9

R2#show ip bgp labels
   Network          Next Hop      In label/Out label
   2.2.2.2/32       1.1.1.1         nolabel/19
   5.5.5.5/32       1.1.1.1         nolabel/20
   6.6.6.6/32       1.1.1.1         nolabel/22
   20.20.20.20/32   1.1.1.1         nolabel/23

R2#show bgp vpnv4 unicast all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 101:201 (VPN_A)
   9.9.9.9/32       30.2.9.9        25/nolabel
   10.10.10.10/32   6.6.6.6         nolabel/19
   30.2.9.0/24      0.0.0.0         21/nolabel(VPN_A)
   30.6.10.0/24     6.6.6.6         nolabel/20
Route Distinguisher: 102:202 (VPN_B)
   7.7.7.7/32       40.2.7.7        24/nolabel
   8.8.8.8/32       6.6.6.6         nolabel/21
   40.2.7.0/24      0.0.0.0         22/nolabel(VPN_B)
   40.6.8.0/24      6.6.6.6         nolabel/22
```

The depth of the label stack increases as the recursion depth increases. We can add an additional label in the stack by introducing an additional step in the recursion process:

CONTENTS

```
R1#conf t

Enter configuration commands, one per line.  End with CNTL/Z.

R1(config)#router bgp 100

R1(config-router)# template peer-policy iBGP_LABEL_POLICY

R1(config-router-ptmp)# no next-hop-self

R1(config-router-ptmp)#exit

R1(config-router)#network 12.1.19.19 mask 255.255.255.255

R1(config-router)#end

R1#
```

R1 is not setting itself as the next-hop for the BGP + Label route of 6.6.6.6, as it was previously. Instead, the next-hop is left unchanged - XR1's G0/0/0/0.119 12.1.19.19. To provide labeled reachability towards 12.1.19.19, R1 advertises the /32 connected host route into BGP that was auto-generated by the 'mpls bgp forwarding' command. This is how the additional level of recursion is introduced.

```
R2#show ip route 6.6.6.6
Routing entry for 6.6.6.6/32
  Known via "bgp 100", distance 200, metric 20
  Tag 200, type internal
  Last update from 12.1.19.19 00:23:08 ago
  Routing Descriptor Blocks:
  * 12.1.19.19, from 1.1.1.1, 00:23:08 ago
      Route metric is 20, traffic share count is 1
      AS Hops 1
      Route tag 200
      MPLS label: 16008

R2#show ip route 12.1.19.19
Routing entry for 12.1.19.19/32
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 1.1.1.1 00:23:15 ago
  Routing Descriptor Blocks:
  * 1.1.1.1, from 1.1.1.1, 00:23:15 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: 21

R2#show ip route 1.1.1.1
Routing entry for 1.1.1.1/32
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.2.3.3 on GigabitEthernet1.23, 01:20:18 ago
  Routing Descriptor Blocks:
  * 10.2.3.3, from 1.1.1.1, 01:20:18 ago, via GigabitEthernet1.23
      Route metric is 3, traffic share count is 1

R2#show ip cef vrf VPN_A 10.10.10.10/32 detail
10.10.10.10/32, epoch 0, flags [rib defined all labels]
  recursive via 6.6.6.6 label 19
    recursive via 12.1.19.19 label 16008
      recursive via 1.1.1.1 label 21
        nexthop 10.2.3.3 GigabitEthernet1.23 label 16

R9#traceroute 10.10.10.10 source loopback 0
Type escape sequence to abort.
Tracing the route to 10.10.10.10
VRF info: (vrf in name/id, vrf out name/id)
  1 30.2.9.2 4 msec 1 msec 1 msec
  2 10.2.3.3 [MPLS: Labels 16/21/16008/19 Exp 0] 15 msec 9 msec 12 msec
  3 10.1.3.1 [MPLS: Labels 21/16008/19 Exp 0] 24 msec 32 msec 31 msec
  4 12.1.19.19 [MPLS: Labels 16008/19 Exp 0] 32 msec 32 msec 32 msec
  5 20.4.19.4 [MPLS: Labels 16/19 Exp 0] 32 msec 30 msec 32 msec
  6 30.6.10.6 [MPLS: Label 19 Exp 0] 20 msec 20 msec 20 msec
  7 30.6.10.10 20 msec *  14 msec
```

- Label 16: LDP Label for 1.1.1.1, the next-hop towards 12.1.19.19.
- Label 21: BGP Label for 12.1.19.19, the next-hop towards 6.6.6.6
- Label 16008: BGP Label for 6.6.6.6, the next-hop towards VPN_A route 10.10.10.10/32
- Label 19: VPN Label for 10.10.10.10/32.

It could then be argued that this design is less efficient from a data plane point of view, since there are one or two extra labels of overhead in the traffic forwarding. Realistically though this small additional overhead should be negligible in the SP network.

« MPLS L3VPN Inter-AS Option C - Multihop VPNv4 EBGP Exchange (/workbook/view/service-provider-v4/task/mpls-l3vpn-inter-as-option-c-multihop-vpnv4-ebgp-exchange-Mjg2OQ%3D%3D) | MPLS L3VPN Carrier Supporting Carrier (/workbook/view/service-provider-v4/task/mpls-l3vpn-carrier-supporting-carrier-Mjg3MQ%3D%3D) »