MPLS L3VPN Inter-AS Option C - Multihop VPNv4 EBGP Exchange

« MPLS L3VPN Inter-AS Option B - VPNv4 EBGP Exchange (/workbook/view/service-provider-v4/task/mpls-l3vpn-inter-as-option-b-vpnv4-ebgp-exchange-Mjg2OA%3D%3D) | MPLS L3VPN Inter-AS Option C with iBGP + Label (/workbook/view/service-provider-v4/task/mpls-l3vpn-inter-as-option-c-with-ibgp-label-Mjg3MA%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.

CONTENTS ›

- Advertise the Loopback0 networks of R7 & R8 into RIP.

- Configure IPv4 Labeled Unicast BGP peerings as follows:
  - R1 is in AS 100, and XR1 is in AS 200.
  - 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 with all four of these networks.
- Redistribute IPv4 Unicast BGP into IGP as follows:
  - R1 should redistribute the Loopback0 networks of R6 & XR2 into OSPF that were learned from XR1.
  - XR1 should redistribute the Loopback0 networks of R2 & R5 into IS-IS that were learned from R1.
- 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***

CONTENTS ›

```
R1:
interface Loopback0
 ip ospf 1 area 0
!
interface GigabitEthernet1.119
 mpls bgp forwarding
!
router ospf 1
 redistribute bgp 100 subnets
 passive-interface Loopback0
 network 10.0.0.0 0.255.255.255 area 0
 mpls ldp autoconfig area 0
!
router bgp 100
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 12.1.19.19 remote-as 200
 !
 address-family ipv4
  network 3.3.3.3 mask 255.255.255.255
  network 5.5.5.5 mask 255.255.255.255
  neighbor 12.1.19.19 activate
  neighbor 12.1.19.19 send-label
 exit-address-family
!
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
```

CONTENTS

```
 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 5.5.5.5 remote-as 100
 neighbor 5.5.5.5 update-source Loopback0
 !
 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
!
```

```
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
!
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 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 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
 !
```

CONTENTS

```
 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 20.20.20.20 remote-as 200
 neighbor 20.20.20.20 update-source Loopback0
 !
 address-family vpnv4
  neighbor 20.20.20.20 activate
```

CONTENTS

```
  neighbor 20.20.20.20 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

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
  redistribute bgp 200
 mpls ldp auto-config
!
```

CONTENTS >

```
interface Loopback0
 passive
 address-family ipv4 unicast
 !
!
interface GigabitEthernet0/0/0/0.419
 address-family ipv4 unicast
 !
!
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 12.1.19.1
 remote-as 100
 address-family ipv4 labeled-unicast
   route-policy PASS in
   route-policy PASS out
  !
 !
!
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
 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 vpnv4 unicast
 !
 neighbor 5.5.5.5
  remote-as 100
  ebgp-multihop 255
 update-source Loopback0
 address-family vpnv4 unicast
  route-policy PASS in
  route-policy PASS out
  next-hop-unchanged
 !
!
neighbor 6.6.6.6
 remote-as 200
 update-source Loopback0
 address-family vpnv4 unicast
   route-reflector-client
  !
 !
!
mpls ldp
 router-id 20.20.20.20
```

Verification

Like previous L3VPN designs, the final verification is always the end-to-end reachability between the final customer sites. In this case customers in VPN_A and VPN_B have full reachability to their remote sites.

```
R9#show ip route rip
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 1 subnets
R        10.10.10.10 [120/1] via 30.2.9.2, 00:00:04, GigabitEthernet1.29
      30.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
R        30.6.10.0/24 [120/1] via 30.2.9.2, 00:00:04, GigabitEthernet1.29

R9#ping 10.10.10.10 source 9.9.9.9

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 2 seconds:
Packet sent with a source address of 9.9.9.9
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/11/15 ms

R7#show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      8.0.0.0/32 is subnetted, 1 subnets
D        8.8.8.8 [90/131072] via 40.2.7.2, 00:05:17, GigabitEthernet1.27
      40.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
D        40.6.8.0/24 [90/3072] via 40.2.7.2, 00:05:17, GigabitEthernet1.27

R7#ping 8.8.8.8 source 7.7.7.7

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
Packet sent with a source address of 7.7.7.7
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/15/39 ms
```

What is different about this design vs. previous ones though is that a single Label Switched Path (LSP) is used between the multiple Autonomous Systems that are doing Inter-AS exchange. This can be seen from the VPN Label number in the traceroute between the 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 4 msec 1 msec 1 msec

  2 10.2.3.3 [MPLS: Labels 20/ 21  Exp 0] 13 msec 12 msec 12 msec

  3 10.1.3.1 [MPLS: Labels 22/ 21  Exp 0] 20 msec 32 msec 32 msec

  4 12.1.19.19 [MPLS: Labels 16008/ 21  Exp 0] 32 msec 32 msec 31 msec

  5 20.4.19.4 [MPLS: Labels 16/ 21  Exp 0] 32 msec 28 msec 32 msec

  6 30.6.10.6 [MPLS: Label  21  Exp 0] 20 msec 20 msec 23 msec

  7 30.6.10.10 20 msec *  10 msec




R7#traceroute 8.8.8.8 source loopback 0
```

Type escape sequence to abort. Tracing the route to 8.8.8.8 VRF info: (vrf in name/id, vrf out name/id) 1 40.2.7.2 4 msec 1 msec 1 msec 2 10.2.3.3 [MPLS: Labels 20/**23** Exp 0] 10 msec 12 msec 12 msec 3 10.1.3.1 [MPLS: Labels 22/**23** Exp 0] 20 msec 32 msec 32 msec 4 12.1.19.19 [MPLS: Labels 16008/**23** Exp 0] 31 msec 31 msec 28 msec 5 20.4.19.4 [MPLS: Labels 16/**23** Exp 0] 32 msec 32 msec 32 msec 6 40.6.8.6 [MPLS: Label **23** Exp 0] 20 msec 20 msec 20 msec 7 40.6.8.8 23 msec * 11 msec

From the above output we can see that when traffic comes from the VPN_A site attached to R2 and transits to the VPN_A site attached to R6, which are in separate Autonomous Systems, the VPN Label remains the same end-to-end. For VPN_A specifically this VPN Label value is 21. For VPN_B the same occurs, maintaining VPN Label value 23.

The reason why the LSP remains the same end-to-end is that the VPNv4 route information is also maintained end-to-end, without changes in the next-hop value. Lets examine the route origination process for prefix 10.10.10.10/32 in VRF VPN_A. First R10 advertises this to its PE router via IGP, R6 then redistributes this PE-CE IGP learned route into VPNv4 BGP, which generates a local VPN Label value.

```
R6#show ip route vrf VPN_A 10.10.10.10


Routing Table: VPN_A
Routing entry for 10.10.10.10/32
  Known via "rip" , distance 120, metric 1
  Redistributing via bgp 200, rip
  Advertised by bgp 200
  Last update from 30.6.10.10 on GigabitEthernet1.610, 00:00:22 ago
  Routing Descriptor Blocks:
  * 30.6.10.10, from 30.6.10.10, 00:00:22 ago, via GigabitEthernet1.610
      Route metric is 1, traffic share count is 1


R6#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 2
Paths: (1 available, best #1, table VPN_A )
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local
    30.6.10.10 (via vrf VPN_A) from 0.0.0.0 (6.6.6.6)
      Origin incomplete, metric 1, localpref 100, weight 32768, valid, sourced, best
       Extended Community: RT:101:201
       mpls labels in/out 21/nolabel
      rx pathid: 0, tx pathid: 0x0
```

R6 now takes this VPNv4 route and advertises it to its VPNv4 BGP peer XR2. The next-hop value is set to R6's Loopback0 network 6.6.6.6, as this is the address that R6 and XR2 are peering with for the VPNv4 BGP session. Note that the VPN label has been set to 21 by R6, and this label will be maintained end-to-end.

```
RP/0/0/CPU0:XR2#show bgp vpnv4 unicast rd 101:201 10.10.10.10/32
Wed May 13 23:43:48.915 UTC
BGP routing table entry for 10.10.10.10/32, Route Distinguisher: 101:201
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  2           2
Last Modified: May 13 23:25:20.625 for 00:18:28
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    5.5.5.5
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    5.5.5.5
  Local, ( Received from a RR-client )
    6.6.6.6 (metric 20) from 6.6.6.6 (6.6.6.6)
      Received Label 21
      Origin incomplete, metric 1, localpref 100, valid, internal, best, group-best, import-candidate, not-in-vrf
      Received Path ID 0, Local Path ID 1, version 2
      Extended community: RT:101:201
```

Specifically the VPNv4 prefix is now 101:201:10.10.10.10/32 with a next-hop value of 6.6.6.6, a Route Target of 101:201, and a VPN Label of 21. XR2 is configured as a VPNv4 Route Reflector for R6, so there are no restrictions as to what type of peers this route can now be advertised to. In this case XR2 only has one other VPNv4 BGP peer, the multihop EBGP peering to R5. In a practical design the

VPNv4 Route Reflector would have other VPNv4 iBGP peers internal to the AS with other VRFs, but for this example, having just one VPNv4 iBGP RR client illustrates the same concept. Specifically this concept is that normally when a route is learned from an iBGP peer and then advertised to an EBGP peer, the next-hop value is updated to the local peering address.

For this example this would normally mean that XR2 would set the next-hop of 101:201:10.10.10.10/32 to 20.20.20.20, its own local Loopback0 interface, when the route is advertised to R5. Recall that in VPNv4 routing the next-hop value has an extra significance, because this is where the Label Switch Path terminates. This means that if XR2 were to update the next-hop value to to its Loopback0 as it advertises the VPN routes to R5, the LSP would change, and a new VPN Label would need to be generated. However in this case XR2 and R5 have the command **next-hop-unchanged** on their VPNv4 EBGP peering. The result of this can be seen below, as the next-hop value of the prefix is not changed when advertised between the EBGP peers.

```
RP/0/0/CPU0:XR2#show bgp vpnv4 unicast neighbors 5.5.5.5 advertised-routes
Wed May 13 23:47:50.428 UTC
Network            Next Hop        From            AS Path
 Route Distinguisher: 101:201
 10.10.10.10/32     6.6.6.6         6.6.6.6         200?
30.6.10.0/24       6.6.6.6         6.6.6.6         200?
Route Distinguisher: 102:202
8.8.8.8/32         6.6.6.6         6.6.6.6         200?
40.6.8.0/24        6.6.6.6         6.6.6.6         200?

Processed 4 prefixes, 4 paths

R5#show bgp vpnv4 unicast all 10.10.10.10/32
BGP routing table entry for 101:201:10.10.10.10/32, version 22
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  200
    6.6.6.6 (metric 1) (via default) from 20.20.20.20 (20.20.20.20)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:101:201
      mpls labels in/out nolabel/21
      rx pathid: 0, tx pathid: 0x0
```

When R5 receives this route, the next-hop value is still 6.6.6.6, R6's Loopback0. Note that the VPN Label value has not changed. Instead it remains as 21, which is what R6 originally allocated for this route. R5 now takes this prefix and advertises it to R2, its VPNv4 iBGP peer. By default the next-hop value is not changed when advertising a route from EBGP peers to iBGP peers (i.e. the **next-hop-self** command is not configured), thus R2 will receive the route with the same original next-hop value of 6.6.6.6.

```
R5#show bgp vpnv4 unicast all neighbors 2.2.2.2 advertised-routes

BGP table version is 25, local router ID is 5.5.5.5

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

Route Distinguisher: 101:201

 *>  10.10.10.10/32   6.6.6.6                                0 200 ?

 *>  30.6.10.0/24     6.6.6.6                                0 200 ?

Route Distinguisher: 102:202

 *>  8.8.8.8/32       6.6.6.6                                0 200 ?

 *>  40.6.8.0/24      6.6.6.6                                0 200 ?


Total number of prefixes 4


R2#show bgp vpnv4 unicast all 10.10.10.10/32

BGP routing table entry for 101:201:10.10.10.10/32, version 36

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

  Not advertised to any peer

  Refresh Epoch 1

  200

    6.6.6.6 (metric 1) (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/21

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

Now R2 has the route in the local VPNv4 BGP table for VRF VPN_A, and can redistribute it to the VRF aware IGP process of the PE-CE facing link. Nothing about this portion of the design changes as compared to previous L3VPN examples. As long as R2 has a VRF with a Route Target import policy of 101:201, the route can be imported to that VRF table. What is different however is that R2 must now form an end-to-end Label Switch Path to the final next-hop value of 6.6.6.6, which resides the other Service Provider's Autonomous System.

Recall that for MPLS L3VPN to work, two distinct label values must be correlated. The first is the VPN Label, which tells the final PE router which customer VRF table to do the routing lookup in. The VPN Label is originated by the VPNv4 BGP process, as routes are redistributed into the VRF's BGP process. The second is the Transport Label, which tells the core of the MPLS network which exit PE to label switch the traffic towards. The Transport Label is normally originated by IGP+LDP, but it could also be allocated via RSVP for MPLS TE or even BGP - as shown in this example between R1 and XR1. Regardless of how this label is allocated, the concept stays the same. The core of the network must have an end-to-end LSP for the Transport Label in order to get traffic from the ingress PE to the egress PE. This is where the second part of this design comes in, which is the BGP + Label exchange between the Inter-AS edge routers, or what is sometimes referred to as IPv4 Labeled Unicast BGP, described under RFC 3107 - Carrying Label Information in BGP-4 (https://tools.ietf.org/html/rfc3107).

We know that since BGP does not provide its own transport protocol, normally an IGP like OSPF or IS-IS provides transport between non-connected peers so they can establish their TCP peering. For example in AS 100 the iBGP peers R2 and R5 are not directly connected. This means that an IGP is needed in order to give them IP reachability between their Loopback0 interfaces before the TCP session can be established. The same concept is true for the Inter-AS multihop VPNv4 peering that is occurring between R5 and XR2.

R5 and XR2 need routes to each other's Loopback0 interfaces as a first step in establishing the peering these two routers. In this design, the route is learned through a normal IPv4 Unicast BGP peering between the Inter-AS edge routers R1 and XR1.

```
R1#show bgp ipv4 unicast
BGP table version is 11, local router ID is 1.1.1.1
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
 *>  2.2.2.2/32       10.1.3.3                 3         32768 i
 *>  5.5.5.5/32       10.1.5.5                 2         32768 i
 *>  6.6.6.6/32       12.1.19.19              20             0 200 i
 *>  20.20.20.20/32   12.1.19.19              10             0 200 i




RP/0/0/CPU0:XR1#show bgp ipv4 unicast
Thu May 14 00:04:26.280 UTC
BGP router identifier 19.19.19.19, local AS number 200
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0xe0000000   RD version: 16
BGP main routing table version 16
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 2.2.2.2/32         12.1.19.1                3             0 100 i
*> 5.5.5.5/32         12.1.19.1                2             0 100 i
*> 6.6.6.6/32         20.4.19.4               20         32768 i
*> 20.20.20.20/32     20.19.20.20             10         32768 i

Processed 4 prefixes, 4 paths
```

As seen in the above output R1 and XR1 are advertising the Loopback0 networks of R2/R5 and R6/XR2 respectively. This is just a normal IPv4 Unicast BGP design where the BGP routers learn their own internal routes via an IGP, and then inject them into the BGP topology with a **network** command under the BGP process. This is the same as how normal IPv4 Unicast BGP routing works for Internet transit. However in this design there is an additional requirement that there must be an end-to-end MPLS LSP between the Loopback0 networks of R2 and R6.

To provide this, R1 and XR1 advertise not only the IPv4 Unicast BGP routes to each other, but they also allocate and advertise MPLS Labels via IPv4 Unicast BGP. This is different from a VPNv4 BGP allocated label, because a VPNv4 BGP peer is allocating a *VPN Label*, while the IPv4 Unicast BGP peer is allocating a *Transport Label*. This is what the **send-label** command does in regular IOS and the **allocate-label-all** under the **address-family ipv4 labeled-unicast** of IOS XR. The specific label allocations can be verified as follows:

CONTENTS

```
R1#show ip bgp labels
  Network          Next Hop      In label/Out label
  2.2.2.2/32       10.1.3.3        20/nolabel
  5.5.5.5/32       10.1.5.5        19/nolabel
  6.6.6.6/32       12.1.19.19      nolabel/16008
  20.20.20.20/32   12.1.19.19      nolabel/16000

RP/0/0/CPU0:XR1#show bgp ipv4 unicast labels
Thu May 14 00:07:23.497 UTC
BGP router identifier 19.19.19.19, local AS number 200
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0xe0000000   RD version: 16
BGP main routing table version 16
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        Rcvd Label      Local Label
*> 2.2.2.2/32         12.1.19.1       20               16002
*> 5.5.5.5/32         12.1.19.1       19               16010
*> 6.6.6.6/32         20.4.19.4       nolabel         16008
*> 20.20.20.20/32     20.19.20.20     nolabel         16000

Processed 4 prefixes, 4 paths
```

The above output shows that R1 is creating label value 20 for its BGP prefix 2.2.2.2/32 and label value 19 for prefix 5.5.5.5/32. Likewise XR1 is allocating label value 16008 for prefix 6.6.6.6/32 and label value 16000 for prefix 20.20.20.20/32. This now means that the edge routers agree on a Label Switch Path to reach routes that would normally be internal to their own network. In other words, for Inter-AS Option C to work the Service Providers have to leak their *internal* routing information to each other via IPv4 Labeled Unicast BGP. These local labels that R1 and XR1 have "allocated" into BGP are the same local labels allocated by each router for the LDP process.

```
R1#show mpls ldp bindings 2.2.2.2 32

  lib entry: 2.2.2.2/32, rev 18

        local binding:  label: 20

        remote binding: lsr: 3.3.3.3:0, label: 19

        remote binding: lsr: 5.5.5.5:0, label: 20



R1#show mpls ldp bindings 5.5.5.5 32

  lib entry: 5.5.5.5/32, rev 16

        local binding:  label: 19

        remote binding: lsr: 3.3.3.3:0, label: 18

        remote binding: lsr: 5.5.5.5:0, label: imp-null



RP/0/0/CPU0:XR1#show mpls ldp bindings 20.20.20.20/32

Thu May 14 00:14:16.249 UTC

20.20.20.20/32, rev 26

        Local binding: label: 16000

        Remote bindings: (2 peers)

            Peer              Label

            -----------------   ---------

            4.4.4.4:0          21

            20.20.20.20:0      ImpNull



RP/0/0/CPU0:XR1#show mpls ldp bindings 6.6.6.6/32

Thu May 14 00:16:17.181 UTC

6.6.6.6/32, rev 21

        Local binding: label: 16008

        Remote bindings: (2 peers)

            Peer              Label

            -----------------   ---------

            4.4.4.4:0          16

            20.20.20.20:0      16011
```

The next step is forming the LSP within the Autonomous Systems. In this example, routers in the transit path such as R2, R3, R4, R5, R6, and XR2, are not running IPv4 Unicast BGP and thus will not receive the advertisements from R1 or XR1 respectively. One option to accomplish this to enable regular BGP on the routers internal to each AS and exchange the BGP + Label prefixes. However, a simpler option is to redistribute the BGP learned routes on R1 and XR1 into the IGP at Service Provider. Since the routers are already running IGP + LDP for Intra-AS reachability, we can simply redistribute the BGP + Label learned routes into IGP, and have LDP create a label for it. This allows all internal routers running IGP + LDP to label switch traffic towards the Inter-AS edge routers, then the Inter-AS edge routers (R1 and XR1 in this case) can use the BGP learned labels to label switch traffic on the connected Inter-AS link. When the traffic reaches the opposing Inter-AS Edge router, the LSP will continue towards the final destination (the terminating PE router). This LSP can be verified end-to-end as follows:

```
R2#show ip route 6.6.6.6
Routing entry for 6.6.6.6/32
  Known via "ospf 1", distance 110, metric 1
  Tag 200, type extern 2, forward metric 2
  Last update from 10.2.3.3 on GigabitEthernet1.23, 00:55:50 ago
  Routing Descriptor Blocks:
  * 10.2.3.3, from 1.1.1.1, 00:55:50 ago, via GigabitEthernet1.23
      Route metric is 1, traffic share count is 1
      Route tag 200

R2#show mpls forwarding-table 6.6.6.6
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
26         20         6.6.6.6/32       0             Gi1.23     10.2.3.3

R2#show ip cef 6.6.6.6 detail
6.6.6.6/32, epoch 2
  local label info: global/26
  1 RR source [no flags]
  nexthop 10.2.3.3 GigabitEthernet1.23 label 20
```

R2 learns the route to R6's Loopback – 6.6.6.6/32 – via External OSPF, and via LDP with a Transport Label value of 20, which was allocated by R3.

```
R3#show ip route 6.6.6.6
Routing entry for 6.6.6.6/32
  Known via "ospf 1", distance 110, metric 1
  Tag 200, type extern 2, forward metric 1
  Last update from 10.1.3.1 on GigabitEthernet1.13, 00:58:54 ago
  Routing Descriptor Blocks:
  * 10.1.3.1, from 1.1.1.1, 00:58:54 ago, via GigabitEthernet1.13
      Route metric is 1, traffic share count is 1
      Route tag 200

R3#show mpls forwarding-table 6.6.6.6
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
20         22         6.6.6.6/32       3180          Gi1.13     10.1.3.1

R3#show ip cef 6.6.6.6 detail
6.6.6.6/32, epoch 2
  local label info: global/20
  nexthop 10.1.3.1 GigabitEthernet1.13 label 22
```

R3 learns the route to R6's Loopback via External OSPF, and via LDP with a Transport Label value of 22, which was allocated by R1.

CONTENTS ›

```
R1#show ip route 6.6.6.6
Routing entry for 6.6.6.6/32
  Known via "bgp 100" , distance 20, metric 20
  Tag 200, type external
  Redistributing via ospf 1
  Advertised by ospf 1 subnets
  Last update from 12.1.19.19 01:01:17 ago
  Routing Descriptor Blocks:
  * 12.1.19.19 , from 12.1.19.19, 01:01:17 ago
      Route metric is 20, traffic share count is 1
      AS Hops 1
      Route tag 200
      MPLS label: 16008

R1#show ip bgp labels
    Network          Next Hop      In label/Out label
    2.2.2.2/32       10.1.3.3        20/nolabel
    5.5.5.5/32       10.1.5.5        19/nolabel
    6.6.6.6/32       12.1.19.19      nolabel/ 16008
    20.20.20.20/32   12.1.19.19      nolabel/16000

R1#show mpls forwarding-table 6.6.6.6
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
 22          16008      6.6.6.6/32       3984          Gi1.119    12.1.19.19

R1#show ip cef 6.6.6.6 detail
6.6.6.6/32, epoch 2, flags [rib defined all labels]
  local label info: global/22
  recursive via 12.1.19.19 label 16008
    attached to GigabitEthernet1.119
```

R1 learns the route to R6’s Loopback via External BGP. This means that if traffic towards the destination is going to be label switched it has to use a label that was derived from BGP. This is essentially where R1 ties the BGP allocated label from XR1 together with its locally allocated label via LDP. Traffic on the LSP towards 6.6.6.6 will use the incoming LDP label of 22 and the outgoing BGP label of 16008.

```
RP/0/0/CPU0:XR1#show ip route 6.6.6.6
Thu May 14 00:34:25.066 UTC

Routing entry for 6.6.6.6/32
  Known via "isis 1", distance 115, metric 20, type level-2
  Installed May 13 23:22:37.601 for 01:11:47
  Routing Descriptor Blocks
    20.4.19.4, from 6.6.6.6, via GigabitEthernet0/0/0/0.419
      Route metric is 20
  No advertising protos.

RP/0/0/CPU0:XR1#show bgp ipv4 unicast labels
Thu May 14 00:35:55.520 UTC
BGP router identifier 19.19.19.19, local AS number 200
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0xe0000000   RD version: 16
BGP main routing table version 16
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        Rcvd Label      Local Label
*> 2.2.2.2/32         12.1.19.1       20              16002
*> 5.5.5.5/32         12.1.19.1       19              16010
 *> 6.6.6.6/32         20.4.19.4       nolabel         16008
*> 20.20.20.20/32     20.19.20.20     nolabel         16000

Processed 4 prefixes, 4 paths

RP/0/0/CPU0:XR1#show mpls forwarding prefix 6.6.6.6/32
Thu May 14 00:34:40.865 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ -------------- ------------
 16008   16          6.6.6.6/32         Gi0/0/0/0.419 20.4.19.4       5148

RP/0/0/CPU0:XR1#show cef 6.6.6.6/32
Thu May 14 00:35:19.193 UTC
6.6.6.6/32, version 1084, internal 0x4004001 0x0 (ptr 0xa0edbdf4) [1], 0x0 (0xa0ea7d88), 0x228 (0xa13f6500)
 Updated May 13 23:23:14.779
 local adjacency 20.4.19.4
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 20.4.19.4, GigabitEthernet0/0/0/0.419 , 7 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0xa15d6384 0x0]
    next hop 20.4.19.4
    tx adjacency
      local label 16008      labels imposed {16}
```

XR1 learns the route to R6's Loopback via IS-IS. This means that if traffic towards the destination is going to be label switched it has to use a label that was derived from LDP. This is where XR1 ties the BGP allocated label back to the LDP label allocated from the IGP route. Traffic on the LSP towards 6.6.6.6 will use the incoming BGP label of 16008 and the outgoing LDP label of 16.

CONTENTS

The process then continues below on R4 who is the Penultimate (next to last) Hop for R6's Loopback0. This makes R4 pop the top label off the stack and forward the remaining payload towards R6.

```
R4#show ip route 6.6.6.6

Routing entry for 6.6.6.6/32

  Known via "isis" , distance 115, metric 10, type level-2

  Redistributing via isis

  Last update from 20.4.6.6 on GigabitEthernet1.46, 01:19:00 ago

  Routing Descriptor Blocks:

  * 20.4.6.6, from 6.6.6.6, 01:19:00 ago, via GigabitEthernet1.46

      Route metric is 10, traffic share count is 1


R4#show mpls forwarding-table 6.6.6.6

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop

Label      Label      or Tunnel Id     Switched      interface

 16         Pop Label   6.6.6.6/32       16161         Gi1.46     20.4.6.6


R4#sh ip cef 6.6.6.6/32 detail

6.6.6.6/32, epoch 2

  local label info: global/16

   nexthop 20.4.6.6 GigabitEthernet1.46
```

Another interesting point about this design is that even though the VPNv4 Route Reflectors are in the path of the control plane advertisements for the VPNv4 routes, they are not in the actual data forwarding plane. This can be seen from the traceroutes below, as neither R5 nor XR2's IP addresses appear as hops in the path.

```
R7#traceroute 8.8.8.8 source loopback 0

Type escape sequence to abort.

Tracing the route to 8.8.8.8

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

  1 40.2.7.2 4 msec 1 msec 1 msec

  2 10.2.3.3 [MPLS: Labels 20/23 Exp 0] 10 msec 12 msec 12 msec

  3 10.1.3.1 [MPLS: Labels 22/23 Exp 0] 20 msec 32 msec 32 msec

  4 12.1.19.19 [MPLS: Labels 16008/23 Exp 0] 31 msec 31 msec 28 msec

  5 20.4.19.4 [MPLS: Labels 16/23 Exp 0] 32 msec 32 msec 32 msec

  6 40.6.8.6 [MPLS: Label 23 Exp 0] 20 msec 20 msec 20 msec

  7 40.6.8.8 23 msec *  11 msec



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 20/21 Exp 0] 13 msec 12 msec 12 msec

  3 10.1.3.1 [MPLS: Labels 22/21 Exp 0] 20 msec 32 msec 32 msec

  4 12.1.19.19 [MPLS: Labels 16008/21 Exp 0] 32 msec 32 msec 31 msec

  5 20.4.19.4 [MPLS: Labels 16/21 Exp 0] 32 msec 28 msec 32 msec

  6 30.6.10.6 [MPLS: Label 21 Exp 0] 20 msec 20 msec 23 msec

  7 30.6.10.10 20 msec *  10 msec
```

CONTENTS ›

The route-reflectors are not in the path because the original next-hop values are maintained end-to-end, the Transport Labels used on the LSP between R2 and R6 are for each other's Loopback0 interfaces, instead of the Loopback0 interfaces of the Route Reflectors. If we were to change this design so that R5 and XR2 update the next-hop values in their VPNv4 advertisements, we would see that the Route Reflectors would insert themselves in the forwarding plane. This fact can be demonstrated as follows:

```
R5#config t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#router bgp 100
R5(config-router)#address-family vpnv4 unicast
R5(config-router-af)#no neighbor 20.20.20.20 next-hop-unchanged
R5(config-router-af)#end
R5#
%SYS-5-CONFIG_I: Configured from console by console
R5#clear bgp vpnv4 unicast * out
R5#clear bgp vpnv4 unicast * in
R5#

RP/0/0/CPU0:XR2#config t
Wed Mar 21 17:45:49.067 UTC
RP/0/0/CPU0:XR2(config)#router bgp 200
RP/0/0/CPU0:XR2(config-bgp)#neighbor 5.5.5.5
RP/0/0/CPU0:XR2(config-bgp-nbr)# address-family vpnv4 unicast
RP/0/0/CPU0:XR2(config-bgp-nbr-af)#no next-hop-unchanged
RP/0/0/CPU0:XR2(config-bgp-nbr-af)#commit
Thu May 14 00:43:12.671 UTC
RP/0/0/CPU0:May 14 00:43:12.891 : config[65710]: %MGBL-CONFIG-6-DB_COMMIT : Configuration committed by user 'admin'. Use 'show configuration
 commit changes 1000000122' to view the changes.
RP/0/0/CPU0:XR2(config-bgp-nbr-af)#end
RP/0/0/CPU0:May 14 00:43:17.400 : config[65710]: %MGBL-SYS-5-CONFIG_I : Configured from console by admin
RP/0/0/CPU0:XR2#clear bgp vpnv4 unicast * soft in
Thu May 14 00:43:22.350 UTC
RP/0/0/CPU0:XR2#clear bgp vpnv4 unicast * soft out
Thu May 14 00:43:24.390 UTC
```

Now R2 will see the next-hop value of the remote VPN_A and VPN_B sites' routes as XR2, and R6 will see the next-hop as R5.

```
R2#show bgp vpnv4 unicast all
BGP table version is 53, local router ID is 2.2.2.2
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
Route Distinguisher: 101:201 (default for vrf VPN_A)
*> 9.9.9.9/32       30.2.9.9                 1            32768  ?
 *>i10.10.10.10/32 20.20.20.20             0   100         0   200 ?
*> 30.2.9.0/24      0.0.0.0                  0            32768  ?
 *>i30.6.10.0/24   20.20.20.20             0   100         0   200 ?
Route Distinguisher: 102:202 (default for vrf VPN_B)
*> 7.7.7.7/32       40.2.7.7                 156160      32768  ?
 *>i8.8.8.8/32         20.20.20.20             0   100         0   200 ?
*> 40.2.7.0/24      0.0.0.0                  0           32768   ?
 *>i40.6.8.0/24        20.20.20.20             0   100         0   200 ?

R6#show bgp vpnv4 unicast all
BGP table version is 25, local router ID is 6.6.6.6
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
Route Distinguisher: 101:201 (default for vrf VPN_A)
 *>i9.9.9.9/32     5.5.5.5             100         0   100 ?
*> 10.10.10.10/32   30.6.10.10      1           32768   ?
 *>i30.2.9.0/24        5.5.5.5             100         0   100 ?
*> 30.6.10.0/24     0.0.0.0         0           32768   ?
Route Distinguisher: 102:202 (default for vrf VPN_B)
 *>i7.7.7.7/32     5.5.5.5             100         0   100 ?
*> 8.8.8.8/32       40.6.8.8    156160          32768   ?
 *>i40.2.7.0/24        5.5.5.5             100         0   100 ?
*> 40.6.8.0/24      0.0.0.0         0           32768   ?
```

This now means that R5 and XR2 will be in the data plane path for the traffic between the customer sites.

```
R7#traceroute 8.8.8.8 source loopback 0

Type escape sequence to abort.

Tracing the route to 8.8.8.8

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

  1 40.2.7.2 3 msec 1 msec 1 msec

  2 10.2.3.3 [MPLS: Labels 21/16007 Exp 0] 22 msec 20 msec 20 msec

  3 10.1.3.1 [MPLS: Labels 23/16007 Exp 0] 20 msec 20 msec 28 msec

  4 12.1.19.19 [MPLS: Labels 16000/16007 Exp 0] 31 msec 29 msec 32 msec

   5 20.19.20.20 [MPLS: Label 16007 Exp 0] 31 msec 32 msec 31 msec

  6 20.4.20.4 [MPLS: Labels 16/23 Exp 0] 28 msec 32 msec 36 msec

  7 40.6.8.6 [MPLS: Label 23 Exp 0] 16 msec 20 msec 20 msec

  8 40.6.8.8 20 msec *  18 msec


R8#traceroute 7.7.7.7 source loopback 0

Type escape sequence to abort.

Tracing the route to 7.7.7.7

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

  1 40.6.8.6 4 msec 1 msec 1 msec

  2 20.4.6.4 [MPLS: Labels 20/23 Exp 0] 20 msec 15 msec 15 msec

  3 20.4.19.19 [MPLS: Labels 16010/23 Exp 0] 16 msec 24 msec 32 msec

  4 12.1.19.1 [MPLS: Labels 19/23 Exp 0] 32 msec 40 msec 28 msec

   5 10.1.5.5 [MPLS: Label 23 Exp 0] 24 msec 32 msec 32 msec

  6 10.3.5.3 [MPLS: Labels 19/24 Exp 0] 31 msec 27 msec 31 msec

  7 40.2.7.2 [MPLS: Label 24 Exp 0] 21 msec 23 msec 20 msec

  8 40.2.7.7 20 msec *  14 msec
```

In a real world design this behavior is undesirable because not only is the route now suboptimal (e.g. R2 > R3 > R1 > XR1 > XR2 > R4 > R6 instead of R2 > R3 > R1 > XR1 > R4 > R6) but it means that the route reflectors who are potentially servicing hundreds or thousands of PEs need to not only maintain the routing control plane for all of them, but actually perform the data forwarding. By using the **next-hop-unchanged** command this removes the RR's need to be in the data plane when there a more optimal path that avoids the RR is available. Hardware resources are consumed on the route-reflectors as now they have to hold forwarding entries, as seen in the Option-B design.

This Inter-AS Layer 3 VPNs design is documented under Section 10.c in RFC 4364 (https://tools.ietf.org/html/rfc4364#section-10)

« MPLS L3VPN Inter-AS Option B - VPNv4 EBGP Exchange (/workbook/view/service-provider-v4/task/mpls-l3vpn-inter-as-option-b-vpnv4-ebgp-exchange-Mjg2OA%3D%3D) | MPLS L3VPN Inter-AS Option C with iBGP + Label (/workbook/view/service-provider-v4/task/mpls-l3vpn-inter-as-option-c-with-ibgp-label-Mjg3MA%3D%3D) »

CONTENTS