

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 6PE

« [MPLS L3VPN Carrier Supporting Carrier \(/workbook/view/service-provider-v4/task/mpls-l3vpn-carrier-supporting-carrier-Mjg3MQ%3D%3D\)](/workbook/view/service-provider-v4/task/mpls-l3vpn-carrier-supporting-carrier-Mjg3MQ%3D%3D) | [MPLS L3VPN 6VPE \(/workbook/view/service-provider-v4/task/mpls-l3vpn-6vpe-Mjg3Mw%3D%3D\)](/workbook/view/service-provider-v4/task/mpls-l3vpn-6vpe-Mjg3Mw%3D%3D) »

Last updated: April 22, 2016

>
CONTENTS

Note:

Initial Configuration & Diagrams: [Load the initial configuration files for the section named 6PE, 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 6PE / 6VPE Diagram in order to complete this task.](#)

Task

- The network is preconfigured as follows:
 - The Service Provider AS 100 consists of R2, R3, R4, R5, R6 and XR1.
 - AS 100 runs OSPF + LDP on all transit links, and is IPv4 only enabled.
 - Customer routers R1 & XR2 run both IPv4 and IPv6 to their PE routers R2 and XR1.
- Configure BGP on R1, R2, XR1, and XR2 as follows:
 - R1 should be in AS 1.
 - R2 and XR1 should be in AS 100.
 - XR2 should be in AS 20.
 - R1 and R2 should peer EBGP using their connected link for both IPv4 and IPv6 Unicast.
 - R2 and XR1 should peer iBGP using the Loopback0 interfaces, for both IPv4 and IPv6 Labeled Unicast, and use **next-hop-self**.
 - XR1 and XR2 should peer EBGP using their connected link for both IPv4 and IPv6 Unicast.
 - Advertise the prefix 1.1.1.1/32 and 2000::1:1:1:1/128 into BGP on R1.
 - Advertise the prefix 20.20.20.20/32 and 2000::20:20:20:20/128 into BGP on XR2.
- Once complete R1 and XR2 should be able to reach each other's IPv4 and IPv6 Loopback0 interfaces when sourcing traffic from their own Loopback0 interface.

Configuration [Click to collapse](#)

```
R1:
ipv6 unicast-routing
ipv6 cef distributed
!
router bgp 1
  no bgp default ipv4-unicast
  neighbor 10.1.2.2 remote-as 100
  neighbor 2001:10:1:2::2 remote-as 100
  !
  address-family ipv4
    network 1.1.1.1 mask 255.255.255.255
    neighbor 10.1.2.2 activate
  exit-address-family
  !
  address-family ipv6
    network 2001::1:1:1/128
    neighbor 2001:10:1:2::2 activate
  exit-address-family

R2:
ipv6 unicast-routing
ipv6 cef distributed
!
router bgp 100
  no bgp default ipv4-unicast
  neighbor 10.1.2.1 remote-as 1
  neighbor 19.19.19.19 remote-as 100
  neighbor 19.19.19.19 update-source Loopback0
  neighbor 2001:10:1:2::1 remote-as 1
  !
  address-family ipv4
    neighbor 10.1.2.1 activate
    neighbor 19.19.19.19 activate
    neighbor 19.19.19.19 next-hop-self
  exit-address-family
  !
  address-family ipv6
    neighbor 19.19.19.19 activate
    neighbor 19.19.19.19 send-label
    neighbor 2001:10:1:2::1 activate
  exit-address-family

XR1:
route-policy PASS
  pass
end-policy
!
router bgp 100
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
    allocate-label all
  !
```

CONTENTS >

```
neighbor 2.2.2.2
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
    next-hop-self
  !
  address-family ipv6 labeled-unicast
```

```
!
!
neighbor 10.19.20.20
  remote-as 20
  address-family ipv4 unicast
    route-policy PASS in
    route-policy PASS out
  !
  !
```

```
neighbor 2001:10:19:20::20
  remote-as 20
  address-family ipv6 unicast
    route-policy PASS in
    route-policy PASS out
  !
  !
  !
```

```
XR2:
  route-policy PASS
  pass
  end-policy
  !
  router bgp 20
    address-family ipv4 unicast
      network 20.20.20.20/32
    !
    address-family ipv6 unicast
      network 2001::20:20:20:20/128
    !
    neighbor 10.19.20.19
      remote-as 100
      address-family ipv4 unicast
        route-policy PASS in
        route-policy PASS out
      !
    !
    neighbor 2001:10:19:20::19
      remote-as 100
      address-family ipv6 unicast
        route-policy PASS in
        route-policy PASS out
      !
      !
      !
```

Verification

This example is similar to the “Basic MPLS Tunnels” lab in the terms that the customer’s traffic is tunneled inside MPLS over the Service Provider network, but the provider is not running L3VPN with the customer. If the provider was running L3VPN in this case, this design would be considered 6VPE instead of 6PE, where 6VPE uses VPNv6 BGP extensions, while 6PE simply uses IPv6 Unicast BGP extensions. 6PE is described under RFC 4798 (<https://tools.ietf.org/html/rfc4798>). The basic logic of this 6PE design is as follows.

The Service Provider network is IPv4 IGP and LDP enabled to start. This means that the PE routers can form an MPLS LSP between their Loopback interfaces. The link between the Customer Edge routers and Provider Edge routers is dual-stack enabled, running both Pv4 and IPv6. Since the PE routers already have label values allocated for each other’s IPv4 Loopbacks, there’s no problem with tunneling the Customer’s IPv4 traffic over the provider network. We can see this below as R1 and XR2 have reachability to each other’s IPv4 Loopbacks, even though the core of the network does not have these routes.

```
R1#traceroute 20.20.20.20 source 1.1.1.1

Type escape sequence to abort.
Tracing the route to 20.20.20.20

VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.2.2 4 msec 2 msec 0 msec
  2 20.2.4.4 [MPLS: Label 25 Exp 0] 8 msec 6 msec 5 msec
  3 * * *
  4 20.6.19.19 11 msec 5 msec 7 msec
  5 10.19.20.20 8 msec * 13 msec

R1#ping 20.20.20.20 source 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.20.20.20, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1

!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/8/13 ms

R4#show ip route 1.1.1.1

% Network not in table

R4#show ip cef 1.1.1.1

0.0.0.0/0
no route

R4#show ip route 20.20.20.20

% Subnet not in table

R4#show ip cef 20.20.20.20

0.0.0.0/0
no route
```

The reason this IPv4 over MPLS tunnel works is that the core of the network is simply label switching packets between the Loopbacks of R2 and XR1, the PE routers. However for this to work for IPv6 by the same logic, the core of the network would need IPv6 IGP routes to the IPv6 Loopbacks of the PE routers, and have LDPv6 labels for the IPv6 routes (LDPv6 is still current under draft stages under draft-ietf-mpls-ldp-ipv6 (<https://datatracker.ietf.org/doc/draft-ietf-mpls-ldp-ipv6/>)). In essence this would require the Service Provider to be dual-stack enabled to handle IPv6 transit to their

customers. Instead, 6PE offers an alternative “hack” on the MPLS process by allowing the next-hop value of an IPv6 BGP learned prefix to point towards an IPv4 next-hop for which the core of the MPLS network already has a label for.

The logic of this is similar to the MPLS L3VPN Inter-AS Option C, where the EBGp neighbors are exchanging IPv4 Unicast BGP routes along with MPLS Labels. However in this case the routers are exchanging *IPv6* Unicast BGP routes along with MPLS Labels, and have the next-hop value pointing to an IPv4 address.

The step by step verification of this process is as follows. First, the customer router R1 forms an IPv6 EBGp peering session with the provider, and advertises its IPv6 prefixes. This portion of the network could likewise be IPv6 IGP, but then BGP to IGP redistribution would be required.

```
R1#show bgp ipv6 unicast summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 5, main routing table version 5
2 network entries using 544 bytes of memory
2 path entries using 288 bytes of memory
2/2 BGP path/bestpath attribute entries using 496 bytes of memory
1 BGP AS-PATH entries using 40 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1368 total bytes of memory
BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2001:10:1:2::2	4	100	25	25	5	0	0	00:19:25	1

```
R1#show bgp ipv6 unicast neighbor 2001:10:1:2::2 advertised-routes
BGP table version is 5, 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
*> 2001::1:1:1/128	::	0		32768	i

Total number of prefixes 1

Here we see that R1 is advertising the prefix 2001::1:1:1/128 (it's Loopback) to R2. R2 learns this in the regular global IPv6 Unicast routing table, and installs it with the link-local next-hop of the connected neighbor.

```
R2#show ipv6 route 2001::1:1:1/128
Routing entry for 2001::1:1:1/128
  Known via "bgp 100", distance 20, metric 0, type external
  Route count is 1/1, share count 0
  Routing paths:
    FE80::250:56FF:FE9E:5913, GigabitEthernet1.12
      MPLS label: noLabel
    Last updated 00:19:10 ago
```

R2 has two IPv6 BGP peers, the CE router R1 and the PE router XR1. The transport for the BGP session between R1 and R2 uses IPv6, but the transport for the session between R2 and XR1 uses IPv4.

```
R2#show bgp ipv6 unicast summary
BGP router identifier 2.2.2.2, local AS number 100
BGP table version is 6, main routing table version 6
2 path entries using 288 bytes of memory
2/2 BGP path/bestpath attribute entries using 496 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1376 total bytes of memory
BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
19.19.19.19   4      100     21     27      6     0     0 00:18:17      1
2001:10:1:2::1 4        1     27     26      6     0     0 00:20:56      1
```

Two important things take place When R2 advertises R1's Loopback to XR1. The next-hop value of the IPv6 route gets a special encoding pointing it to the IPv4 Loopback0 of R2, and an MPLS Label is allocated. You can think of this label similar to one that would be allocated by VPNv4 BGP in MPLS L3VPN designs. Below we see the label value and next-hop that R2 assigns to the route being advertised to XR1.

```
R2#show bgp ipv6 unicast 2001::1:1:1:1/128
BGP routing table entry for 2001::1:1:1:1/128, version 5
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2
  Refresh Epoch 1
  1
  2001:10:1:2::1 (FE80::250:56FF:FE9E:5913) from 2001:10:1:2::1 (1.1.1.1)
    Origin IGP, metric 0, localpref 100, valid, external, best
    mpls labels in/out 17/nolabel
    rx pathid: 0, tx pathid: 0x0
```

```
R2#show bgp ipv6 unicast neighbors 19.19.19.19 advertised-routes
BGP table version is 6, 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
*> 2001::1:1:1:1/128	2001:10:1:2::1	0		0	1 i

Total number of prefixes 1

```
RP/0/0/CPU0:XR1#show bgp ipv6 unicast 2001::1:1:1:1/128
Sat May 16 14:23:15.412 UTC
BGP routing table entry for 2001::1:1:1:1/128
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          3         3
Last Modified: May 16 14:02:50.451 for 00:20:25
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    2001:10:19:20::20
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    2001:10:19:20::20
  1
  2.2.2.2 (metric 4) from 2.2.2.2 (2.2.2.2)
    Received Label 17
    Origin IGP, metric 0, localpref 100, valid, internal, best, group-best, import-candidate
    Received Path ID 0, Local Path ID 1, version 3
```

When XR1 installs this in the routing table, the next-hop points to the IPv4 compatible address ::ffff:2.2.2.2. This essentially tells the routing process to find the MPLS Transport Label for 2.2.2.2 and use that for the top of the stack. Meanwhile the IPv6 Unicast BGP learned label is inserted at the bottom of the stack, similar to how an MPLS L3VPN label would be.

```
RP/0/0/CPU0:XR1#show route ipv6 bgp
Sat May 16 14:25:09.454 UTC

B 2001::1:1:1:1/128
   [200/0] via ::ffff:2.2.2.2 (nexthop in vrf default), 00:22:18
B 2001::20:20:20:20/128
   [20/0] via fe80::250:56ff:fe9e:27ac. 00:20:43. GigabitEthernet0/0/0.1920
```

Further details can be seen in the hardware FIB entries of XR1. Notice that XR1 will encapsulate packets destined to 2001::1:1:1:1/128 with a two label stack, using label 28 and 17. Label 28 is the LDP label for the IPv4 next-hop of 2.2.2.2/32, and Label 17 is the BGP label (similar to a VPN label) that R1 advertised for 2001::1:1:1:1/128.

```
RP/0/0/CPU0:XR1#show cef ipv6 2001::1:1:1:1/128 detail
Sat May 16 14:26:08.510 UTC

2001::1:1:1:1/128, version 1235, internal 0x14004001 0x0 (ptr 0xa0ef5cf4) [1], 0x0 (0x0), 0x208 (0xa13f6460)

Updated May 16 14:02:50.895

Prefix Len 128, traffic index 0, precedence n/a, priority 4

gateway array (0xa0d49ed8) reference count 1, flags 0x4038, source rib (6), 0 backups
    [1 type 1 flags 0x48089 (0xa141062c) ext 0x0 (0x0)]

LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]

via ::ffff:2.2.2.2, 3 dependencies, recursive [flags 0x6000]

path-idx 0 NHID 0x0 [0xa0b8f17c 0x0]

next hop VRF - 'default', table - 0xe0000000

next hop ::ffff:2.2.2.2 via ::ffff:2.2.2.2:0

next hop 20.5.19.5/32 Gi0/0/0.519 labels imposed {28 17}

next hop 20.6.19.6/32 Gi0/0/0.619 labels imposed {28 17}

Load distribution: 0 (refcount 1)

Hash OK Interface Address
0 Y Unknown ::ffff:2.2.2.2:0
```

```
RP/0/0/CPU0:XR1#show mpls forwarding prefix 2.2.2.2/32
Sat May 16 14:32:08.895 UTC

Local Outgoing Prefix Outgoing Next Hop Bytes
Label Label or ID Interface Interface Switched
-----
16000 28 2.2.2.2/32 Gi0/0/0.519 20.5.19.5 6743
      28 2.2.2.2/32 Gi0/0/0.619 20.6.19.6 540
```

The route is then advertised from PE router XR1 to CE router XR2 over their normal IPv6 Unicast BGP session. XR2 receives this route and installs it via XR1's next-hop address.

```
RP/0/0/CPU0:XR2#show bgp ipv6 unicast
Sat May 16 14:33:17.351 UTC
BGP router identifier 20.20.20.20, local AS number 20
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0xe0800000 RD version: 4
BGP main routing table version 4
```

```
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
*> 2001::1:1:1/128	2001:10:19:20::19			0 100	1 i
*> 2001::20:20:20/128	::	0		32768	i

```
Processed 2 prefixes, 2 paths
```

```
RP/0/3/CPU0:XR2#show route ipv6 2001::1:1:1
```

```
Sat May 16 14:33:45.689 UTC
```

```
Routing entry for 2001::1:1:1/128
```

```
Known via "bgp 20", distance 20, metric 0
```

```
Tag 100, type external
```

```
Installed May 16 14:03:24.134 for 00:30:21
```

```
Routing Descriptor Blocks
```

```
fe80::250:56ff:fe9e:59fe, from 2001:10:19:20::19, via GigabitEthernet0/0/0.1920, BGP external
```

```
Route metric is 0
```

```
No advertising protos.
```

Now let's look at this from a data plane forwarding point of view. XR2, the CE router, sends a native IPv6 packet to the destination 2001::1:1:1:1.

```
RP/0/0/CPU0:XR2#ping 2001::1:1:1:1 source 2001::20:20:20
Sat May 16 14:34:40.365 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001::1:1:1:1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/15/19 ms
```

XR1, the PE router, gets the packet and consults the CEF table on what to do with it.

```
RP/0/0/CPU0:XR1#show cef ipv6 2001::1:1:1:1/128 detail
Sat May 16 14:35:04.173 UTC
2001::1:1:1:1/128, version 1235, internal 0x14004001 0x0 (ptr 0xa0ef5cf4) [1], 0x0 (0x0), 0x208 (0xa13f6460)
Updated May 16 14:02:50.895
Prefix Len 128, traffic index 0, precedence n/a, priority 4
gateway array (0xa0d49ed8) reference count 1, flags 0x4038, source rib (6), 0 backups
[1 tvpe 1 flaes 0x48089 (0xa141062c) ext 0x0 (0x0)]
```

```
LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
via ::ffff:2.2.2.2, 3 dependencies, recursive [flags 0x6000]
path-idx 0 NHID 0x0 [0xa0b8f17c 0x0]
next hop VRF - 'default', table - 0xe0000000
next hop ::ffff:2.2.2.2 via ::ffff:2.2.2.2:0
next hop 20.5.19.5/32 Gi0/0/0/0.519 labels imposed {28 17}
next hop 20.6.19.6/32 Gi0/0/0/0.619 labels imposed {28 17}
```

Load distribution: 0 (refcount 1)

Hash	OK	Interface	Address
0	Y	Unknown	::ffff:2.2.2.2:0

This output shows that even though XR1 has this as an IPv6 entry in the CEF table, the next-hop of ::ffff:2.2.2.2 recurses towards the IPv4 next-hop of 20.5.19.5 and 20.6.19.6, and should be label switched with the stack 17 28.

To simplify the verification below, shut down then link from XR1 to R5 to eliminate redundancy.

```
RP/0/0/CPU0:XR1#conf t
Sat May 16 14:41:37.467 UTC
RP/0/0/CPU0:XR1(config)#interface Gi0/0/0.519
RP/0/0/CPU0:XR1(config-subif)#shutdown
RP/0/0/CPU0:XR1(config-subif)#commit
Sat May 16 14:41:51.046 UTC
RP/0/0/CPU0:XR1(config-subif)#end
```

Enable the packet tracer feature on R6 to look at the MPLS switches packets are they are processed.

```
R6#debug platform packet-trace enable
Please remember to turn on 'debug platform condition start' for packet-trace to work
R6#debug platform packet-trace packet 1024 fia-trace
R6#debug platform condition interface g1.619 ingress
R6#debug platform condition start

RP/0/0/CPU0:XR2#ping 2001::1:1:1:1 source 2001::20:20:20:20
Sat May 16 14:43:32.589 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001::1:1:1:1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/23/69 ms
```

Notice that the label stack on R6, after XR1 performs label imposition, is seen as 28 and 17 when the packets are switched between the interfaces of P router R6.

R6#show platform packet-trace summary

Pkt	Input	Output	State	Reason
0	Gi1.619	internal0/0/rp:0	PUNT	55 (For-us control)
1	Gi1.619	Gi1.46	FWD	
2	Gi1.619	Gi1.46	FWD	
3	Gi1.619	Gi1.46	FWD	
4	Gi1.619	Gi1.46	FWD	
5	Gi1.619	Gi1.46	FWD	
6	Gi1.619	Gi1.36	FWD	

R6#show platform packet-trace packet 1

Packet: 1 CBUG ID: 1

Summary

Input : GigabitEthernet1.619

Output : GigabitEthernet1.46

State : FWD

Timestamp

Start : 2329445468506737 ns (05/16/2015 14:43:33.108414 UTC)

Stop : 2329445468573030 ns (05/16/2015 14:43:33.108481 UTC)

Path Trace

Feature: MPLS

Label Stack Entry[1]: 0x0001c03a

StackEnd:NO TTL: 58 EXP:0 Label:28

Label Stack Entry[2]: 0x0001113b

StackEnd:YES TTL: 59 EXP:0 Label:17

Feature: FIA_TRACE

Entry : 0x80880270 - DEBUG_COND_INPUT_PKT

Lapsed time: 8533 ns

Feature: FIA_TRACE

Entry : 0x804ac9c0 - MPLS_OUTPUT_LABEL_LOOKUP_ISSUE

Lapsed time: 9840 ns

Feature: FIA_TRACE

Entry : 0x804ac2e0 - MPLS_INPUT_LABEL_LOOKUP_CONSUME

Lapsed time: 4693 ns

Feature: FIA_TRACE

Entry : 0x804abdb0 - MPLS_OUTPUT_LABEL_LOOKUP_PROCESS

Lapsed time: 327360 ns

Feature: FIA_TRACE

Entry : 0x8075bdb0 - LAYER2_INPUT_VPLS

Lapsed time: 34293 ns

Feature: FIA_TRACE

Entry : 0x804ab540 - MPLS2IP_TRANSITION_FIA

Lapsed time: 3360 ns

Feature: FIA_TRACE

Entry : 0x800af6b0 - MPLS_INPUT_GOTO_OUTPUT_FEATURE

Lapsed time: 18986 ns

Feature: FIA_TRACE

Entry : 0x8045bd00 - IPV4_VFR_REFRAG

Lapsed time: 7040 ns

Feature: FIA_TRACE

Entry : 0x80466d20 - IPV6_MC_INPUT_VFR_REFRAG

Lapsed time: 5920 ns

Feature: FIA_TRACE

```

Entry      : 0x807bf9a0 - MPLS_OUTPUT_ADD_LABEL
Lapsed time: 7626 ns
Feature: FIA_TRACE

Entry      : 0x807b7030 - IPV6_OUTPUT_L2_REWRITE
Lapsed time: 78266 ns
Feature: FIA_TRACE

Entry      : 0x804aadf0 - MPLS_OUTPUT_FRAG
Lapsed time: //60 ns
Feature: FIA_TRACE

Entry      : 0x8060bc80 - MPLS_OUTPUT_DROP_POLICY
Lapsed time: 24213 ns
Feature: FIA_TRACE

Entry      : 0x80954900 - MARMOT_SPA_D_TRANSMIT_PKT
Lapsed time: 111520 ns

```

R6 swaps top label 28 for label 16 or 17. For the specific packet above, we know that the label was swapped for Label 17 instead of 16, as the output interface was Gig1.46 towards R4.

```

R6#show mpls forwarding-table labels 28 detail
Local   Outgoing Prefix          Bytes Label  Outgoing  Next Hop
Label   Label    or Tunnel Id    Switched     interface
28      16       2.2.2.2/32      1368         Gi1.36    20.3.6.3
MAC/Encaps=18/22, MRU=1500, Label Stack{16}
0050569E6E6A0050569E5CEC810000248847 00010000
No output feature configured
Per-destination load-sharing, slots: 0 2 4 6 8 10 12 14
17      17       2.2.2.2/32      1890         Gi1.46    20.4.6.4
MAC/Encaps=18/22, MRU=1500, Label Stack{17}
0050569E13020050569E5CEC8100002E8847 00011000
No output feature configured
Per-destination load-sharing, slots: 1 3 5 7 9 11 13 15

```

Notice that just like in a L3VPN design, P routers have no FEC bindings for the "VPN labels". In this case, P routers have no FEC bindings for the label advertised via BGP from R1 towards XR1 for IPv6 prefix 2001::1:1:1/128.

```

R6#show mpls forwarding-table labels 17 detail
Local   Outgoing Prefix          Bytes Label  Outgoing  Next Hop
Label   Label    or Tunnel Id    Switched     interface

```

The same thing occurs in the reverse direction, where XR2 advertises its IPv6 Unicast BGP route to XR1, XR1 sends it as an IPv6 Labeled Unicast route to R2, and finally R2 sends it as a regular IPv6 Unicast BGP route to R1.

```
RP/0/0/CPU0:XR2#show bgp ipv6 unicast neighbors 2001:10:19:20::19 advertised-routes
```

```
Sat May 16 14:54:02.056 UTC
```

Network	Next Hop	From	AS Path
2001::20:20:20:20/128			
	2001:10:19:20::20		
		Local	20i

```
Processed 1 prefixes, 1 paths
```

```
RP/0/0/CPU0:XR1#show bgp ipv6 unicast 2001::20:20:20:20/128
```

```
Sat May 16 14:54:25.264 UTC
```

```
BGP routing table entry for 2001::20:20:20:20/128
```

```
Versions:
```

Process	bRIB/RIB	SendTblVer
Speaker	5	5
Local Label: 16029		

```
Last Modified: May 16 14:04:26.451 for 00:49:59
```

```
Paths: (1 available, best #1)
```

```
Advertised to peers (in unique update groups):
```

```
2.2.2.2
```

```
Path #1: Received by speaker 0
```

```
Advertised to peers (in unique update groups):
```

```
2.2.2.2
```

```
20
```

```
2001:10:19:20::20 from 2001:10:19:20::20 (20.20.20.20)
```

```
Origin IGP, metric 0, localpref 100, valid, external, best, group-best, import-candidate
```

```
Received Path ID 0, Local Path ID 1, version 5
```

```
Origin-AS validity: not-found
```

Notice that IOS displays the IPv4 Compatible IPv6 address as the next-hop in the following output, but IOS-XR displays the regular IPv4 next-hop in its corresponding output.

```
R2#show bgp ipv6 unicast 2001::20:20:20:20/128
BGP routing table entry for 2001::20:20:20:20/128, version 6
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  20
  ::FFFF:19.19.19.19 (metric 4) from 19.19.19.19 (19.19.19.19)
  Origin IGP, metric 0, localpref 100, valid, internal, best
  mpls labels in/out nlabel/16029
  rx pathid: 0, tx pathid: 0x0
```

```
R2#show bgp ipv6 unicast neighbors 2001:10:1:2::1 advertised-routes
BGP table version is 7, 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
*>i 2001::20:20:20:20/128	::FFFF:19.19.19.19	0	100	0	20 i

Total number of prefixes 1

```
R1#show bgp ipv6 unicast 2001::20:20:20:20/128
BGP routing table entry for 2001::20:20:20:20/128, version 7
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  100 20
  2001:10:1:2::2 (FE80::250:56FF:FE9E:35D1) from 2001:10:1:2::2 (2.2.2.2)
  Origin IGP, localpref 100, valid, external, best
  rx pathid: 0, tx pathid: 0x0
```

Traffic in the data plane goes from R1 to R2, is label encapsulated by R2, is label switched through the core, and then comes back out as unlabeled IPv6 packets on the XR1 to XR2 link.

```
R1#traceroute ipv6
```

```
Target IPv6 address: 2001::20:20:20:20
```

```
Source address: 2001::1:1:1:1
```

```
Insert source routing header? [no]:
```

```
Numeric display? [no]:
```

```
Timeout in seconds [3]:
```

```
Probe count [3]:
```

```
Minimum Time to Live [1]:
```

```
Maximum Time to Live [30]: Priority [0]:
```

```
Port Number [0]:
```

```
Type escape sequence to abort.
```

```
Tracing the route to 2001::20:20:20:20
```

```
1 2001:10:1:2::2 3 msec 2 msec 1 msec
2 ::FFFF:20.2.4.4 [MPLS: Labels 25/16029 Exp 0] 21 msec 19 msec 34 msec
3 ::FFFF:20.4.6.6 [MPLS: Labels 34/16029 Exp 0] 18 msec 18 msec 18 msec
4 2001:10:19:20::19 [MPLS: Label 16029 Exp 0] 36 msec 20 msec 28 msec
5 2001::20:20:20:20 [AS 20] 20 msec 20 msec 16 msec
```

R2's IPv6 CEF entries point towards and IPv4 Label Switch Path.

```
R2#show ipv6 cef 2001::20:20:20:20/128 detail
```

```
2001::20:20:20:20/128, epoch 0, flags [rib defined all labels]
```

```
recursive via 19.19.19.19 label 16029
```

```
nexthop 20.2.3.3 GigabitEthernet1.23 label 26
```

```
nexthop 20.2.4.4 GigabitEthernet1.24 label 25
```

« [MPLS L3VPN Carrier Supporting Carrier \(/workbook/view/service-provider-v4/task/mpls-l3vpn-carrier-supporting-carrier-Mjg3MQ%3D%3D\)](#) | [MPLS L3VPN 6VPE \(/workbook/view/service-provider-v4/task/mpls-l3vpn-6vpe-Mjg3Mw%3D%3D\)](#) »