MPLS L3VPN Carrier Supporting Carrier

« 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) | MPLS L3VPN 6PE (/workbook/view/service-provider-v4/task/mpls-l3vpn-6pe-Mjg3Mg%3D%3D) »

Last updated: April 22, 2016

Note:

Initial Configuration & Diagrams: Load the initial configuration files for the section named **Carrier Supporting Carrier**, 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 **Carrier Supporting Carrier Diagram** in order to complete this task.

Task

- This scenario consists of two carrier networks and four customer sites as follows:
  - The "Core Carrier":
    - Consists of R1, R7, R8, and XR1.
    - Uses OSPF + LDP for internal label distribution
    - Uses BGP AS 17819
  - The "Customer Carrier":
    - Consists of R2, R3, R4, R5, R6, and XR2.
    - Uses IS-IS + LDP for internal label distribution
    - Uses BGP AS 100
  - Customer Site VPN_A
    - Consists of R9 and R10's 30.0.0.0 networks.
    - Preconfigured for VRF Lite RIPv2 routing.
  - Customer Site VPN_B
    - Consists of R9 and R10's 40.0.0.0 networks.
    - Preconfigured for VRF Lite RIPv2 routing.
- Configure IGP routing and LDP in the Core Carrier network as follows:
  - Enable OSPF Area 0 on the links between R1 & R7, R7 & R8, and R8 & XR1.
  - Enable OSPF Area 0 on the Loopback0 interfaces of these routers as passive interfaces.
  - Enable LDP on the links between between R1 & R7, R7 & R8, and R8 & XR1.
- Configure IGP routing and LDP in the Customer Carrier network as follows:
  - Use IS-IS Process-ID 1 and NET addresses in the format 49.0001.0000.0000.000X.00 where X is the router number.
  - Enable IS-IS Level 2 on the links between R2 & R5, R5 & R3, R4 & R6, and R6 & XR2.
  - Advertise the Loopback0 interfaces of these routers into IS-IS Level 2 as passive interfaces.
  - Enable LDP on the links between R2 & R5, R5 & R3, R4 & R6, and R6 & XR2.
- Configure the VRF CSC on the Core Carrier PE routers R1 and XR1 as follows:
  - VRF Name: CSC

- Route Distinguisher: 17819:1
- Route Target Import: 17819:1
- Route Target Export: 17819:1
- Assign the VRF to the links connecting to R3 and R4 respectively.

CONTENTS

- Configure the following VRFs on the Customer Carrier PE routers R2 and XR2 follows:
  - VRF VPN_A:
    - Route Distinguisher: 100:1
    - Route Target Import: 100:1
    - Route Target Export: 100:1
    - Assign this VRF on the links in the 30.0.0.0 network on R2 and XR2.
    - Enable RIPv2 routing for the VRF on R2 and XR2.
  - VRF VPN_B:
    - Route Distinguisher: 100:2
    - Route Target Import: 100:2
    - Route Target Export: 100:2
    - Assign this VRF on the links in the 40.0.0.0 network on R2 and XR2.
    - Enable RIPv2 routing for the VRF on R2 and XR2.
- Configure IPv4 Labeled Unicast BGP peerings as follows:
  - R1 and XR1 are in AS 17819.
  - R3 and R4 are in AS 100.
  - R1 should form an IPv4 Unicast EBGP peering with R3.
  - XR1 should form an IPv4 Unicast EBGP peering with R4.
  - Advertise all links that are part of the Customer Carrier network into BGP on R3 and R4, including Loopbacks.
  - Include BGP MPLS Labels advertisements between R1 & R3 and R4 & XR1.
- Configure a VPNv4 iBGP peering between the Customer Carrier PE routers R2 and XR2.
- Redistribute between VPNv4 BGP and the VRF aware IGP processes on R2 and XR2.
- Once complete the following reachability should be achieved:
  - Customer routers R9 and R10 should have full IP reachability to each other’s 30.x.x.x networks in VRF VPN_A.
  - Customer routers R9 and R10 should have full IP reachability to each other’s 40.x.x.x networks in VRF VPN_B.
  - Traceroutes between these networks should indicate that an additional level of labels is used in the LSP through the Core Carrier network.

Configuration **Click to *collapse***

```
R1:
vrf definition CSC
 rd 17819:1
 !
 address-family ipv4
 route-target export 17819:1
 route-target import 17819:1
 exit-address-family
!
interface Loopback0
 ip ospf 1 area 0
!
interface GigabitEthernet1.13
 vrf forwarding CSC
 ip address 20.1.3.1 255.255.255.0
 mpls bgp forwarding
!
interface GigabitEthernet1.17
 ip ospf 1 area 0
 mpls ip
!
router ospf 1
 passive-interface Loopback0
!
router bgp 17819
 no bgp default ipv4-unicast
 neighbor 19.19.19.19 remote-as 17819
 neighbor 19.19.19.19 update-source Loopback0
!
address-family vpnv4
 neighbor 19.19.19.19 activate
 neighbor 19.19.19.19 send-community extended
exit-address-family
!
address-family ipv4 vrf CSC
 neighbor 20.1.3.3 remote-as 100
 neighbor 20.1.3.3 activate
 neighbor 20.1.3.3 as-override
 neighbor 20.1.3.3 send-label
exit-address-family
!

R2:
vrf definition VPN_A
 rd 100:1
 !
 address-family ipv4
 route-target export 100:1
 route-target import 100:1
 exit-address-family
!
vrf definition VPN_B
 rd 100:2
```

CONTENTS

```
 !
 address-family ipv4
 route-target export 100:2
 route-target import 100:2
 exit-address-family
!
interface GigabitEthernet1.25
 ip router isis 1
 mpls ip
!
interface GigabitEthernet1.29
 vrf forwarding VPN_A
 ip address 30.2.9.2 255.255.255.0
!
interface GigabitEthernet1.210
 vrf forwarding VPN_B
 ip address 40.2.10.2 255.255.255.0
!
router isis 1
 net 49.0001.0000.0000.0002.00
 is-type level-2-only
 passive-interface Loopback0
!
router rip
 !
 address-family ipv4 vrf VPN_B
  redistribute bgp 100 metric 1
  network 40.0.0.0
  no auto-summary
  version 2
exit-address-family
!
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 20.20.20.20 remote-as 100
 neighbor 20.20.20.20 update-source Loopback0
 !
 address-family vpnv4
  neighbor 20.20.20.20 activate
  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
```

CONTENTS

```
  redistribute rip
 exit-address-family

R3:
interface GigabitEthernet1.13
 mpls bgp forwarding
!
interface GigabitEthernet1.35
 ip router isis 1
 mpls ip
!
router isis 1
 net 49.0001.0000.0000.0003.00
 is-type level-2-only
 redistribute bgp 100
 passive-interface Loopback0
!
router bgp 100
 network 2.2.2.2 mask 255.255.255.255
 network 3.3.3.3 mask 255.255.255.255
 network 5.5.5.5 mask 255.255.255.255
 network 20.1.3.0 mask 255.255.255.0
 network 20.2.5.0 mask 255.255.255.0
 network 20.3.5.0 mask 255.255.255.0
 neighbor 20.1.3.1 remote-as 17819
 neighbor 20.1.3.1 send-label

R4:
interface GigabitEthernet1.46
 ip router isis 1
 mpls ip
!
interface GigabitEthernet1.419
 mpls bgp forwarding
!
router isis 1
 net 49.0001.0000.0000.0004.00
 is-type level-2-only
 redistribute bgp 100
 passive-interface Loopback0
!
router bgp 100
 network 4.4.4.4 mask 255.255.255.255
 network 6.6.6.6 mask 255.255.255.255
 network 20.4.6.0 mask 255.255.255.0
 network 20.4.19.0 mask 255.255.255.0
 network 20.6.20.0 mask 255.255.255.0
 network 20.20.20.20 mask 255.255.255.255
 neighbor 20.4.19.19 remote-as 17819
 neighbor 20.4.19.19 send-label

R5:
interface GigabitEthernet1.25
 ip router isis 1
```

CONTENTS

```
 mpls ip
!
interface GigabitEthernet1.35
 ip router isis 1
 mpls ip
!
router isis 1
 net 49.0001.0000.0000.0005.00
 is-type level-2-only
 passive-interface Loopback0
!

R6:
interface GigabitEthernet1.46
 ip router isis 1
 mpls ip
!
interface GigabitEthernet1.620
 ip router isis 1
 mpls ip
!
router isis 1
 net 49.0001.0000.0000.0006.00
 is-type level-2-only
 passive-interface Loopback0



R7:
mpls label protocol ldp
!
interface GigabitEthernet1.17
 mpls ip
!
interface GigabitEthernet1.78
 mpls ip
!
router ospf 1
 passive-interface Loopback0
 network 7.7.7.7 0.0.0.0 area 0
 network 10.0.0.0 0.255.255.255 area 0

R8:
mpls label protocol ldp
!
interface GigabitEthernet1.78
 mpls ip
!
interface GigabitEthernet1.819
 mpls ip
!
router ospf 1
 passive-interface Loopback0
 network 8.8.8.8 0.0.0.0 area 0
 network 10.0.0.0 0.255.255.255 area 0
```

```
XR1:
vrf CSC
 address-family ipv4 unicast
  import route-target 17819:1
 !
 export route-target
  17819:1
!
interface GigabitEthernet0/0/0/0.419
 vrf CSC
 no ipv4 address
 ipv4 address 20.4.19.19 255.255.255.0
!
route-policy PASS
 pass
end-policy
!
router static
 vrf CSC
  address-family ipv4 unicast
   20.4.19.4/32 GigabitEthernet0/0/0/0.419
!
router ospf 1
 area 0
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0.819
!
router bgp 17819
 address-family vpnv4 unicast
 !
 neighbor 1.1.1.1
  remote-as 17819
  update-source Loopback0
  address-family vpnv4 unicast
 !
!
vrf CSC
 rd 17819:1
 address-family ipv4 unicast
  allocate-label all
 !
 neighbor 20.4.19.4
  remote-as 100
  address-family ipv4 labeled-unicast
    route-policy PASS in
    route-policy PASS out
    as-override
   !
  !
 !
!
```

CONTENTS

```
mpls ldp
 interface GigabitEthernet0/0/0/0.819
 !
!

XR2:
vrf VPN_A
 address-family ipv4 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
!
vrf VPN_B
 address-family ipv4 unicast
  import route-target
   100:2
  !
  export route-target
   100:2
!
interface GigabitEthernet0/0/0/0.920
 vrf VPN_B
 no ipv4 address
 ipv4 address 40.9.20.20 255.255.255.0
!
interface GigabitEthernet0/0/0/0.1020
 vrf VPN_A
 no ipv4 address
 ipv4 address 30.10.20.20 255.255.255.0
!
route-policy BGP_TO_RIP
  set rip-metric 1
end-policy
!
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0020.00
 interface Loopback0
  passive
  address-family ipv4 unicast
  !
!
interface GigabitEthernet0/0/0/0.620
 address-family ipv4 unicast
!
router bgp 100
 address-family vpnv4 unicast
 !
 neighbor 2.2.2.2
  remote-as 100
  update-source Loopback0
```

```
  address-family vpnv4 unicast
 !
!
vrf VPN_A
 rd 100:1
 address-family ipv4 unicast
  redistribute rip
 !
!
vrf VPN_B
 rd 100:2
 address-family ipv4 unicast
  redistribute rip
  !
 !
!
mpls ldp
 interface GigabitEthernet0/0/0/0.620
 !
!
router rip
 vrf VPN_A
  interface GigabitEthernet0/0/0/0.1020
  !
  redistribute bgp 100 route-policy BGP_TO_RIP
 !
 vrf VPN_B
  interface GigabitEthernet0/0/0/0.920
  !
  redistribute bgp 100 route-policy BGP_TO_RIP
 !
!
End
```

Verification

Carrier Supporting Carrier (CsC), or what is sometimes referred to as Hierarchical MPLS VPNs, is when a typically smaller Service Provider uses another larger Service Provider's MPLS network for transport between the smaller SP's sites, and ultimately between the sites of the smaller SP's customers. In this type of design the larger Service Provider is considered to be the "Core Carrier", while the smaller Service Provider is considered the "Customer Carrier". Carrier Supporting Carrier, or Carrier's Carrier, is documented under RFC-4364 Section 9 (https://tools.ietf.org/html/rfc4364#section-9).

This design is common in cases where a Service Provider has customers in geographically diverse areas, for example in London and Los Angeles, but does not own long haul transit links between these locations. With CsC the Customer Carrier can still transparently offer services to its customers in London and Los Angeles without them knowing that they are actually transiting through a third party to provide services.

From a configuration point of view a CsC design is similar to an Inter-AS MPLS L3VPN, with the exception that the Core Carrier to Customer Carrier link is MPLS enabled, as is treated like a normal L3VPN customer site from the Core Carrier's point of view. These links are then referred to as the CsC-PE to CsC-CE links, with the Core Carrier being the PE side and the Customer Carrier being the CE side.

One of the key points to keep in mind about this design is that since the Core Carrier does not have knowledge of the final customer prefixes, all traffic must follow an end-to-end LSP as it moves from the Customer Carrier through the Core Carrier network. In order to achieve this, the next-hop values that are used for the Customer Carrier's VPNv4 BGP peering sessions must have corresponding LSPs inside the Core Carrier.

In this specific configuration example the final customer networks are represented by the VPN_A and VPN_B networks on R9 and R10. Note that this portion of the configuration is unrelated to the rest of the design, as the VRF Lite/ Multi VRF CE configuration on R9 and R10 is simply used to simulate more routers than are physically used in the topology.

Like our other previous examples the final verification is to test end-to-end reachability between these sites, as seen below:

```
R9#show ip route vrf VPN_A rip

Routing Table: VPN_A
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

      30.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
R        30.10.10.10/32 [120/1] via 30.2.9.2, 00:00:04, GigabitEthernet1.29
R        30.10.20.0/24 [120/1] via 30.2.9.2, 00:00:04, GigabitEthernet1.29

R9#ping vrf VPN_A 30.10.10.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.10.10.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/5/9 ms

R9#show ip route vrf VPN_B rip

Routing Table: VPN_B
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

      40.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
R        40.2.10.0/24 [120/1] via 40.9.20.20, 00:00:17, GigabitEthernet1.920
R        40.10.10.10/32 [120/1] via 40.9.20.20, 00:00:17, GigabitEthernet1.920

R9#ping vrf VPN_B 40.10.10.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 40.10.10.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/9 ms
```

To see how this design is different from previous examples, we need to follow the individual control plane and data plane paths separately of a final end prefix, as many common problems in this design will appear in the data plane that are not completely evident in the control plane. For example, you may see that the final customer is learning and installing the prefixes correctly, but they don't actually have transport between the sites. To verify the individual steps, we first start at the final PE to CE link between R2 in the Customer Carrier network and its final customer R10.


```
R2#show ip route vrf VPN_B 40.10.10.10

Routing Table: VPN_B
Routing entry for 40.10.10.10/32
  Known via "rip", distance 120, metric 1
  Redistributing via rip, bgp 100
  Advertised by bgp 100
  Last update from 40.2.10.10 on GigabitEthernet1.210, 00:00:17 ago
  Routing Descriptor Blocks:
  * 40.2.10.10, from 40.2.10.10, 00:00:17 ago, via GigabitEthernet1.210
      Route metric is 1, traffic share count is 1

R2#show bgp vpnv4 unicast vrf VPN_B 40.10.10.10
BGP routing table entry for 100:2:40.10.10.10/32, version 5
Paths: (1 available, best #1, table VPN_B)
 Advertised to update-groups:
    1
 Refresh Epoch 1
 Local
   40.2.10.10 (via vrf VPN_B) from 0.0.0.0 (2.2.2.2)
     Origin incomplete, metric 1, localpref 100, weight 32768, valid, sourced, best
     Extended Community: RT:100:2
     mpls labels in/out 23/nolabel
     rx pathid: 0, tx pathid: 0x0
```

R2 learns the prefix 40.10.10.10/32 in VRF VPN_B via RIP, and redistributes this into VPNv4 BGP. Like our previous designs we can see that this first step creates two important building blocks of the L3VPN network, the MPLS VPN Label and the VPNv4 BGP next-hop value. R2 then advertises this route to its VPNv4 BGP peer XR2, who is servicing the customer sites on the remote end of the network.

> CONTENTS

```
R2#show bgp vpnv4 unicast all neighbors 20.20.20.20 advertised-routes
BGP table version is 21, 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: 100:1 (default for vrf VPN_A)
*> 30.2.9.0/24      0.0.0.0         0           32768 ?
*> 30.9.9.9/32      30.2.9.9        1           32768 ?
Route Distinguisher: 100:2 (default for vrf VPN_B)
*> 40.2.10.0/24     0.0.0.0         0           32768 ?
 *> 40.10.10.10/32 40.2.10.10      1           32768 ?

Total number of prefixes 4

RP/0/0/CPU0:XR2#show bgp vpnv4 unicast vrf VPN_B 40.10.10.10/32
Fri May 15 22:51:15.813 UTC
BGP routing table entry for 40.10.10.10/32, Route Distinguisher: 100:2
Versions:
 Process           bRIB/RIB  SendTblVer
 Speaker                 17          17
Last Modified: May 15 22:33:09.625 for 00:18:06
Paths: (1 available, best #1)
 Not advertised to any peer
 Path #1: Received by speaker 0
 Not advertised to any peer
 Local
    2.2.2.2 (metric 20) from 2.2.2.2 (2.2.2.2)
      Received Label 23
      Origin incomplete, metric 1, localpref 100, valid, internal, best, group-best, import-candidate, imported
      Received Path ID 0, Local Path ID 1, version 17
      Extended community: RT:100:2
   Source VRF: VPN_B, Source Route Distinguisher: 100:2
```

XR2 receives the VPNv4 route from R2 with a next-hop of 2.2.2.2, a VPN Label of 23, and a Route Target of 100:2. Assuming that XR2 properly imports this route into the VRF and redistributes it into the PE to CE routing process, the next point we need to verify is whether or not XR2 has a Label Switch Path towards the next-hop value 2.2.2.2. To form this LSP we must first know about the route in the global routing table.

```
RP/0/0/CPU0:XR2#show route 2.2.2.2/32
Fri May 15 22:52:48.067 UTC

Routing entry for 2.2.2.2/32
     Known via "isis 1", distance 115, metric 20, type level-2
    Installed Mar 21 21:40:41.915 for 00:19:58
    Routing Descriptor Blocks
        20.6.20.6, from 4.4.4.4, via GigabitEthernet0/0/0/0.620
            Route metric is 20
    No advertising protos.
```

XR2 learns about 2.2.2.2/32 via IS-IS. Since this is an IGP learned prefix, it means that the MPLS Label must be coming from either LDP or RSVP-TE in order to be used. The MPLS FIB (LFIB) will tell us if the label exists, as follows:

```
RP/0/0/CPU0:XR2#show mpls forwarding prefix 2.2.2.2/32
Fri May 15 22:53:31.344 UTC
Local   Outgoing    Prefix             Outgoing       Next Hop       Bytes
Label   Label       or ID              Interface                     Switched
------ ----------- ------------------ ------------    --------------- ------------
16007   19          2.2.2.2/32         Gi0/4/0/0.620   20.6.20.6       13585

RP/0/0/CPU0:XR2#show cef 2.2.2.2/32
Fri May 15 23:15:03.285 UTC
 2.2.2.2/32 , version 124, internal 0x4004001 0x0 (ptr 0xa0edbaf4) [1], 0x0 (0xa0ea79e0), 0x228 (0xa13f62d0)
  Updated May 15 22:32:52.718
  local adjacency 20.6.20.6
  Prefix Len 32, traffic index 0, precedence n/a, priority 3
    via 20.6.20.6, GigabitEthernet0/0/0/0.620 , 5 dependencies, weight 0, class 0 [flags 0x0]
     path-idx 0 NHID 0x0 [0xa15c80cc 0x0]
     next hop 20.6.20.6
     tx adjacency
      local label 16007      labels imposed {19}
```

XR2 does in fact have an LSP for this prefix, specifically using the label value 19 via the next- hop 20.6.20.6 (R6). The next-hop router R6 likewise learns the route via IS-IS, and has a label value for it derived from LDP.

```
R6#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "isis", distance 115, metric 10, type level-2
  Redistributing via isis 1
  Last update from 20.4.6.4 on GigabitEthernet1.46, 00:21:47 ago
  Routing Descriptor Blocks:
  * 20.4.6.4, from 4.4.4.4, 00:21:47 ago, via GigabitEthernet1.46
      Route metric is 10, traffic share count is 1

R6#show mpls forwarding-table 2.2.2.2
Local   Outgoing    Prefix          Bytes Label Outgoing    Next Hop
Label   Label       or Tunnel Id    Switched    interface
19         20          2.2.2.2/32       15372         Gi1.46      20.4.6.4

R6#show ip cef 2.2.2.2/32 detail
2.2.2.2/32, epoch 2
  local label info: global/19
   nexthop 20.4.6.4 GigabitEthernet1.46 label 20
```

Beyond this we go to R4, who is considered to be the Customer Carrier CE, or the CsC CE router. Here R4 is learning the prefix 2.2.2.2/32 via the EBGP neighbor XR1, who is the CSC PE. R4 and XR1 are regular IPv4 Unicast BGP peers, but are also exchanging MPLS labels for the BGP routes. This can be seen from the global routing table output below, as R4 has a BGP derived label for the prefix.

```
R4#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "bgp 100", distance 20, metric 0
  Tag 17819, type external
  Redistributing via isis 1
  Advertised by isis 1 metric-type internal level-2
  Last update from 20.4.19.19 00:23:20 ago
  Routing Descriptor Blocks:
  * 20.4.19.19, from 20.4.19.19, 00:23:20 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 17819
      MPLS label: 16017

R4#show ip cef 2.2.2.2/32 detail
2.2.2.2/32, epoch 2, flags [rib defined all labels]
  local label info: global/20
  recursive via 20.4.19.19 label 16017
    attached to GigabitEthernet1.419



R4#show ip bgp labels
    Network          Next Hop      In label/Out label
    2.2.2.2/32       20.4.19.19      nolabel/16017
    3.3.3.3/32       20.4.19.19      nolabel/16018
    4.4.4.4/32       0.0.0.0         imp-null/nolabel
    5.5.5.5/32       20.4.19.19      nolabel/16019
    6.6.6.6/32       20.4.6.6        16/nolabel
    20.1.3.0/24      20.4.19.19      nolabel/16020
    20.2.5.0/24      20.4.19.19      nolabel/16021
    20.3.5.0/24      20.4.19.19      nolabel/16022
    20.4.6.0/24      0.0.0.0         imp-null/nolabel
    20.4.19.0/24     0.0.0.0         imp-null/nolabel
    20.6.20.0/24     20.4.6.6        17/nolabel
    20.20.20.20/32   20.4.6.6        19/nolabel
```

R4 is then taking the BGP learned route and redistributing it into IS-IS. This is the reason that the rest of the Customer Carrier network (i.e. R6 and XR2) knows about the prefix via IS-IS. R6 and XR2 learned the labels for these redistributed routes, as LDP generated labels for them after entering the ISIS process. Another option would be to run BGP + Label everywhere in the Customer Carrier network, however this design is typically more difficult to maintain due to the administrative overhead.

From the Core Carrier Provider Edge Router (XR1’s) point of view, this route is being learned via VPNv4 iBGP from the other CSC PE (R1).

```
RP/0/0/CPU0:XR1#show route vrf CSC 2.2.2.2/32
Fri May 15 23:01:24.871 UTC


Routing entry for 2.2.2.2/32
  Known via "bgp 17819" , distance 200, metric 20
  Tag 100, type internal
  Installed May 15 22:32:48.109 for 00:28:36
  Routing Descriptor Blocks
    1.1.1.1, from 1.1.1.1
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000
      Route metric is 20
  No advertising protos.
```

Note that since this is a new VPNv4 route that is being originated, a new VPN Label and next-hop value are being set. This is why this design is sometimes called Hierarchical MPLS VPNs, because it's basically one L3VPN inside of another L3VPN. The impact of this will be evident when we verify the actual traffic flow in the data plane.

From the CSC PE's point of view as seen below, the VPNv4 route 2.2.2.2/32 is learned via the remote PE 1.1.1.1. This means that XR1 needs a LSP for 1.1.1.1, which is solved by running OSPF + LDP in the Core Carrier network.

```
RP/0/0/CPU0:XR1#sh bgp vpnv4 unicast vrf CSC 2.2.2.2
Fri May 15 23:02:38.256 UTC
BGP routing table entry for 2.2.2.2/32, Route Distinguisher: 17819:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 11          11
    Local Label: 16017
Last Modified: May 15 22:32:48.451 for 00:29:49
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  100
    1.1.1.1 (metric 4) from 1.1.1.1 (1.1.1.1)
      Received Label 19
      Origin IGP, metric 20, localpref 100, valid, internal, best, group-best, import-candidate, imported
      Received Path ID 0, Local Path ID 1, version 11
   Extended community: RT:17819:1
   Source VRF: CSC , Source Route Distinguisher: 17819:1


RP/0/0/CPU0:XR1#show cef vrf CSC 2.2.2.2/32
Fri May 15 23:16:12.120 UTC
2.2.2.2/32, version 6, internal 0x4004001 0x0 (ptr 0xa0edc374) [1], 0x0 (0x0), 0x208 (0xa13f60f0)
 Updated May 15 22:32:48.128
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 1.1.1.1, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa145faf4 0x0]
    next hop VRF - 'default', table - 0xe0000000
    next hop 1.1.1.1 via 16011/0/21
      local label 16017
      next hop 10.8.19.8/32 Gi0/0/0/0.819 labels imposed {17 19}


RP/0/0/CPU0:XR1#traceroute 1.1.1.1 source 19.19.19.19
Fri May 15 23:03:22.183 UTC

Type escape sequence to abort.
Tracing the route to 1.1.1.1

 1  10.8.19.8 [ MPLS: Label 17 Exp 0 ] 9 msec  0 msec  0 msec
 2  10.7.8.7 [ MPLS: Label 16 Exp 0 ] 0 msec  0 msec  0 msec
 3  10.1.7.1 0 msec  *  0 msec
```

This LSP, between R1 and XR1's Loopback0 (the VPNv4 next-hop), will serve as the "transport" LSP for the L3VPN of the Customer Carrier.

At the next CSC PE router R1, this prefix is being learned an IPv4 Unicast Labeled EBGP route from the CSC CE router R3. R3 is advertising this BGP route to R1 with a label of 18.

CONTENTS

```
R1#show bgp vpnv4 unicast vrf CSC 2.2.2.2/32
BGP routing table entry for _17819:1:2.2.2.2/32 , version 5
Paths: (1 available, best #1, table CSC)
 Advertised to update-groups:
    2
 Refresh Epoch 1
 100
   20.1.3.3 (via vrf CSC) from 20.1.3.3 (3.3.3.3)
     Origin IGP, metric 20, localpref 100, valid, external, best
      Extended Community: RT:17819:1
      mpls labels in/out 19/18
     rx pathid: 0, tx pathid: 0x0
```

The MPLS Label value allocated by BGP from R3 can be verified from the above output, or as follows:

```
R1#show ip cef vrf CSC 2.2.2.2/32 detail
2.2.2.2/32, epoch 0, flags [rib defined all labels]
  local label info: other/19
  recursive via 20.1.3.3 label 18
    attached to GigabitEthernet1.13



R1#show bgp vpnv4 unicast all labels
Network          Next Hop      In label/Out label
Route Distinguisher: 17819:1 (CSC)
  2.2.2.2/32       20.1.3.3        19/18
  3.3.3.3/32       20.1.3.3        16/imp-null
  4.4.4.4/32       19.19.19.19     27/16023
  5.5.5.5/32       20.1.3.3        21/17
  6.6.6.6/32       19.19.19.19     30/16024
  20.1.3.0/24      20.1.3.3        17/imp-null
  20.2.5.0/24      20.1.3.3        20/19
  20.3.5.0/24      20.1.3.3        18/imp-null
  20.4.6.0/24      19.19.19.19     28/16025
  20.4.19.0/24     19.19.19.19     29/16026
  20.6.20.0/24     19.19.19.19     31/16027
  20.20.20.20/32   19.19.19.19     32/16028

R3#show ip bgp labels
   Network          Next Hop      In label/Out label
    2.2.2.2/32       20.3.5.5        18/nolabel
   3.3.3.3/32       0.0.0.0         imp-null/nolabel
   4.4.4.4/32       20.1.3.1        nolabel/27
   5.5.5.5/32       20.3.5.5        17/nolabel
   6.6.6.6/32       20.1.3.1        nolabel/30
   20.1.3.0/24      0.0.0.0         imp-null/nolabel
   20.2.5.0/24      20.3.5.5        19/nolabel
   20.3.5.0/24      0.0.0.0         imp-null/nolabel
   20.4.6.0/24      20.1.3.1        nolabel/28
   20.4.19.0/24     20.1.3.1        nolabel/29
   20.6.20.0/24     20.1.3.1        nolabel/31
   20.20.20.20/32   20.1.3.1        nolabel/32
```

From the CSC CE router R3's perspective, this route is being learned via IGP and has an LDP bound label for outbound traffic. However the inbound label is allocated via IPv4 Labeled Unicast BGP. R3 is "stitching" these two values together in the MPLS LFIB.

```
R3#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "isis", distance 115, metric 20, type level-2
  Redistributing via isis 1
  Advertised by bgp 100
  Last update from 20.3.5.5 on GigabitEthernet1.35, 00:40:07 ago
  Routing Descriptor Blocks:
  * 20.3.5.5, from 2.2.2.2, 00:40:07 ago, via GigabitEthernet1.35
      Route metric is 20, traffic share count is 1



R3#show mpls forwarding-table 2.2.2.2
Local   Outgoing    Prefix          Bytes Label Outgoing    Next Hop
Label   Label       or Tunnel Id    Switched    interface
18      16          2.2.2.2/32      18112       Gi1.35  20.3.5.5



R3#show ip cef 2.2.2.2/32 detail
2.2.2.2/32, epoch 2
  local label info: global/18
   nexthop 20.3.5.5 GigabitEthernet1.35 label 16
```

The next router, R5, is the Penultimate Hop, and will remove the topmost label for traffic going towards the 2.2.2.2/32 Loopback of R2.

```
R5#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "isis", distance 115, metric 10, type level-2
  Redistributing via isis 1
  Last update from 20.2.5.2 on GigabitEthernet1.25, 00:41:42 ago
  Routing Descriptor Blocks:
  * 20.2.5.2, from 2.2.2.2, 00:41:42 ago, via GigabitEthernet1.25
      Route metric is 10, traffic share count is 1

R5#show mpls forwarding-table 2.2.2.2
Local   Outgoing    Prefix          Bytes Label Outgoing    Next Hop
Label   Label       or Tunnel Id    Switched    interface
16      Pop Label   2.2.2.2/32      17713       Gi1.25  20.2.5.2

R5#show ip cef 2.2.2.2/32 detail
2.2.2.2/32, epoch 2
  local label info: global/16
   nexthop 20.2.5.2 GigabitEthernet1.25
```

Once the advertisements are completed end-to-end, R2 and XR2 should have an LSP built between their Loopback0 networks. Additionally, the output of the traceroute between them should indicate that an additional level of labels is used when traffic is transiting the Core Carrier network, as seen below.

CONTENTS

```
R9#traceroute vrf VPN_B 40.10.10.10 source loopback 40
Type escape sequence to abort.
Tracing the route to 40.10.10.10
VRF info: (vrf in name/id, vrf out name/id)
  1 40.9.20.20 4 msec 1 msec 1 msec
  2 20.6.20.6 [MPLS: Labels 19/23 Exp 0] 28 msec 20 msec 20 msec
  3 20.4.6.4 [MPLS: Labels 20/23 Exp 0] 24 msec 20 msec 27 msec
  4 20.4.19.19 [MPLS: Labels 16017/23 Exp 0] 31 msec 32 msec 148 msec
  5 10.8.19.8 [MPLS: Labels  17/19/23  Exp 0] 28 msec 32 msec 31 msec
  6 10.7.8.7 [MPLS: Labels  16/19/23  Exp 0] 31 msec 32 msec 32 msec
  7 10.1.7.1 [MPLS: Labels 19/23 Exp 0] 32 msec 32 msec 32 msec
  8 20.1.3.3 [MPLS: Labels 18/23 Exp 0] 32 msec 32 msec 32 msec
  9 20.3.5.5 [MPLS: Labels 16/23 Exp 0] 27 msec 31 msec 31 msec
 10 40.2.10.2 [MPLS: Label 23 Exp 0] 20 msec 24 msec 20 msec
 11 40.2.10.10 25 msec *  28 msec
```

These extra transport labels of 17 and 16 seen above are what hide the final customer traffic from the Core Carrier network. Without them the PHP process would happen one hop too soon, and a P router in the core of the topology would be exposed to a VPN Label that it does not know about. If MPLS TE were used anywhere in the network you would see more than three labels in the stack in the Core Carrier.

Visualize the design as a set of tunnels, representing the LSPs. There is a tunnel between R1 and XR1, this is the "core" tunnel - which is accomplished by establishing an LSP between them using IGP + LDP. Next, there is a second tunnel, between R2 and XR2, which rides on top of the R1-XR1 tunnel. The tunnel between R2 and XR2 is established by the L3VPN service that the Core Carrier is providing to the Customer Carrier, which unlike a traditional L3VPN, exchanges labels between PE and CE. Without the underlying tunnel between R1 and XR1, the tunnel between R2 and XR2 would not work, as core routers R7 and R8 have no knowledge of these devices. Now that the tunnel between R2 and XR2 is established, the Customer Carrier can provide a L3VPN service to its attached Customers (VPN_A and VPN_B). This hierarchy can expand even further, although it would not be very practical. For example, if R9 and R10 were PE routers and had CEs attached to them, an additional layer of L3VPN could be established to provide the attached CEs a VPN service - by creating yet another tunnel between R9 and R10, which would ride on top of the R2-XR2 tunnel, which in turn is riding on top of the R1-XR1 tunnel. In order to do this, label exchange would need to occur between R2 and R9/R10, and XR2 and R9/R10, just like its occurring between R1 and R3, and XR1 and R4.

Now let’s look at this topology with a common break in the control plane. Below the CSC CE R3 removes its MPLS Labeling capability under the IPv4 Unicast BGP process.

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router bgp 100
R3(config-router)#no neighbor 20.1.3.1 send-label
%BGP-5-ADJCHANGE: neighbor 20.1.3.1 Down Capability changed
%BGP_SESSION-5-ADJCHANGE: neighbor 20.1.3.1 IPv4 Unicast topology base removed from
session Capability changed
%BGP_LMM-6-MPLS_INIT: MPLS has been disabled for the BGP address-family IPv4
R3(config-router)#end
%BGP-5-ADJCHANGE: neighbor 20.1.3.1 Up
```

Once removed, R3 still learns IPv4 Unicast BGP routes from the Core Carrier, however it does not learn label values for them. The most important ones here are 2.2.2.2 and 20.20.20.20 (the 'tunnel' endpoints from the previous analogy), which need to be forwarded via an LSP that is reachable end to end within the Customer Carrier.

```
R3#show bgp ipv4 unicast
BGP table version is 43, 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
 *>  2.2.2.2/32       20.3.5.5                20         32768 i
 *>  3.3.3.3/32       0.0.0.0                  0         32768 i
 *>  4.4.4.4/32       20.1.3.1                               0 17819 17819 i
 *>  5.5.5.5/32       20.3.5.5                10         32768 i
 *>  6.6.6.6/32       20.1.3.1                               0 17819 17819 i
 *>  20.1.3.0/24      0.0.0.0                  0         32768 i
 *>  20.2.5.0/24      20.3.5.5                20         32768 i
 *>  20.3.5.0/24      0.0.0.0                  0         32768 i
 *>  20.4.6.0/24      20.1.3.1                               0 17819 17819 i
 *>  20.4.19.0/24     20.1.3.1                               0 17819 17819 i
 *>  20.6.20.0/24     20.1.3.1                               0 17819 17819 i
 *>  20.20.20.20/32   20.1.3.1                               0 17819 17819 i

R3#show bgp ipv4 unicast labels
   Network          Next Hop      In label/Out label
   2.2.2.2/32       20.3.5.5        nolabel/nolabel
   3.3.3.3/32       0.0.0.0         nolabel/nolabel
   4.4.4.4/32       20.1.3.1        nolabel/nolabel
   5.5.5.5/32       20.3.5.5        nolabel/nolabel
   6.6.6.6/32       20.1.3.1        nolabel/nolabel
   20.1.3.0/24      0.0.0.0         nolabel/nolabel
   20.2.5.0/24      20.3.5.5        nolabel/nolabel
   20.3.5.0/24      0.0.0.0         nolabel/nolabel
   20.4.6.0/24      20.1.3.1        nolabel/nolabel
   20.4.19.0/24     20.1.3.1        nolabel/nolabel
   20.6.20.0/24     20.1.3.1        nolabel/nolabel
   20.20.20.20/32   20.1.3.1        nolabel/nolabel
```

From the final customer’s point of view, this does not affect their control plane:

CONTENTS >

```
R9#show ip route vrf VPN_B rip

Routing Table: VPN_B
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

      40.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
R        40.2.10.0/24 [120/1] via 40.9.20.20, 00:00:05, GigabitEthernet1.920
R        40.10.10.10/32 [120/1] via 40.9.20.20, 00:00:05, GigabitEthernet1.920
```

However it does affect their data plane:

```
R9#ping vrf VPN_B 40.10.10.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 40.10.10.10, timeout is 2 seconds:
.....
 Success rate is 0 percent  (0/5)
```

Tracking this problem becomes very difficult because the MPLS Labels used need to be verified on a hop by hop basis. Furthermore it becomes more difficult because when the LSP is broken, a traceroute does not return any useful output:

```
RP/0/0/CPU0:XR2#traceroute vrf VPN_B 40.10.10.10
Wed Mar 21 23:13:54.107 UTC

Type escape sequence to abort.
Tracing the route to 40.10.10.10

 1  *   *   *
 2  *   *   *
 3  *   *   *
 4  *   *   *
<snip>
```

One way to troubleshoot this is to look at the specific label value that should be being used, and correlate this with the **debug mpls packet** output. However, this particular debug does not work properly in the devices used during this lab, the CSR1000v. Another alternative is to use the MPLS traceroute utilities on IOS and IOS-XR.

Notice that the packets sent to 20.20.20.20/32 are labeled until they reach R3, giving us a great indication of where they problem may be.

```
R2#traceroute mpls ipv4 20.20.20.20/32
more work needed here to demux the tfs subtlv and to display the right output

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface,  'B' - unlabeled output interface ,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 20.2.5.2 MRU 1500 [Labels: 21 Exp: 0]
L 1 20.2.5.5 MRU 1500 [Labels: 25 Exp: 0] 14 ms
 B  2 20.3.5.3 MRU 1500 [No Label] 2 ms
B 3 20.3.5.3 MRU 1500 [No Label] 2 ms
B 4 20.3.5.3 MRU 1500 [No Label] 2 ms
B 5 20.3.5.3 MRU 1500 [No Label] 9 ms
B 6 20.3.5.3 MRU 1500 [No Label] 21 ms
B 7 20.3.5.3 MRU 1500 [No Label] 22 ms
B 8 20.3.5.3 MRU 1500 [No Label] 19 ms
B 9 20.3.5.3 MRU 1500 [No Label] 22 ms
B 10 20.3.5.3 MRU 1500 [No Label] 21 ms
B 11 20.3.5.3 MRU 1500 [No Label] 21 ms
B 12 20.3.5.3 MRU 1500 [No Label] 22 ms
B 13 20.3.5.3 MRU 1500 [No Label] 21 ms
B 14 20.3.5.3 MRU 1500 [No Label] 22 ms
B 15 20.3.5.3 MRU 1500 [No Label] 21 ms
B 16 20.3.5.3 MRU 1500 [No Label] 22 ms
B 17 20.3.5.3 MRU 1500 [No Label] 21 ms
B 18 20.3.5.3 MRU 1500 [No Label] 22 ms
B 19 20.3.5.3 MRU 1500 [No Label] 21 ms
B 20 20.3.5.3 MRU 1500 [No Label] 21 ms
B 21 20.3.5.3 MRU 1500 [No Label] 20 ms
B 22 20.3.5.3 MRU 1500 [No Label] 21 ms
B 23 20.3.5.3 MRU 1500 [No Label] 22 ms
B 24 20.3.5.3 MRU 1500 [No Label] 21 ms
B 25 20.3.5.3 MRU 1500 [No Label] 22 ms
B 26 20.3.5.3 MRU 1500 [No Label] 21 ms
B 27 20.3.5.3 MRU 1500 [No Label] 21 ms
B 28 20.3.5.3 MRU 1500 [No Label] 22 ms
B 29 20.3.5.3 MRU 1500 [No Label] 21 ms
B 30 20.3.5.3 MRU 1500 [No Label] 22 ms
```

On IOS-XR, 'mpls oam' must be enabled in order for the MPLS traceroute utility to work. Notice that the hop to R3 shows "No Label".

CONTENTS

```
RP/0/0/CPU0:XR2#

RP/0/0/CPU0:XR2#conf t

Fri May 15 23:49:53.422 UTC

RP/0/0/CPU0:XR2(config)#mpls oam

RP/0/0/CPU0:XR2(config-oam)#commit

Fri May 15 23:49:59.012 UTC

RP/0/0/CPU0:XR2(config-oam)#end

RP/0/0/CPU0:XR2#traceroute mpls ipv4 2.2.2.2/32

Fri May 15 23:50:04.041 UTC


Tracing MPLS Label Switched Path to 2.2.2.2/32, timeout is 2 seconds


Codes: '!' - success, 'Q' - request not sent, '.' - timeout,

  'L' - labeled output interface, 'B' - unlabeled output interface,

  'D' - DS Map mismatch, 'F' - no FEC mapping,  'f' - FEC mismatch ,

  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,

  'P' - no rx intf label prot, 'p' - premature termination of LSP,

  'R' - transit router, 'I' - unknown upstream index,

  'X' - unknown return code, 'x' - return code 0


Type escape sequence to abort.


  0 20.6.20.20 MRU 1500 [Labels: 19 Exp: 0]

L 1 20.6.20.6 MRU 1500 [Labels: 20 Exp: 0] 0 ms

L 2 20.4.6.4 MRU 1500 [Labels: 16017 Exp: 0] 0 ms

. 3 *

. 4 *

. 5 *

. 6 *

 f  7 20.1.3.3 MRU 0 [ No Label ] 10 ms
```

With the MPLS Traceroute, we can quickly narrow down the problem scope and find that R3 does not have a label for XR2.

```
R3#show ip cef 20.20.20.20/32

20.20.20.20/32

  nexthop 20.1.3.1 GigabitEthernet1.13


R4#show ip cef 2.2.2.2

2.2.2.2/32

  nexthop 20.4.19.19 GigabitEthernet1.419 label 16017
```

Note that the BGP session between R2 and XR2 remains up even though the LSP is broken. IP transport between these devices is stable because the L3VPN service provided by the Core Carrier to the Customer carrier is not affected by the removal of labeling on R3. However, the end-to-end LSP between R2 and XR2 is broken, breaking the data-plane between the customers of the Customer Carrier.

This can be further verified with the MPLS Ping utility. Regular ICMP (IP reachability) works fine, but an MPLS ping does not.

CONTENTS

```
R2#ping 20.20.20.20 source loopback 0
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 2.2.2.2
 !!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/23/36 ms

R2#ping mpls ipv4 20.20.20.20/32 source 2.2.2.2
Sending 5, 72-byte MPLS Echos to Target FEC Stack TLV descriptor,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface,  'B' - unlabeled output interface ,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
 BBBBB
Success rate is 0 percent (0/5)
 Total Time Elapsed 21 ms
```

With correct labeling on the CSC CE to CSC PE link the debug output would look similar to the following:

CONTENTS

```
R3#config t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router bgp 100
R3(config-router)#address-family ipv4 unicast
R3(config-router-af)#neighbor 20.1.3.1 send-label
R3(config-router-af)#end
R3#
%BGP-5-ADJCHANGE: neighbor 20.1.3.1 Down Capability changed
%BGP_SESSION-5-ADJCHANGE: neighbor 20.1.3.1 IPv4 Unicast topology base removed from
session Capability changed
%BGP-5-ADJCHANGE: neighbor 20.1.3.1 Up
%SYS-5-CONFIG_I: Configured from console by console

R2#traceroute mpls ipv4 20.20.20.20/32
more work needed here to demux the tfs subtlv and to display the right output

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 20.2.5.2 MRU 1500 [Labels: 21 Exp: 0]
L 1 20.2.5.5 MRU 1500 [Labels: 25 Exp: 0] 15 ms
L 2 20.3.5.3 MRU 1500 [Labels: 32 Exp: 0] 3 ms
. 3 *
. 4 *
. 5 *
. 6 *
L 7 20.4.19.4 MRU 1500 [Labels: 18 Exp: 0] 17 ms
L 8 20.4.6.6 MRU 1500 [Labels: implicit-null Exp: 0] 12 ms
! 9 20.6.20.20 18 ms

RP/0/0/CPU0:XR2#traceroute mpls ipv4 2.2.2.2/32
Fri May 15 23:59:55.511 UTC

Tracing MPLS Label Switched Path to 2.2.2.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 20.6.20.20 MRU 1500 [Labels: 19 Exp: 0]
```

CONTENTS

```
L 1 20.6.20.6 MRU 1500 [Labels: 20 Exp: 0] 0 ms
L 2 20.4.6.4 MRU 1500 [Labels: 16017 Exp: 0] 0 ms
. 3 *
. 4 *
. 5 *
. 6 *
L 7 20.1.3.3 MRU 1500 [Labels: 16 Exp: 0] 10 ms
L 8 20.3.5.5 MRU 1500 [Labels: implicit-null Exp: 0] 10 ms
! 9 20.2.5.2 20 ms

R2#ping mpls ipv4 20.20.20.20/32 source 2.2.2.2
Sending 5, 72-byte MPLS Echos to Target FEC Stack TLV descriptor,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'l' - Label switched with FEC change, 'd' - see DDMAP for return code,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/20/23 ms
 Total Time Elapsed 104 ms

RP/0/0/CPU0:XR2#ping mpls ipv4 2.2.2.2/32 source 20.20.20.20
Sat May 16 00:32:02.259 UTC

Sending 5, 100-byte MPLS Echos to 2.2.2.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/14/20 ms
```

Another important part about this design, like in the previous Inter-AS MPLS L3VPN scenarios, is that IOS XR cannot label switch traffic towards a non /32 next-hop value. This means that XR1 must have the following route statically configured in the CSC VRF table:

```
RP/0/0/CPU0:XR1#sh run router static
Sat May 16 00:01:20.445 UTC
router static
 vrf CSC
  address-family ipv4 unicast
   20.4.19.4/32 GigabitEthernet0/0/0/0.419
  !
 !
!

RP/0/0/CPU0:XR1#show route vrf CSC static
Sat May 16 00:01:32.304 UTC

S   20.4.19.4/32 is directly connected, 01:29:53, GigabitEthernet0/0/0/0.419


RP/0/0/CPU0:XR1#show cef vrf CSC 20.20.20.20 detail
Sat May 16 00:02:31.600 UTC
20.20.20.20/32, version 27, internal 0x4004001 0x0 (ptr 0xa0edb9f4) [1], 0x0 (0x0), 0x208 (0xa13f6168)
 Updated May 15 22:32:49.139
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
  gateway array (0xa0d308b0) reference count 9, flags 0x4078, source rib (6), 0 backups
                [1 type 5 flags 0x48089 (0xa1410578) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
   via 20.4.19.4, 5 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0xa145fcf4 0x0]
    next hop 20.4.19.4 via 16016/0/21
     local label 16028
     next hop 20.4.19.4/32 Gi0/0/0/0.419 labels imposed {ImplNull 19}



    Load distribution: 0 (refcount 1)

    Hash  OK  Interface                 Address
    0     Y   Unknown                   16016/0
```

If this route is removed, traffic in the data plane fails.

```
R2#ping vrf VPN_B 40.9.9.9


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 40.9.9.9, timeout is 2 seconds:

!!!!!

 Success rate is 100 percent (5/5), round-trip min/avg/max = 8/11/16 ms


RP/0/0/CPU0:XR1#config t

Thu Mar 22 14:47:49.062 UTC

RP/0/0/CPU0:XR1(config)#no router static

RP/0/0/CPU0:XR1(config)#commit

Sat May 16 00:03:23.736 UTC

RP/0/0/CPU0:May 16 00:03:23.966 : config[65729]: %MGBL-CONFIG-6-DB_COMMIT : Configuration committed by user 'admin'. Use 'show configuration

 commit changes 1000000243' to view the changes.



R2#ping vrf VPN_B 40.9.9.9


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 40.9.9.9, timeout is 2 seconds:

.....

 Success rate is 0 percent (0/5)
```

Notice that without the static route, CEF considers this entry as "unresolved" and will drop traffic that hits this entry.

```
RP/0/0/CPU0:XR1#show cef vrf CSC 20.20.20.20 detail

Sat May 16 00:03:54.864 UTC

20.20.20.20/32, version 27, internal 0x4004001 0x0 (ptr 0xa0edb9f4) [1], 0x0 (0x0), 0x208 (0xa13f6168)

 Updated May 15 22:32:49.138

 Prefix Len 32, traffic index 0, precedence n/a, priority 3

  gateway array (0xa0d308b0) reference count 9, flags 0x407a, source rib (6), 0 backups

               [1 type 5 flags 0x140089 (0xa1410578) ext 0x0 (0x0)]

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

   via 20.4.19.4, 0 dependencies, recursive, bgp-ext [flags 0x6020]

    path-idx 0 NHID 0x0 [0xa0c1156c 0x0]

    unresolved

     local label 16028

     labels imposed {19}



    Load distribution: 0 (refcount 1)


    Hash  OK  Interface                 Address

    0     Y   Unknown                   drop
```

« 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) | MPLS L3VPN 6PE (/workbook/view/service-provider-v4/task/mpls-l3vpn-6pe-Mjg3Mg%3D%3D) »

› CONTENTS