MPLS L3 VPN with Static Routing

« Basic MPLS Tunnels (/workbook/view/service-provider-v4/task/basic-mpls-tunnels-Mjg1NA%3D%3D) | MPLS L3 VPN with RIPv2 (/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-ripv2-Mjg1Ng%3D%3D) »

Last updated: April 23, 2016

CONTENTS

Note:

Initial Configuration & Diagrams: Load the initial configuration files for the section named **OSPFv2 and LDP**, 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 **Base IPv4 Diagram** in order to complete this task.

Task

- Configure a VRF on R2 and XR1 as follows:
  - VRF Name: VPN_A
  - Route Distinguisher: 100:1
  - Route Target Import: 100:1
  - Route Target Export: 100:1
  - Assign the VRF to the links connecting to R1 and XR2 respectively.
- Configure routing for the VRF as follows:
  - R1 should have a default route pointing towards R2.
  - R2 should have a static route for 1.1.1.1/32 pointing towards R1.
  - XR1 should have a static route for 20.20.20.20/32 pointing towards XR2.
  - XR2 should have a default route pointing towards XR1.
- Configure BGP on R2 and XR1 as follows:
  - Use BGP AS 100.
  - R2 and XR1 should be iBGP peers for the VPNv4 Address Family.
  - Use their Loopback0 interfaces as the source of the BGP session.
  - Advertise the static routes towards R1 and XR2 into BGP on R2 and XR1 respectively.
- Once complete R1 and XR2 should have reachability to each other's Loopback0 interfaces when sourcing traffic from their own Loopback0 interfaces.

Configuration **Click to *collapse***

```
R1:
ip route 0.0.0.0 0.0.0.0 10.1.2.2


R2:
vrf definition VPN_A
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 !
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet1.12
 vrf forwarding VPN_A
 ip address 10.1.2.2 255.255.255.0
!
router bgp 100
 no bgp default ipv4-unicast
 neighbor 19.19.19.19 remote-as 100
 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 VPN_A
  network 1.1.1.1 mask 255.255.255.255
 exit-address-family
 !
 ip route vrf VPN_A 1.1.1.1 255.255.255.255 10.1.2.1


XR1:


vrf VPN_A
 address-family ipv4 unicast
  import route-target
   100:1
  !
  export route-target
   100:1
  !
 !
!
interface GigabitEthernet0/0/0/0.1920
 vrf VPN_A
 ipv4 address 10.19.20.19 255.255.255.0
!
router static
 vrf VPN_A
 address-family ipv4 unicast
   20.20.20.20/32 GigabitEthernet0/0/0/0.1920 10.19.20.20
  !
```

```
 !
!
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
   network 20.20.20.20/32
  !
 !
!


XR2:
router static
 address-family ipv4 unicast
  0.0.0.0/0 GigabitEthernet0/0/0/0.1920 10.19.20.19
 !
!
```

CONTENTS

Verification

show vrf detail is useful to quickly verify configured VRFs names, RDs, RT import and export policy, and assigned links.

```
R2#show vrf detail
VRF VPN_A (VRF Id = 3);  default RD 100:1 ; default VPNID <not set>
  New CLI format, supports multiple address-families
  Flags: 0x180C
  Interfaces:
    Gi1.12
Address family ipv4 unicast (Table ID = 0x3):
  Flags: 0x0
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1
  No import route-map
  No global export route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv6 unicast not active
Address family ipv4 multicast not active

RP/0/0/CPU0:XR1#show vrf VPN_A detail
Sun May  3 15:36:57.384 UTC

VRF VPN_A;  RD 100:1 ; VPN ID not set
VRF mode: Regular
Description not set
Interfaces:
   GigabitEthernet0/0/0/0.1920
Address family IPV4 Unicast
  Import VPN route-target communities:
    RT:100:1
  Export VPN route-target communities:
    RT:100:1
  No import route policy
  No export route policy
Address family IPV6 Unicast
  No import VPN route-target communities
  No export VPN route-target communities
  No import route policy
  No export route policy
```

Note that in IOS XR, once an interface is removed from the global routing table and assigned to a VRF table, it no longer appears in the **show ipv4 interface brief** output, as seen below. Instead, interfaces can be verified with the command **show ipv4 vrf all interface brief**.

```
RP/0/0/CPU0:XR1#show ip interface brief
Sun May  3 15:37:45.400 UTC

Interface                      IP-Address      Status          Protocol
Loopback0                      19.19.19.19     Up              Up
MgmtEth0/0/CPU0/0              unassigned      Up              Up
GigabitEthernet0/0/0/0         unassigned      Up              Up
GigabitEthernet0/0/0/0.519     20.5.19.19      Up              Up
GigabitEthernet0/0/0/0.619     20.6.19.19      Up              Up
```

```
RP/0/0/CPU0:XR1#show ipv4 vrf all interface brief
Sun May  3 15:38:16.118 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      19.19.19.19     Up              Up       default
MgmtEth0/0/CPU0/0              unassigned      Up              Up       default
GigabitEthernet0/0/0/0         unassigned      Up              Up       default
GigabitEthernet0/0/0/0.519     20.5.19.19      Up              Up       default
GigabitEthernet0/0/0/0.619     20.6.19.19      Up              Up       default
GigabitEthernet0/0/0/0.1920    10.19.20.19     Up              Up       VPN_A

RP/0/0/CPU0:XR1#show ipv4 vrf VPN_A interface brief
Sun May  3 15:38:44.146 UTC

Interface                      IP-Address      Status          Protocol
GigabitEthernet0/0/0/0.1920    10.19.20.19     Up              Up
```

In this example, the CE routers R1 and XR2 simply have default routes pointing to the PE routers R2 and XR1, resulting in one of the simplest MPLS L3VPN designs. From the CE routers perspective these are just normal static routes in the global routing table.

```
R1#show ip route
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 10.1.2.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.1.2.2
```

```
RP/0/3/CPU0:XR2#show route ipv4 static
Sun May  3 15:39:48.222 UTC

S*   0.0.0.0/0 [1/0] via 10.19.20.19, 00:04:48, GigabitEthernet0/0/0/0.1920
```

From the PE routers' R2 and XR1's perspective, their static routes to the customers exist in the VRF table.

```
R2#show ip route vrf VPN_A static


 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


      1.0.0.0/32 is subnetted, 1 subnets
 S        1.1.1.1 [1/0] via 10.1.2.1


RP/0/0/CPU0:XR1#show route vrf VPN_A ipv4 static
Sun May  3 15:40:49.918 UTC


 S    20.20.20.20/32 [1/0] via 10.19.20.20, 00:06:12, GigabitEthernet0/0/0/0.1920
```

R2 and XR1 then advertise these static routes into the VPNv4 BGP topology. In this case it is done with the network statement, but it could also be done with redistribution.

CONTENTS ›

```
R2#show bgp vpnv4 unicast all
BGP table version is 5, 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)
 *>   1.1.1.1/32       10.1.2.1                 0         32768 i
 *>i  20.20.20.20/32   19.19.19.19              0    100      0 i


RP/0/0/CPU0:XR1#show bgp vpnv4 unicast
Sun May  3 15:41:22.385 UTC
BGP router identifier 19.19.19.19, local AS number 100
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 5
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
Route Distinguisher: 100:1 (default for vrf VPN_A)
*>i 1.1.1.1/32         2.2.2.2                  0    100      0 i
*>  20.20.20.20/32     10.19.20.20              0         32768 i

Processed 2 prefixes, 2 paths
```

Note that R2 and XR1 use the Loopback0 interfaces of each other as the next-hop value for the VPNv4 learned routes, since this is the **update-source** of the iBGP session. In addition to the next-hop value, the VPN label derived from VPNv4 BGP can be seen in the below output. This is the label value that the PE routers use to find the final customer route in the VRF.

```
R2#show bgp vpnv4 unicast all 20.20.20.20/32
BGP routing table entry for 100:1:20.20.20.20/32, version 5
Paths: (1 available, best #1, table VPN_A)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    19.19.19.19  (metric 4) (via default) from 19.19.19.19 (19.19.19.19)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:1
       mpls labels  in/ out  nolabel/ 16000
      rx pathid: 0, tx pathid: 0x0


R2#show bgp vpnv4 unicast all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (VPN_A)
    1.1.1.1/32       10.1.2.1        31 /nolabel
    20.20.20.20/32   19.19.19.19     nolabel/ 16000


RP/0/0/CPU0:XR1#show bgp vrf VPN_A ipv4 unicast 1.1.1.1/32
Sun May  3 15:42:08.092 UTC
BGP routing table entry for 1.1.1.1/32, Route Distinguisher: 100:1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 5           5
Last Modified: May  3 15:34:43.451 for 00:07:24
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 4) from 2.2.2.2 (2.2.2.2)
      Received Label 31
      Origin IGP, metric 0, localpref 100, valid, internal, best, group-best, import-candidate, imported
      Received Path ID 0, Local Path ID 1, version 5
      Extended community: RT:100:1
      Source VRF: VPN_A, Source Route Distinguisher: 100:1


RP/0/0/CPU0:XR1#show bgp vrf VPN_A ipv4 unicast labels
Sun May  3 15:42:43.450 UTC
BGP VRF VPN_A, state: Active
BGP Route Distinguisher: 100:1
VRF ID: 0x60000005
BGP router identifier 19.19.19.19, local AS number 100
BGP table state: Active
Table ID: 0xe0000014   RD version: 5
BGP main routing table version 5


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
Route Distinguisher: 100:1 (default for vrf VPN_A)
*>i 1.1.1.1/32         2.2.2.2         31              nolabel
```

```
*> 20.20.20.20/32     10.19.20.20     nolabel         16000

Processed 2 prefixes, 2 paths
```

R2 and XR1 then combine this VPN label with the transport label used to reach each other's Loopback0 interfaces (the next-hop for the VPNv4 route). In this case the transport label is derived from OSPF + LDP. The transport label is used to tell the MPLS core what the exit PE is out of the network.

```
R2#show ip route 19.19.19.19

Routing entry for 19.19.19.19/32

  Known via "ospf 1", distance 110, metric 4, type intra area

  Last update from 20.2.4.4 on GigabitEthernet1.24, 00:21:54 ago

  Routing Descriptor Blocks:

    20.2.4.4, from 19.19.19.19, 00:21:54 ago, via GigabitEthernet1.24

      Route metric is 4, traffic share count is 1

  * 20.2.3.3, from 19.19.19.19, 00:21:54 ago, via GigabitEthernet1.23

      Route metric is 4, traffic share count is 1


R2#show mpls forwarding-table 19.19.19.19

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop

Label      Label      or Tunnel Id     Switched      interface

29          26         19.19.19.19/32   0             Gi1.23     20.2.3.3

            25         19.19.19.19/32   0             Gi1.24     20.2.4.4


RP/0/0/CPU0:XR1#show route ipv4 2.2.2.2

Sun May  3 15:44:12.624 UTC


Routing entry for 2.2.2.2/32

  Known via "ospf 1", distance 110, metric 4, type intra area

  Installed May  3 15:21:39.646 for 00:22:33

  Routing Descriptor Blocks

    20.5.19.5, from 2.2.2.2, via GigabitEthernet0/0/0/0.519

      Route metric is 4

    20.6.19.6, from 2.2.2.2, via GigabitEthernet0/0/0/0.619

      Route metric is 4

  No advertising protos.


RP/0/0/CPU0:XR1#show mpls forwarding prefix 2.2.2.2/32

Sun May  3 15:44:41.522 UTC

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes

Label  Label       or ID              Interface                    Switched

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

16005  18          2.2.2.2/32         Gi0/0/0/0.519 20.5.19.5       4454

       19          2.2.2.2/32         Gi0/0/0/0.619 20.6.19.6       0
```

The two of these together, the VPN label and the transport label, make up the full label stack that is imposed when the PE routers receive traffic from the CE. This can be verified in the CEF table on the PE routers. The below cef table output of R2 indicates that the VPN label is 16000, and the transport label is either 25 or 26, depending which interface the traffic is CEF switched to.

```
R2#show ip cef vrf VPN_A 20.20.20.20 detail
20.20.20.20/32, epoch 0, flags [rib defined all labels]
  recursive via 19.19.19.19 label 16000
    nexthop 20.2.3.3 GigabitEthernet1.23 label 26
    nexthop 20.2.4.4 GigabitEthernet1.24 label 25
```

In XR1's case, the label stack consists of 31 as the VPN label, and either 18 or 19 as the transport label.

```
RP/0/0/CPU0:XR1#show cef vrf VPN_A 1.1.1.1/32 detail
Sun May  3 15:46:07.026 UTC
1.1.1.1/32, version 11, internal 0x14004001 0x0 (ptr 0xa0edc8f4) [1], 0x0 (0x0), 0x208 (0xa13f6348)
 Updated May  3 15:34:43.533
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
  gateway array (0xa0d300b8) reference count 1, flags 0x4038, source rib (6), 0 backups
                [1 type 1 flags 0x48089 (0xa141044c) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
   via 2.2.2.2, 3 dependencies, recursive  [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa145e574 0x0]
    next hop VRF - 'default', table - 0xe0000000
     next hop 2.2.2.2 via 16005/0/21
      next hop 20.5.19.5/32 Gi0/0/0/0.519 labels imposed {18 31}
      next hop 20.6.19.6/32 Gi0/0/0/0.619 labels imposed {19 31}



    Load distribution: 0 (refcount 1)


    Hash  OK  Interface                 Address
    0     Y   Unknown                   16005/0
```

The full label stack can also be verified by looking at a traceroute output from the CE. Note that the transport label changes on a hop-by-hop basis, but the VPN label remains the same end-to-end. R4 is the Penultimate hop for R2, so the top label 17 is being removed for packets going to R2. Likewise on the other side R5 is the Penultimate hop for XR1, as the transport label 24 is being popped for traffic going towards XR1.

```
RP/0/0/CPU0:XR2#traceroute 1.1.1.1 source 20.20.20.20

Sun May  3 15:48:51.955 UTC


Type escape sequence to abort.

Tracing the route to 1.1.1.1


 1  10.19.20.19 0 msec  0 msec  0 msec

 2  20.5.19.5 [MPLS: Labels 18/31 Exp 0] 9 msec  0 msec  0 msec

 3  20.4.5.4 [MPLS: Labels 17/31 Exp 0] 0 msec  9 msec  0 msec

 4  10.1.2.2 [MPLS: Label 31 Exp 0] 0 msec  0 msec  0 msec

 5  10.1.2.1 0 msec  *  0 msec


R4#show mpls forwarding-table 2.2.2.2

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop

Label      Label      or Tunnel Id     Switched      interface

17          Pop Label   2.2.2.2/32       13237         Gi1.24     20.2.4.2


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 1 msec

  2 20.2.4.4 [MPLS: Labels 25/16000 Exp 0] 10 msec 6 msec 6 msec

  3 20.4.5.5 [MPLS: Labels 24/16000 Exp 0] 14 msec 8 msec 24 msec

  4 20.5.19.19 19 msec 14 msec 14 msec

  5 10.19.20.20 14 msec *  10 msec


R5#show mpls forwarding-table 19.19.19.19

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop

Label      Label      or Tunnel Id     Switched      interface

24          Pop Label   19.19.19.19/32   7188          Gi1.519    20.5.19.19
```

The final result is that even though the devices in the core, i.e. R3, R4, R5, & R6, do not have routes for the customer VRF VPN_A, they are able to transport label switched packets that go between the PE routers of R2 and XR1.

```
R3#show ip route 1.1.1.1
 % Network not in table
R3#show ip route 20.20.20.20
 % Subnet not in table


R4#show ip route 1.1.1.1
% Network not in table
R4#show ip route 20.20.20.20
% Subnet not in table


R5#show ip route 1.1.1.1
% Network not in table
R5#show ip route 20.20.20.20
% Subnet not in table


R6#show ip route 1.1.1.1
% Network not in table
R6#show ip route 20.20.20.20
% Subnet not in table
```

« Basic MPLS Tunnels (/workbook/view/service-provider-v4/task/basic-mpls-tunnels-Mjg1NA%3D%3D) | MPLS L3 VPN with RIPv2 (/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-ripv2-Mjg1Ng%3D%3D) »

CONTENTS