

# 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 L3 VPN with RIPv2

« [MPLS L3 VPN with Static Routing \(/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-static-routing-Mjg1NQ%3D%3D\)](/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-static-routing-Mjg1NQ%3D%3D) |

[MPLS L3 VPN with EIGRP \(/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-eigrp-Mjg1Nw%3D%3D\)](/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-eigrp-Mjg1Nw%3D%3D) »

Last updated: April 23, 2016

#### 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.](#)

CONTENTS >

### 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 RIPv2 routing for the VRF as follows:
  - Enable RIPv2 between R1 & R2.
  - Enable RIPv2 between XR1 & XR2.
  - Advertise the Loopback0 networks of R1 & XR2 into RIP.
- 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.
- Redistribute between BGP and the RIPv2 as follows:
  - RIPv2 routes learned by the PEs from the CEs should have their metric transparently redistributed via the BGP MED attribute.
  - The connected PE to CE links should appear with a metric of 1 on the remote CEs
- Once complete R1 and XR2 should have reachability to each other's Loopback0 interfaces and PE-CE links.

### Configuration [Click to collapse](#)

```
R1:
router rip
  version 2
  network 1.0.0.0
  network 10.0.0.0
  no auto-summary

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 rip
  !
  address-family ipv4 vrf VPN_A
    redistribute bgp 100 metric transparent
  network 10.0.0.0
  no auto-summary
  version 2
  exit-address-family
  !
router bgp 100
  bgp log-neighbor-changes
  no bgp default ipv4-unicast
  neighbor 19.19.19.19 remote-as 100
  neighbor 19.19.19.19 update-source Loopback0
  !
  address-family ipv4
  exit-address-family
  !
  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
    redistribute rip
  exit-address-family
  !

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  
!  
route-policy BGP_TO_RIP  
  if destination in (10.1.2.0/24) then  
    set rip-metric 1  
  else  
    pass  
  endif  
end-policy  
!  
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  
  !  
  !  
  !  
router rip  
vrf VPN_A  
  interface GigabitEthernet0/0/0/0.1920  
  !  
  redistribute bgp 100 route-policy BGP_TO_RIP  
  !  
  !  
end  
  
XR2:  
router rip  
  interface Loopback0  
  !  
  interface GigabitEthernet0/0/0/0.1920  
  !  
  !
```

VRF aware RIP routing uses one global process, with sub-processes on a per VRF basis. When routes are redistributed from a VRF aware RIP process into BGP, the RIP metric is automatically copied into the BGP MED field. This can be seen in the BGP output below.

```
R2#show bgp vpnv4 unicast all
BGP table version is 23, 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	1		32768	?
*> 10.1.2.0/24	0.0.0.0	0		32768	?
*>i 10.19.20.0/24	19.19.19.19	0	100	0	?
*>i 20.20.20.20/32	19.19.19.19	1	100	0	?

```
RP/0/0/CPU0:XR1#show bgp vpnv4 unicast
Sun May 3 16:35:30.493 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 25
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
```

Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 100:1 (default for vrf VPN_A)					
*>i1.1.1.1/32	2.2.2.2	1	100	0	?
*>i10.1.2.0/24	2.2.2.2	0	100	0	?
*> 10.19.20.0/24	0.0.0.0	0		32768	?
*> 20.20.20.20/32	10.19.20.20	1		32768	?

Processed 4 prefixes, 4 paths

CONTENTS

This can be further verified if an offset list is applied to the RIP process to modify the metric values, as seen below.

```

R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router rip
R1(config-router)#offset-list 0 out 7 Gig1.12
R1(config-router)#end
%SYS-5-CONFIG_I: Configured from console by console
R1#clear ip route *
R1#

R2#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

1.0.0.0/32 is subnetted, 1 subnets
R       1.1.1.1 [120/8] via 10.1.2.1, 00:00:10, GigabitEthernet1.12

```

The new RIP metric of 8 is copied into the BGP MED of the VPNv4 route.

```

R2#show bgp vpnv4 unicast all 1.1.1.1/32
BGP routing table entry for 100:1:1.1.1/32, version 24
Paths: (1 available, best #1, table VPN_A)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
  10.1.2.1 (via vrf VPN_A) from 0.0.0.0 (2.2.2.2)
    Origin incomplete, metric 8, localpref 100, weight 32768, valid, sourced, best
    Extended Community: RT:100:1
    mpls labels in/out 43/nolabel
    rx pathid: 0, tx pathid: 0x0

```

When BGP is redistributed back into RIP on the remote side, the BGP MED is copied back into the RIP metric.

```
RP/0/0/CPU0:XR1#show bgp vpnv4 unicast vrf VPN_A 1.1.1.1/32
Sun May  3 16:38:10.922 UTC
BGP routing table entry for 1.1.1.1/32, Route Distinguisher: 100:1
Versions:
  Process          bRIB/RIB  SendTblVer
  Speaker          26       26
Last Modified: May  3 16:36:37.451 for 00:01:33
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 43
  Origin incomplete, metric 8, localpref 100, valid, internal, best, group-best, import-candidate, imported
  Received Path ID 0, Local Path ID 1, version 26
  Extended community: RT:100:1
  Source VRF: VPN_A, Source Route Distinguisher: 100:1
```

```
RP/0/0/CPU0:XR1#show rip vrf VPN_A database
Sun May  3 16:38:50.439 UTC
Routes held in RIP's topology database:
1.1.1.1/32
  [9] distance: 200   redistributed
1.0.0.0/8   auto-summary
10.1.2.0/24
  [1] distance: 200   redistributed
10.19.20.0/24
  [0]   directly connected, GigabitEthernet0/0/0/0.1920
10.0.0.0/8   auto-summary
20.20.20.20/32
  [1] via 10.19.20.20, next hop 10.19.20.20, Uptime: 5s, GigabitEthernet0/0/0/0.1920
20.0.0.0/8   auto-summary
```

The final result is the remote CE router sees the RIP metric maintained and incremented once across the MPLS VPN network.

```
RP/0/3/CPU0:XR2#show route ipv4 rip
Sun May  3 16:39:29.967 UTC

R   1.1.1.1/32 [120/9] via 10.19.20.19, 00:02:53, GigabitEthernet0/0/0/0.1920
R   10.1.2.0/24 [120/1] via 10.19.20.19, 00:05:11, GigabitEthernet0/0/0/0.1920
```

One potential “bug” in this configuration is that if the BGP MED is zero, some version of IOS XR do not redistribute the route from BGP into the RIP database unless the metric is manually set. This is why the additional Routing Policy Language is used to match the prefix connected between R1 and R2, and manually set the metric as the route is redistributed into RIP. With the RPL policy applied, the route appears in the RIP database of XR1, and hence the routing table of XR2. Note that this bug does not appear to be present in the version of IOS-XR used during these labs - IOS-XR 5.2.0.

```
RP/0/0/CPU0:XR1#show run router rip
Sun May  3 16:40:23.943 UTC
router rip
vrf VPN_A
  interface GigabitEthernet0/0/0.1920
  !
  redistribute bgp 100 route-policy BGP_TO_RIP
  !
!
```

---

```
RP/0/0/CPU0:XR1#show rpl
Sun May  3 16:40:45.811 UTC
route-policy BGP_TO_RIP
  if destination in (10.1.2.0/24) then
    set rip-metric 1
  else
    pass
  endif
end-policy
!
```

```
RP/0/0/CPU0:XR1#show rip vrf VPN_A database 10.1.2.0/24
Sun May  3 16:41:07.430 UTC
10.1.2.0/24
  [ 1 ] distance: 200   redistributed
```

If the Routing Policy is removed, the route continues being advertised on the current version of IOS-XR. On earlier versions of IOS-XR, such as 3.9, the removal of the Routing Policy would prevent the route from being redistributed.

```
RP/0/0/CPU0:XR1#config t
Sun May  3 16:41:30.308 UTC
RP/0/0/CPU0:XR1(config)#router rip
RP/0/0/CPU0:XR1(config-rip)# vrf VPN_A
RP/0/0/CPU0:XR1(config-rip-vrf)#no redistribute bgp 100 route-policy BGP_TO_RIP
RP/0/0/CPU0:XR1(config-rip-vrf)#redistribute bgp 100
RP/0/0/CPU0:XR1(config-rip-vrf)#show config
Sun May  3 16:42:08.056 UTC
Building configuration...
!! IOS XR Configuration 5.2.0
router rip
```

```
vrf VPN_A
  no redistribute bgp 100 route-policy BGP_TO_RIP
  redistribute bgp 100
!
!
end
```

```
RP/0/0/CPU0:XR1(config-rip-vrf)#commit
Sun May  3 16:42:39.674 UTC
RP/0/0/CPU0:XR1(config-rip-vrf)#end
```

```
RP/0/0/CPU0:XR1#show rip vrf VPN_A database 10.1.2.0/24
Sun May  3 16:52:13.694 UTC
```

```
10.1.2.0/24
  [1] distance: 200   redistributed
```

```
RP/0/3/CPU0:XR2#show route ipv4 rip
Sun May  3 16:52:46.202 UTC
```

```
R   1.1.1.1/32 [120/9] via 10.19.20.19, 00:05:12, GigabitEthernet0/0/0.1920
R   10.1.2.0/24 [120/1] via 10.19.20.19, 00:05:12, GigabitEthernet0/0/0.1920
```

An alternate fix to this would be to manually set the BGP MED on the remote side as redistribution from RIP to BGP is occurring, such as the following.

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#route-map RIP_TO_BGP
R2(config-route-map)#set metric 5
R2(config-route-map)#router bgp 100
R2(config-router)#address-family ipv4 vrf VPN_A
R2(config-router-af)#redistribute rip route-map RIP_TO_BGP
R2(config-router-af)#end
R2#
R2#
%SYS-5-CONFIG_I: Configured from console by console
```

```
R2#
R2#show bgp vpnv4 unicast all
BGP table version is 38, 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	5		32768	?
*> 10.1.2.0/24	0.0.0.0	5		32768	?
*>i 10.19.20.0/24	19.19.19.19	0	100	0	?
*>i 20.20.20.0/32	19.19.19.19	1	100	0	?

```
RP/0/3/CPU0:XR2#show route ipv4 rip
Sun May 3 16:54:38.555 UTC
```

```
R 1.1.1.1/32 [120/6] via 10.19.20.19, 00:00:35, GigabitEthernet0/0/0.1920
R 10.1.2.0/24 [120/6] via 10.19.20.19, 00:00:35, GigabitEthernet0/0/0.1920
```

In either case the final result should be that the customer sites have full reachability to each other.

```
R1#ping 20.20.20.20
```

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

```
Sending 5, 100-byte ICMP Echos to 20.20.20.20, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/12/18 ms
```

```
R1#ping 10.19.20.20
```

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

```
Sending 5, 100-byte ICMP Echos to 10.19.20.20, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/9/12 ms
```

```
R1#traceroute 10.19.20.20
```

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

```
Tracing the route to 10.19.20.20
```

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

- ```

 1 10.1.2.2 1 msec 2 msec 1 msec
 2 20.2.3.3 [MPLS: Labels 26/16000 Exp 0] 9 msec 7 msec 4 msec
 3 20.3.6.6 [MPLS: Labels 25/16000 Exp 0] 8 msec 14 msec 22 msec
 4 20.6.19.19 28 msec 8 msec 8 msec
 5 10.19.20.20 16 msec * 9 msec

```

```
RP/0/3/CPU0:XR2#ping 1.1.1.1
```

```
Sun May 3 16:55:24.091 UTC
```

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

```
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
```

```
RP/0/3/CPU0:XR2#ping 10.1.2.1
```

```
Sun May 3 16:55:29.091 UTC
```

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

```
Sending 5, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
```

```
RP/0/0/CPU0:XR2#traceroute 10.1.2.1
```

```
Sun May 3 17:02:16.753 UTC
```

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

```
Tracing the route to 10.1.2.1
```

- ```

 1 10.19.20.19 0 msec 0 msec 0 msec
 2 20.5.19.5 [MPLS: Labels 18/44 Exp 0] 0 msec 0 msec 0 msec
 3 20.4.5.4 [MPLS: Labels 17/44 Exp 0] 0 msec 0 msec 9 msec
 4 10.1.2.2 0 msec 0 msec 9 msec
 5 10.1.2.1 0 msec * 0 msec

```

Inside the core of the network this traffic is label switched towards the Loopback0 interfaces of the PE routers, R2 and XR1.

```
R5#show mpls forwarding-table 2.2.2.2
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing interface	Next Hop
18	17	2.2.2.2/32	32743	Gi1.45	20.4.5.4

```
R6#show mpls forwarding-table 19.19.19.19
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing interface	Next Hop
25	Pop Label	19.19.19.19/32	28624	Gi1.619	20.6.19.19

The inner label values of 1600 and 44 come from the origination of the VPNv4 routes on R2 and XR1.

```
R2#show bgp vpnv4 unicast all 10.19.20.0
```

BGP routing table entry for 100:1:10.19.20.0/24, version 34

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

Extended Community: RT:100:1

mpls labels in/out nlabel/16000

rx pathid: 0, tx pathid: 0x0

```
RP/0/0/CPU0:XR1#show bgp vpnv4 unicast vrf VPN_A 10.1.2.0/24
```

Sun May 3 16:58:56.027 UTC

BGP routing table entry for 10.1.2.0/24, Route Distinguisher: 100:1

Versions:

Process	bRIB/RIB	SendTblVer
Speaker	42	42

Last Modified: May 3 16:54:02.451 for 00:04:53

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 44

Origin incomplete, metric 5, localpref 100, valid, internal, best, group-best, import-candidate, imported

Received Path ID 0, Local Path ID 1, version 42

Extended community: RT:100:1

Source VRF: VPN\_A, Source Route Distinguisher: 100:1

« MPLS L3 VPN with Static Routing (/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-static-routing-Mjg1NQ%3D%3D) | MPLS L3 VPN with EIGRP (/workbook/view/service-provider-v4/task/mpls-l3-vpn-with-eigrp-Mjg1Nw%3D%3D) »