MPLS L3 VPN VPNv4 Route Reflection w/ IOS XR

« MPLS L3 VPN VPNv4 Route Reflection (/workbook/view/service-provider-v4/task/mpls-l3-vpn-vpnv4-route-reflection-Mjg2Mg%3D%3D) | MPLS L3 VPN and OSPF Sham-Links (/workbook/view/service-provider-v4/task/mpls-l3-vpn-and-ospf-sham-links-Mjg2NA%3D%3D) »

Last updated: April 23, 2016

CONTENTS

Note:

Initial Configuration & Diagrams: Load the initial configuration files for the section named **Multisite L3VPN**, 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 **Multisite L3VPN Diagram** in order to complete this task.

Task

- Configure a VRF on R2 and R5 as follows:
  - VRF Name: VPN_A
  - Route Distinguisher: 100:1
  - Route Target Import: 100:1
  - Route Target Export: 100:1
  - Assign the VRF on R2 and R5 to the links connecting to R1 and R8 respectively.
- Configure a VRF on R4 and XR1 as follows:
  - VRF Name: VPN_B
  - Route Distinguisher: 100:2
  - Route Target Import: 100:2
  - Route Target Export: 100:2
  - Assign the VRF on R4 and XR1 to the links connecting to R7 and XR2 respectively.
- Configure EIGRP AS 1 on the links between the PE to CEs, and advertise the CEs' Loopback0 networks into EIGRP.
- Configure BGP on R2, R4, R5, and XR1 as follows:
  - Use BGP AS 100.
  - All devices should peer with XR1 via iBGP for the VPNv4 Address Family.
  - XR1 should be the VPNv4 Route Reflector for these three clients.
  - Use their Loopback0 interfaces as the source of the BGP session.
  - Redistribute between BGP and the VRF aware EIGRP process on the PE routers.
- Once complete R1 should have reachability to all of R8's networks, and vice-versa, and XR2 should have reachability to all of R7's networks, and vice-versa.

Configuration Click to *collapse*

```
R1:
router eigrp 1
 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 eigrp 65535
 !
 address-family ipv4 vrf VPN_A
  autonomous-system 1
  redistribute bgp 100
  network 10.0.0.0
  no auto-summary
 exit-address-family
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 19.19.19.19 remote-as 100
 neighbor 19.19.19.19 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  no auto-summary
 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
  no synchronization
  redistribute eigrp 1
  exit-address-family
!


R4:
vrf definition VPN_B
 rd 100:2
```

```
 route-target export 100:2
 route-target import 100:2
 !
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet1.47
 vrf forwarding VPN_B
 ip address 10.4.7.4 255.255.255.0
!
router eigrp 65535
 !
 address-family ipv4 vrf VPN_B
  autonomous-system 1
  redistribute bgp 100
  network 10.0.0.0
  no auto-summary
 exit-address-family
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 19.19.19.19 remote-as 100
 neighbor 19.19.19.19 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  no auto-summary
 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_B
  no synchronization
  redistribute eigrp 1
 exit-address-family
!


R5:
vrf definition VPN_A
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 !
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet1.58
 vrf forwarding VPN_A
 ip address 10.5.8.5 255.255.255.0
!
```

CONTENTS ›

```
router eigrp 65535
 !
 address-family ipv4 vrf VPN_A
  autonomous-system 1
  redistribute bgp 100
  network 10.0.0.0
  no auto-summary
 exit-address-family
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 19.19.19.19 remote-as 100
 neighbor 19.19.19.19 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  no auto-summary
 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
  no synchronization
  redistribute eigrp 1
 exit-address-family
!


R7:
router eigrp 1
 network 7.0.0.0
 network 10.0.0.0
 no auto-summary


R8:
router eigrp 1
 network 8.0.0.0
 network 10.0.0.0
 no auto-summary


XR1:
vrf VPN_B
 address-family ipv4 unicast
  import route-target
   100:2
  !
  export route-target
   100:2
  !
 !
!
```

CONTENTS

```
interface GigabitEthernet0/0/0/0.1920
 vrf VPN_B
 no ipv4 address
 ipv4 address 10.19.20.19 255.255.255.0

!
router bgp 100
 address-family vpnv4 unicast
 !
 neighbor 2.2.2.2
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 neighbor 4.4.4.4
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
    route-reflector-client
  !
 !
 neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 vrf VPN_B
  rd 100:2
  address-family ipv4 unicast
   redistribute eigrp 1
  !
 !
!
router eigrp 65535
  vrf VPN_B
  address-family ipv4
   no auto-summary
   redistribute bgp 100
   autonomous-system 1
   interface GigabitEthernet0/0/0/0.1920
   !
  !
 !
!

XR2:
router eigrp 1
  address-family ipv4
   no auto-summary
   interface Loopback0
```

```
  !
  interface GigabitEthernet0/0/0/0.1920
 !
!
!
```

Verification

This example is similar to the previous one which used regular IOS and the VPNv4 Route Reflector. In this case XR1 peers with all other PE routers, and is a Route Reflector for the VPNv4 address family. This means that XR1 will be receiving all VPNv4 routes, regardless if it has a local VRF configured with a matching import policy, as seen below.

```
RP/0/0/CPU0:XR1#show bgp vpnv4 unicast
Thu May  7 23:38:38.838 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 11
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
*>i1.1.1.1/32       2.2.2.2         156160     100      0   ?
*>i8.8.8.8/32       5.5.5.5         156160     100      0   ?
*>i10.1.2.0/24      2.2.2.2              0     100      0   ?
*>i10.5.8.0/24      5.5.5.5              0     100      0   ?
Route Distinguisher: 100:2 (default for vrf VPN_B)
*>i7.7.7.7/32       4.4.4.4         156160     100      0   ?
*>i10.4.7.0/24      4.4.4.4              0     100      0   ?
*> 10.19.20.0/24    0.0.0.0              0          32768   ?
*> 20.20.20.20/32   10.19.20.20     131584          32768   ?
Processed 8 prefixes, 8 paths
RP/0/0/CPU0:XR1#
```

Note that the routes with the Route Distinguisher 100:1 do not have a VRF or routing table associated with them, but they are still kept in the BGP RIB and can be advertised.

```
RP/0/0/CPU0:XR1#show bgp vpnv4 unicast rd 100:1 1.1.1.1/32
Thu May  7 23:39:40.504 UTC
BGP routing table entry for 1.1.1.1/32, Route Distinguisher: 100:1
Versions:
 Process           bRIB/RIB  SendTblVer
 Speaker                 3           3
Last Modified: May  7 23:38:21.451 for 00:01:19
Paths: (1 available, best #1)
 Advertised to update-groups (with more than one peer):
   0.2
 Path #1: Received by speaker 0
 Advertised to update-groups (with more than one peer):
   0.2
 Local, (Received from a RR-client)
   2.2.2.2 (metric 4) from 2.2.2.2 (2.2.2.2)
     Received Label 18
     Origin incomplete, metric 130816, localpref 100, valid, internal, best, group-best, import-candidate, not-in-vrf
     Received Path ID 0, Local Path ID 1, version 3
     Extended community: COST:128:128:130816 EIGRP route-info:0x8000:0 EIGRP AD:1:128256 EIGRP RHB:255:1:2560 EIGRP LM:0xff:1:1500 0x8806:0x
00:0x00:0x01:0x01:0x01:0x01 RT:100:1
```

By default all routes are then advertised to all peers, and it is up to them to determine which ones they want or don't want.

```
RP/0/0/CPU0:XR1#show bgp vpnv4 unicast neighbors 5.5.5.5 advertised-routes
Thu May  7 23:45:28.450 UTC
Network            Next Hop        From            AS Path
Route Distinguisher: 100:1
1.1.1.1/32         2.2.2.2         2.2.2.2         ?
8.8.8.8/32         19.19.19.19     5.5.5.5         ?
10.1.2.0/24        2.2.2.2         2.2.2.2         ?
10.5.8.0/24        19.19.19.19     5.5.5.5         ?
Route Distinguisher: 100:2
7.7.7.7/32         4.4.4.4         4.4.4.4         ?
10.4.7.0/24        4.4.4.4         4.4.4.4         ?
10.19.20.0/24      19.19.19.19     Local           ?

Processed 7 prefixes, 7 paths
```

IOS XR VPNv4 Route Reflectors can also be configured to selectively accept VPNv4 routes based on their route targets by using the command **retain route-target route-policy *route-policy-name***. This could be used in a design where one RR services a certain set of PEs, while another RR services a separate set of PEs.

For example if we wanted XR1 to not accept VPNv4 routes that have the Route Target 100:1, this could be implemented as follows:

```
route-policy FILTER_ON_RT
    if extcommunity rt matches-any (100:1) then
        drop
    else
        pass
    endif
end-policy
!
router bgp 100
 address-family vpnv4 unicast
  retain route-target route-policy FILTER_ON_RT
   !
  !
```

Now XR1 denies VPNv4 routes that have the Route Target 100:1.

```
RP/0/0/CPU0:XR1#debug bgp update vpnv4 unicast in
Thu May  7 23:55:40.139 UTC

RP/0/0/CPU0:XR1#clear bgp vpnv4 unicast 2.2.2.2 soft in
Thu May  7 23:55:45.008 UTC

RP/0/0/CPU0:XR1#RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr]: UPDATE from 2.2.2.2 contains nh 2.2.2.2/32, gw_afi 0, flags 0x0,
 nlri_afi 4
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr]: NH-Validate-Create: addr=2.2.2.2/32, len=12, nlriafi=4, nbr=2.2.2.2, gwafi=0, gw
len=4, gwaddrlen=32::: nhout=0x10906b80, validity=1, attrwdrflags=0x00000000
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr]: --bgp4_rcv_attributes--: END: nbr=2.2.2.2:: msg=0x10037cb0/139, updlen=120, attr
bl=0x10037cc7/116, ipv4reachlen=0, msginpath=0x3ef0bf0, asloopcheck=1, attrwdrfl=0x00000000:: samecluster=0, local_as_prepended=0, attr_wdr_
flags 0x00000000, myascount=0:: rcvdata=0x10037d3b/0, errptr=0x10037d00/59
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u): Received UPDATE from 2.2.2.2 with attributes:
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u): nexthop 2.2.2.2/32, origin ?, localpref 100, metric 130816, extended co
mmunity COST:128:128:130816 EIGRP route-info:0x8000:0 EIGRP AD:1:128256 EIGRP RHB:255:1:2560 EIGRP LM:0xff:1:1500 0x8806:0x00:0x00:0x01:0x0
1:0x01:0x01 RT:100:1
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u): Received prefix 2ASN:100:1:1.1.1.1/32 (path ID: none) with MPLS label 18
 from neighbor 2.2.2.2
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u):  Prefix 2ASN:100:1:1.1.1.1/32 (path ID: none)  received from 2.2.2.2 DEN
IED RT extended community is not imported locally
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr]: UPDATE from 2.2.2.2 contains nh 2.2.2.2/32, gw_afi 0, flags 0x0, nlri_afi 4
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr]: NH-Validate-Create: addr=2.2.2.2/32, len=12, nlriafi=4, nbr=2.2.2.2, gwafi=0, gw
len=4, gwaddrlen=32::: nhout=0x10906b80, validity=1, attrwdrflags=0x00000000
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr]: --bgp4_rcv_attributes--: END: nbr=2.2.2.2:: msg=0x10037e48/138, updlen=119, attr
bl=0x10037e5f/115, ipv4reachlen=0, msginpath=0x3ef0bf0, asloopcheck=1, attrwdrfl=0x00000000:: samecluster=0, local_as_prepended=0, attr_wdr_
flags 0x00000000, myascount=0:: rcvdata=0x10037ed2/0, errptr=0x10037e97/59
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u): Received UPDATE from 2.2.2.2 with attributes:
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u): nexthop 2.2.2.2/32, origin ?, localpref 100, metric 0, extended communi
ty COST:128:128:2816 EIGRP route-info:0x8000:0 EIGRP AD:1:256 EIGRP RHB:255:0:2560 EIGRP LM:0xff:1:1500 0x8806:0x00:0x00:0x0a:0x01:0x02:0x02
 RT:100:1
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u): Received prefix 2ASN:100:1:10.1.2.0/24 (path ID: none) with MPLS label 2
7 from neighbor 2.2.2.2
RP/0/0/CPU0:May  7 23:55:45.108 : bgp[1047]: [default-rtr] (vpn4u):  Prefix 2ASN:100:1:10.1.2.0/24 (path ID: none)  received from 2.2.2.2 DE
NIED RT extended community is not imported locally

RP/0/0/CPU0:XR1#show bgp vpnv4 unicast
Thu May  7 23:57:55.049 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 23
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: 100:2 (default for vrf VPN_B)
*>i7.7.7.7/32       4.4.4.4         156160      100        0    ?
*>i10.4.7.0/24      4.4.4.4              0      100        0    ?
*> 10.19.20.0/24    0.0.0.0              0             32768    ?
```

```
*> 20.20.20.20/32   10.19.20.20     131584            32768    ?


Processed 4 prefixes, 4 paths
```

The normal behavior for a VPNv4 Route Reflector is the **retain route-target all**, which means that routes are not filtered based on their RT values.

Another common configuration in IOS XR for Route Reflection would be to group the client peers into a **neighbor-group** to simplify the configuration and its inheritance. This feature is similar to the **peer-group** in regular IOS. An configuration similar to this example, but with the usage of neighbor groups, could read as follows.

```
RP/0/0/CPU0:XR1#sh run router bgp
Fri Mar 9 23:39:18.919 UTC
router bgp 100
 address-family vpnv4 unicast
 !
 neighbor-group VPNv4_CLIENTS
  remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
   route-reflector-client
  !
 !
 neighbor 2.2.2.2
  use neighbor-group VPNv4_CLIENTS
 !
 neighbor 4.4.4.4
  use neighbor-group VPNv4_CLIENTS
 !
 neighbor 5.5.5.5
  use neighbor-group VPNv4_CLIENTS
 !
 vrf VPN_B
  rd 100:2
  address-family ipv4 unicast
   redistribute eigrp 1
  !
 !
!
```

The end result of either of these configurations is the same, that XR1 receives routes from all PEs and reflects them back. The final verification of this design would again be to test connectivity between the customer sites, as follows.

CONTENTS ›

```
RP/0/0/CPU0:XR2#show route ipv4 eigrp

Fri May  8 00:01:37.084 UTC


D    7.7.7.7/32 [90/2575360] via 10.19.20.19, 00:02:13, GigabitEthernet0/0/0/0.1920

D    10.4.7.0/24 [90/15360] via 10.19.20.19, 00:02:13, GigabitEthernet0/0/0/0.1920


RP/0/0/CPU0:XR2#ping 7.7.7.7

Fri May  8 00:01:55.923 UTC

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 7.7.7.7, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/7/9 ms
```

```
R8#sh ip route eigrp

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

       E1 - OSPF external type 1, E2 - OSPF external type 2

       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

       ia - IS-IS inter area, * - candidate default, U - per-user static route

       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

       a - application route

       + - replicated route, % - next hop override


Gateway of last resort is not set


      8.0.0.0/32 is subnetted, 1 subnets

D        8.8.8.8 [90/131072] via 10.1.2.2, 00:00:47, GigabitEthernet1.12

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

D        10.5.8.0/24 [90/3072] via 10.1.2.2, 00:00:47, GigabitEthernet1.12


R8#ping 1.1.1.1


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 = 1/2/4 ms
```

CONTENTS ›

« MPLS L3 VPN VPNv4 Route Reflection (/workbook/view/service-provider-v4/task/mpls-l3-vpn-vpnv4-route-reflection-Mjg2Mg%3D%3D) | MPLS L3 VPN and OSPF Sham-Links (/workbook/view/service-provider-v4/task/mpls-l3-vpn-and-ospf-sham-links-Mjg2NA%3D%3D) »