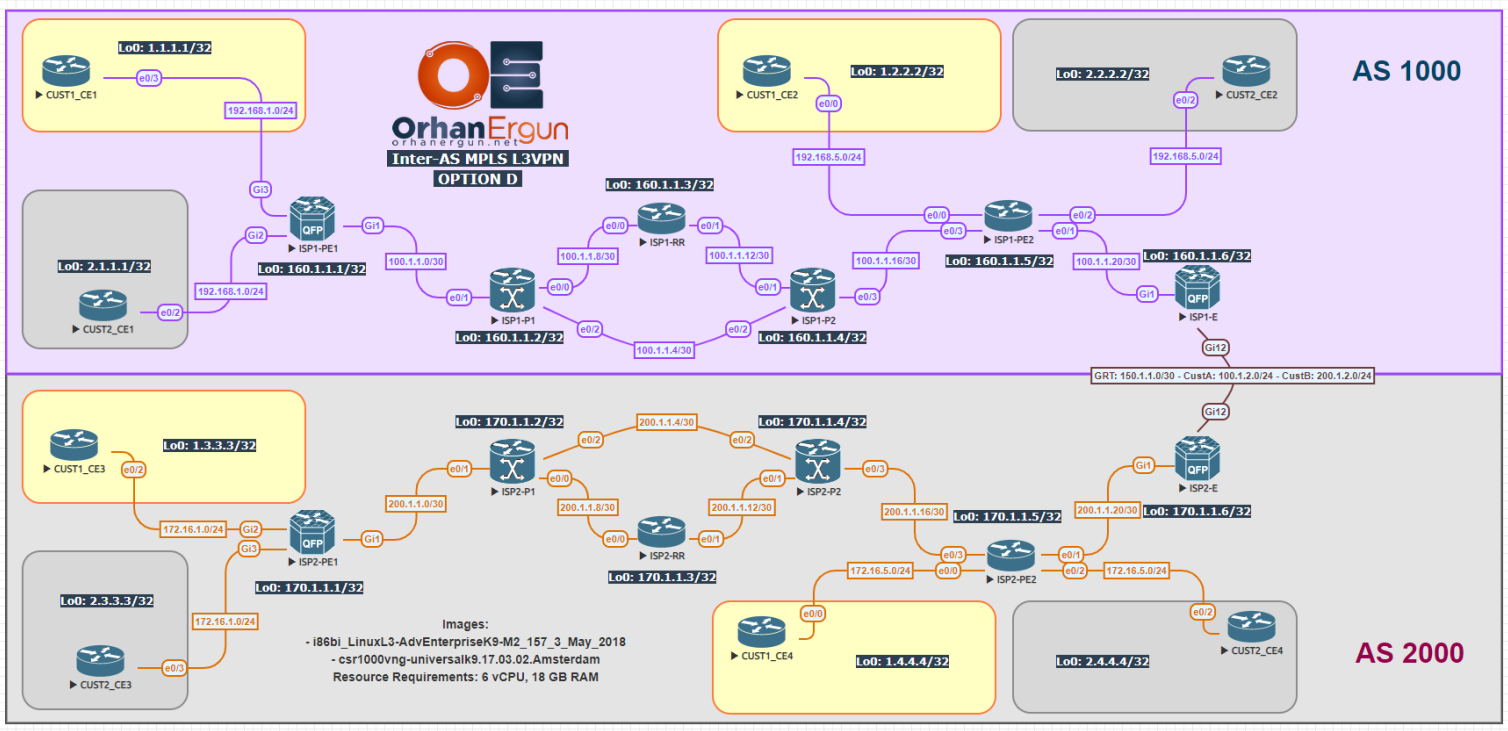
Inter-AS MPLS L3VPN Option D

Topology:



BGP AF/SAF: VPNv4 Unicast

PE-CE Routing Protocols:

- OSPF (Cust1)
- EIGRP Named-Mode (Cust2)

Task 01:

- Provide MPLS L3VPN service to the customers (Cust1 PE-CE: OSPF, Cust2 PE-CE: EIGRP)
- ISP1-RR and ISP2-RR should be configured as the Route-Reflectors

Solution:

You are already familiar with the L3VPN configuration, so we are not going to go through detailed explanation of the configuration steps:

```
ISP1-PE1:
vrf definition CUST1
 rd 1000:1
 !
 address-family ipv4
  route-target export 160.1.1.1:1
  route-target import 160.1.1.5:1
  route-target import 160.1.1.6:1
 exit-address-family
!
!
interface GigabitEthernet3
 vrf forwarding CUST1
 ip address 192.168.1.1 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
router ospf 1 vrf CUST1
 redistribute bgp 1000
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 1000
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
```

```
 exit-address-family
!
vrf definition CUST2
 rd 1000:2
 !
 address-family ipv4
  route-target export 160.1.1.1:2
  route-target import 160.1.1.5:2
  route-target import 160.1.1.6:2
 exit-address-family
!
!
interface GigabitEthernet2
 vrf forwarding CUST2
 ip address 192.168.1.1 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
router eigrp CUST2
!
address-family ipv4 unicast vrf CUST2 autonomous-system 2
  !
  topology base
   redistribute bgp 1000
  exit-af-topology
  network 0.0.0.0
 exit-address-family
!
router bgp 1000
 !
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
```

```
 exit-address-family
!
router bgp 1000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 160.1.1.3 remote-as 1000
 neighbor 160.1.1.3 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 160.1.1.3 activate
  neighbor 160.1.1.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
 exit-address-family
 !
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
 exit-address-family
```

ISP1-RR:

```
router bgp 1000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 160.1.1.1 remote-as 1000
 neighbor 160.1.1.1 update-source Loopback0
 neighbor 160.1.1.5 remote-as 1000
 neighbor 160.1.1.5 update-source Loopback0
 neighbor 160.1.1.6 remote-as 1000
```

```
 neighbor 160.1.1.6 update-source Loopback0
 !
 address-family vpnv4
  neighbor 160.1.1.1 activate
  neighbor 160.1.1.1 send-community extended
  neighbor 160.1.1.1 route-reflector-client
  neighbor 160.1.1.5 activate
  neighbor 160.1.1.5 send-community extended
  neighbor 160.1.1.5 route-reflector-client
  neighbor 160.1.1.6 activate
  neighbor 160.1.1.6 send-community extended
  neighbor 160.1.1.6 route-reflector-client
 exit-address-family
```

ISP1-PE2:

```
router bgp 1000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 160.1.1.3 remote-as 1000
 neighbor 160.1.1.3 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 160.1.1.3 activate
  neighbor 160.1.1.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
 exit-address-family
 !
```

```
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
 exit-address-family
!
vrf definition CUST1
 rd 1000:1
 !
 address-family ipv4
  route-target export 160.1.1.5:1
  route-target import 160.1.1.1:1
  route-target import 160.1.1.6:1
 exit-address-family
!
!
interface Ethernet0/0
 vrf forwarding CUST1
 ip address 192.168.5.1 255.255.255.0
 duplex auto
!
router ospf 1 vrf CUST1
 redistribute bgp 1000 subnets
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 1000
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
 exit-address-family
!
vrf definition CUST2
 rd 1000:2
 !
```

```
 address-family ipv4
  route-target export 160.1.1.5:2
  route-target import 160.1.1.1:2
  route-target import 160.1.1.6:2
 exit-address-family
!
!
interface Ethernet0/2
 vrf forwarding CUST2
 ip address 192.168.5.1 255.255.255.0
 duplex auto
!
router eigrp CUST2
!
address-family ipv4 unicast vrf CUST2 autonomous-system 2
  !
  topology base
   redistribute bgp 1000
  exit-af-topology
  network 0.0.0.0
 exit-address-family
!
router bgp 1000
 !
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
 exit-address-family
!
```

Service Provider 1 basic configuration is done.

We will configure the ISP2 the same way:

ISP2-PE1:

```
router bgp 2000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 170.1.1.3 remote-as 2000
 neighbor 170.1.1.3 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 170.1.1.3 activate
  neighbor 170.1.1.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
 exit-address-family
 !
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
 exit-address-family
vrf definition CUST1
 rd 2000:1
 !
 address-family ipv4
  route-target export 170.1.1.1:1
  route-target import 170.1.1.5:1
  route-target import 170.1.1.6:1
 exit-address-family
!
!
interface GigabitEthernet2
 vrf forwarding CUST1
 ip address 172.16.1.1 255.255.255.0
```

```
 negotiation auto
 no mop enabled
 no mop sysid
!
router ospf 1 vrf CUST1
 redistribute bgp 2000
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 2000
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
 exit-address-family
!
vrf definition CUST2
 rd 2000:2
 !
 address-family ipv4
  route-target export 170.1.1.1:2
  route-target import 170.1.1.5:2
  route-target import 170.1.1.6:2
 exit-address-family
!
!
interface GigabitEthernet3
 vrf forwarding CUST2
 ip address 172.16.1.1 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
router eigrp CUST2
!
```

```
address-family ipv4 unicast vrf CUST2 autonomous-system 2
  !
  topology base
   redistribute bgp 2000
  exit-af-topology
  network 0.0.0.0
 exit-address-family
!
router bgp 2000
 !
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
 exit-address-family
!
```

ISP2-RR:

```
router bgp 2000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 170.1.1.1 remote-as 2000
 neighbor 170.1.1.1 update-source Loopback0
 neighbor 170.1.1.5 remote-as 2000
 neighbor 170.1.1.5 update-source Loopback0
 neighbor 170.1.1.6 remote-as 2000
 neighbor 170.1.1.6 update-source Loopback0
 !
 address-family vpnv4
  neighbor 170.1.1.1 activate
  neighbor 170.1.1.1 send-community extended
  neighbor 170.1.1.1 route-reflector-client
  neighbor 170.1.1.5 activate
  neighbor 170.1.1.5 send-community extended
  neighbor 170.1.1.5 route-reflector-client
  neighbor 170.1.1.6 activate
```

```
  neighbor 170.1.1.6 send-community extended
  neighbor 170.1.1.6 route-reflector-client
 exit-address-family
```

ISP2-PE2:

```
router bgp 2000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 170.1.1.3 remote-as 2000
 neighbor 170.1.1.3 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 170.1.1.3 activate
  neighbor 170.1.1.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
 exit-address-family
 !
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
 exit-address-family
vrf definition CUST1
 rd 2000:1
 !
 address-family ipv4
  route-target export 170.1.1.5:1
  route-target import 170.1.1.1:1
  route-target import 170.1.1.6:1
 exit-address-family
```

```
!
!
interface Ethernet0/0
 vrf forwarding CUST1
 ip address 172.16.5.1 255.255.255.0
 duplex auto
!
router ospf 1 vrf CUST1
 redistribute bgp 2000 subnets
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 2000
 !
 address-family ipv4 vrf CUST1
  redistribute ospf 1 match internal external 1 external 2
 exit-address-family
!
vrf definition CUST2
 rd 2000:2
 !
 address-family ipv4
  route-target export 170.1.1.5:2
  route-target import 170.1.1.1:2
  route-target import 170.1.1.6:2
 exit-address-family
!
!
interface Ethernet0/2
 vrf forwarding CUST2
 ip address 172.16.5.1 255.255.255.0
 duplex auto
!
router eigrp CUST2
```

```
!
address-family ipv4 unicast vrf CUST2 autonomous-system 2
  !
  topology base
   redistribute bgp 2000
  exit-af-topology
  network 0.0.0.0
 exit-address-family
!
router bgp 2000
 !
 address-family ipv4 vrf CUST2
  redistribute eigrp 2
 exit-address-family
!
```

ISP2 basic configuration is also done.

So the customer sites inside each ISP have end to end reachability.

But what about customer sites reachability between two different sites which are connected to the different ISPs?

They don't have any connectivity so far (for example CUST1_CE1 cannot ping CUST1_CE3).

The solution is using Inter-AS MPLS VPN different options.

There are 4 option: A, B, C and D.

The difference between those options are mostly how Edge routers interact with each other at the edge of the network (also in option C we will discuss about RR interaction on different Service Providers).

Task 02:

- Make sure that all customer sites have IP reachability (Use Option D)

Solution:

Option D is a combination of Option A and B, that is why it's called Hybrid.

In order to understand how Option D works, you need to be familiar with both Option A and B.

If you have not completed Option A and B labs, please go back and do them first.

The basic configuration of the RR and PE devices in Option D is just linke Option A and B. the only difference is the configuration of ASBRs (ISP1-E and ISP2-E).

Just linke Option A, we need to configure separate VRFs and Sub-interfaces for each customer on the ASBRs:

```
ISP1-E:
vrf definition CUST1
 rd 10002000:1
 !
 address-family ipv4
  route-target export 10002000:1
  route-target import 10002000:1
!
vrf definition CUST2
 rd 10002000:2
 route-target export 10002000:2
 route-target import 10002000:2
 !
!
interface GigabitEthernet12.100
 encapsulation dot1Q 100
 vrf forwarding CUST1
 ip address 100.1.2.1 255.255.255.0
!
interface GigabitEthernet12.200
 encapsulation dot1Q 200
```

```
 vrf forwarding CUST2
 ip address 200.1.2.1 255.255.255.0
!

ISP2-E:
vrf definition CUST1
 rd 10002000:1
 route-target export 10002000:1
 route-target import 10002000:1
 !
!
vrf definition CUST2
 rd 10002000:2
 route-target export 10002000:2
 route-target import 10002000:2
 !
!
interface GigabitEthernet12.100
 encapsulation dot1Q 100
 vrf forwarding CUST1
 ip address 100.1.2.2 255.255.255.0
!
interface GigabitEthernet12.200
 encapsulation dot1Q 200
 vrf forwarding CUST2
 ip address 200.1.2.2 255.255.255.0
!
```

But unlike Option A, this time there is no need to form IPv4 Unicast eBGP between ASBRs for each of the customers.

This time, we need to configure VPNv4 unicast neighborship between ASBRs (Just linke Option B), in addition to the IPv4 Labeled-Unicast between them, because the packets needs to be label-switched between ASBRs (Again Just linke Option B!).

ISP1-E:

```
interface GigabitEthernet12
 ip address 150.1.1.1 255.255.255.252
 mpls bgp forwarding
!
router bgp 1000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 no bgp default route-target filter
 neighbor 150.1.1.2 remote-as 2000
 neighbor 160.1.1.3 remote-as 1000
 neighbor 160.1.1.3 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 150.1.1.2 activate
  neighbor 150.1.1.2 send-community extended
  neighbor 150.1.1.2 inter-as-hybrid
  neighbor 160.1.1.3 activate
  neighbor 160.1.1.3 send-community extended
  neighbor 160.1.1.3 next-hop-self
 exit-address-family
!
```

ISP2-E:

```
interface GigabitEthernet12
 ip address 150.1.1.2 255.255.255.252
 mpls bgp forwarding
!
router bgp 2000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 no bgp default route-target filter
```

```
 neighbor 150.1.1.1 remote-as 1000
 neighbor 170.1.1.3 remote-as 2000
 neighbor 170.1.1.3 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 150.1.1.1 activate
  neighbor 150.1.1.1 send-community extended
  neighbor 150.1.1.1 inter-as-hybrid
  neighbor 170.1.1.3 activate
  neighbor 170.1.1.3 send-community extended
  neighbor 170.1.1.3 next-hop-self
 exit-address-family
```

In addition to the Option B configuration, there is a need for an additional command under the BGP VPNv4 address family for the neighbor (The neighboring ASBR), the command is:

inter-as-hybrid

One more step to complete the task:

```
ISP1-E:
vrf definition CUST1
 !
 address-family ipv4
  inter-as-hybrid next-hop 100.1.2.2
 exit-address-family
!
vrf definition CUST2
 address-family ipv4
  inter-as-hybrid next-hop 200.1.2.2
 exit-address-family
```

What is this command doing? This command causes the VPNv4 learned routes next-hops to be changed to the IPv4 Labeled-Unicast neighbor address (it is in the GRT or Global Routing Table)

So for all of the Customer routes and traffic we can achieve isolation (using VRFs and Sub-Interfaces just like option A) in addition to the scalability (Using VPNv4 unicast and IPv4 Labeled unicast just like Option B).

That is why this option is called Option D or better to name it Hybrid!

NOTE: Option D is not standardized, Cisco supports it but there is no guarantee that all other vendor's devices support it!

Let's also configure the second ASBR VRFs the same way:

```
ISP2-E:
vrf definition CUST1
 !
 address-family ipv4
  inter-as-hybrid next-hop 100.1.2.1
 exit-address-family
!
vrf definition CUST2
 !
 address-family ipv4
  inter-as-hybrid next-hop 200.1.2.1
 exit-address-family
!
```

Verification:

```
CUST1_CE1#trace 1.3.3.3 source lo 0
Type escape sequence to abort.
Tracing the route to 1.3.3.3
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.1.1 5 msec 0 msec 0 msec
  2 100.1.1.2 [MPLS: Labels 2005/6012 Exp 0] 6 msec 2 msec 1 msec
  3 100.1.1.6 [MPLS: Labels 4006/6012 Exp 0] 1 msec 2 msec 1 msec
  4 100.1.1.18 [MPLS: Labels 5011/6012 Exp 0] 2 msec 1 msec 2 msec
  5 100.1.2.1 [MPLS: Label 6012 Exp 0] 2 msec 1 msec 2 msec
  6 100.1.2.2 1 msec 2 msec 1 msec
  7 200.1.1.21 [MPLS: Labels 5011/1010 Exp 0] 3 msec 3 msec 3 msec
  8 200.1.1.17 [MPLS: Labels 4006/1010 Exp 0] 3 msec 2 msec 3 msec
  9 200.1.1.5 [MPLS: Labels 2006/1010 Exp 0] 2 msec 2 msec 3 msec
 10 172.16.1.1 [MPLS: Label 1010 Exp 0] 3 msec 3 msec 2 msec
 11 172.16.1.100 3 msec 3 msec 2 msec
```

The output is almost the same as Option A verification output with a single difference.

Take a look at the Hop/Step 5 and 6, the IP addresses (100.1.2.1 and 100.1.2.2) were not inside that VRF table, they were in the Global Routing Table or Default VRF but Hybrid command under the VRF definition automaticaly inported them into the VRF routing table:

```
ISP1-E#show ip route vrf CUST1 | begin Gate
Gateway of last resort is not set


      1.0.0.0/32 is subnetted, 4 subnets
B        1.1.1.1 [200/2] via 160.1.1.1, 00:27:59
B        1.2.2.2 [200/11] via 160.1.1.5, 00:28:00
B        1.3.3.3 [20/0] via 100.1.2.2, 00:28:00
B        1.4.4.4 [20/0] via 100.1.2.2, 00:28:00
      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        100.1.2.0/24 is directly connected, GigabitEthernet12.100
L        100.1.2.1/32 is directly connected, GigabitEthernet12.100
      172.16.0.0/24 is subnetted, 2 subnets
B        172.16.1.0 [20/0] via 100.1.2.2, 00:28:00
B        172.16.5.0 [20/0] via 100.1.2.2, 00:28:00
B     192.168.1.0/24 [200/0] via 160.1.1.1, 00:27:59
B     192.168.5.0/24 [200/0] via 160.1.1.5, 00:28:00


ISP1-E#show ip cef vrf CUST1 100.1.2.2
100.1.2.2/32
  attached to GigabitEthernet12.100
```