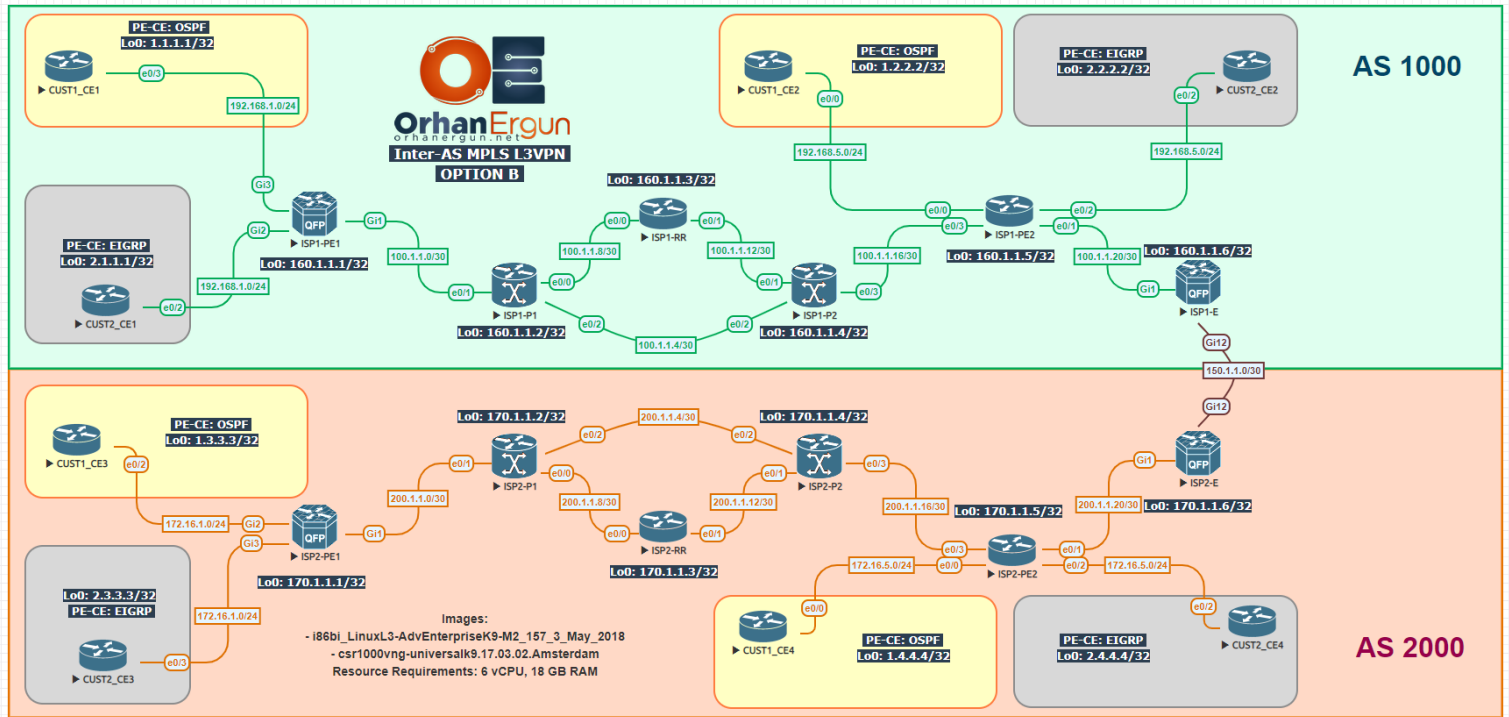


# Inter-AS MPLS L3VPN Option B

## 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 vpv4
  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 vpv4
  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 vpv4
    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 B)



## Solution:

If you have not completed the Inter-AS MPLS VPNs using Option A, please go back and complete that lab first, in order to understand the differences between different options, you should do all of them starting from Option A to D.

As you realized from Task 01, the configuration of some routers in both options (Option A and B) are the same.

The major difference between Option A and B, is where edge routers interact with each other.

In option A, the Edge or ASBRs was forming eBGP IPv4 unicast neighborhood for each customer in an isolated fashion (Each Customer had their own VRF defined on the edge devices). That option had a major problem: Scalability.

How can we solve that issue? The option is a good solution in terms of scalability.

This time (in option B), instead of defining separate VRFs for each customer or creating some sub-interfaces per customer, we can form VPNv4 address family between two ASBRs (Edge routers). Each ASBR can get VPN labels for customer prefixes from PE devices (Or in this example from RRs) and advertise them to other ASBR. So the packet will be label switched even between ASBRs. It is not common to use LDP as a label distribution protocol between ASBRs. The better solution is to use MPLS BGP Forwarding instead. Labels are going to be learned from MP-BGP, and the forwarding will be MPLS using those BGP learned prefixes.

Let's configure the ASBRs to see what exactly Option B does:

```
ISP1-E:
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
```

```
neighbor 150.1.1.2 activate
neighbor 150.1.1.2 send-label
exit-address-family
!
address-family vpnv4
neighbor 150.1.1.2 activate
neighbor 150.1.1.2 send-community extended
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
!
interface GigabitEthernet12
ip address 150.1.1.1 255.255.255.252
negotiation auto
mpls bgp forwarding
!
```

Two ASBRs are forming VPNv4 Unicast neighborship, and also IPv4 Labeled Unicast neighborship. Labeled Unicast is necessary to keep packets label switched between two ASBRs. In IOS and IOS-XE platforms, when you enable MP-BGP IPv4 Labeled-Unicast family, mpls bgp forwarding command will be added by the IOS, IOS-XE to the physical interface that is Labeled Unicast neighborship is being formed (GigabitEthernet 12).

Each of the ASBRs also forming VPNv4 Unicast neighborship with route reflectors, they need to change the next-hop to themselves, otherwise, RRs do not have any idea about the neighboring ASBR IP address as the next-hop of the routes (ISP1-RR has no idea about OSP2-E Gig12 interface IP address).

The second caveat that you need to consider in a scenario like this is:

Keeping all Route-Target values for all customers on ASBRs. In this lab, the ASBRs are only ASBRs! They are not PE devices, they are not connected to any CE device. By default only Route-Reflector keeps all the routes including all Route-Target values, but ISP1-E and ISP2-E are not Route-Reflectors, so they will drop all VPNv4 routes with any RT values (They don't need it).

It is necessary to change that default behavior by using **no bgp default route-target filter** command under BGP process. So the ASBRs won't drop any route at all, they will keep all of them in the MP-BGP VPNv4 unicast Local RIB.

Let's configure the second ASBR the same way:

#### ISP2-E:

```
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
  neighbor 150.1.1.1 activate
  neighbor 150.1.1.1 send-label
exit-address-family
!
address-family vpnv4
  neighbor 150.1.1.1 activate
  neighbor 150.1.1.1 send-community extended
  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
!
interface GigabitEthernet12
  ip address 150.1.1.2 255.255.255.252
  mpls bgp forwarding
!
```

Now it is time to verify the configuration by generating some packets.

### Verification:

```
CUST1_CE1#ping 1.3.3.3 source lo 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!
```



```
CUST1_CE1#trace 1.3.3.3 source lo 0 probe 1
Type escape sequence to abort.
Tracing the route to 1.3.3.3
VRF info: (vrf in name/id, vrf out name/id)
  0/0
  1 192.168.1.1 6 msec
  2 100.1.1.2 [MPLS: Labels 2005/6015 Exp 0] 9 msec
  3 100.1.1.6 [MPLS: Labels 4006/6015 Exp 0] 4 msec
  4 100.1.1.18 [MPLS: Labels 5011/6015 Exp 0] 3 msec
  5 100.1.1.22 [MPLS: Label 6015 Exp 0] 3 msec
  6 150.1.1.2 [MPLS: Label 6013 Exp 0] 3 msec
  7 200.1.1.21 [MPLS: Labels 5011/1010 Exp 0] 3 msec
  8 200.1.1.17 [MPLS: Labels 4006/1010 Exp 0] 3 msec
  9 200.1.1.5 [MPLS: Labels 2006/1010 Exp 0] 3 msec
 10 172.16.1.1 [MPLS: Label 1010 Exp 0] 3 msec
 11 172.16.1.100 3 msec
```

As you can realize from the output, each Service Provider has it's own VPN label for each of the prefixes. ISP1 VPN Label for prefix 1.3.3.3 is 6015, and that label stays intact until it reaches to the ASBR. ISP2 VPN Label for the same prefix is 1010.

In fact ISP1-E device has advertised that label (6015) to ISP1-PE1.

What about label 1010 ? who has advertised it to ISP2-E? the answer is the PE device (ISP2-PE1).

As you can see from the output, the label switching is end to end unlike Option A, which in that case, the packet was not being label switched between two ASBRs.