AGH University of Krakow
Faculty of Computer Science, Electronics and Telecommunications
Teleinformatyka

Inter-AS Networks

Inter-AS Option D

Justyna Gręda, Jan Ściga

Kraków, December 27, 2023

Contents

---

1 Introduction

In the constantly growing world of networking, the seamless interconnection of Autonomous Systems (AS) plays a pivotal role in ensuring robust communication and data exchange across diverse domains. As networks continue to evolve and grow, the need for scalable and flexible solutions becomes increasingly important. Inter-AS Option AB emerges as a compelling framework designed to address these challenges, offering a new approach to interconnecting Autonomous Systems while providing the scalability and flexibility demanded by modern network architectures.

The MPLS VPN–Inter-AS Option D features combine the best functionalities of both Inter-AS Option A and Inter-AS Option B networks to allow a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) service provider to interconnect different autonomous systems in order to provide VPN services. This results in a scalable and simple solution with the possibility to maintain even QoS options between Autonomous System Boundary Router (ASBR) peers.

2 Inter AS Option D

The MPLS VPN–Inter–AS Option D (also referred as AB) feature combines the best functionality of an Inter-AS Option A and Inter-AS Option B network to allow a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) service provider to interconnect different autonomous systems to provide VPN services. Below list present crucial aspects of Inter-AS Option A and B with their drawbacks bolded in red and addressed by the Inter-AS Option D.

Inter-AS Option A

- ASBR peers are connected by multiple subinterfaces with at least one interface VPN that spans the two autonomous systems
- ASBRs associate each subinterface with a VPN routing and forwarding (VRF) instance and a Border Gateway Protocol (BGP) session to signal unlabeled IP prefixes
- **One BGP session is needed for each subinterface (and at least one subinterface for each VPN), which causes scalability concerns as this network grows**

Inter-AS Option B

- A Multiprotocol Border Gateway Protocol (MP-BGP) session is used to distribute labeled VPN prefixes between the ASBR
- ASBR peers are connected by one or more subinterfaces that are enabled to receive MPLS traffic
- **Traffic between ASBRs is MPLS, QoS mechanisms that can be applied only to IP traffic cannot be applied and the VRFs cannot be isolated**

MPLS VPN service providers need to interconnect different AS-es to provide service for multiple VPNs. The MPLS VPN–Inter-AS Option D feature allows the different autonomous systems to interconnect **by using a single MP-BGP session** in the global routing table to carry control plane traffic. This also reduces CPU utilization and makes the network more scalable. MP-BGP session signals VPN prefixes between two ASBRs for each VRF instance. The data plane traffic is on a VRF interface and is **unlabelled that ensures IP QoS functions between ASBR peers** for customer SLAs.

3 Topology

For the purpose of appropriate Inter-AS Option D configuration, the topology presented in Figure 1 has been set in the lab.

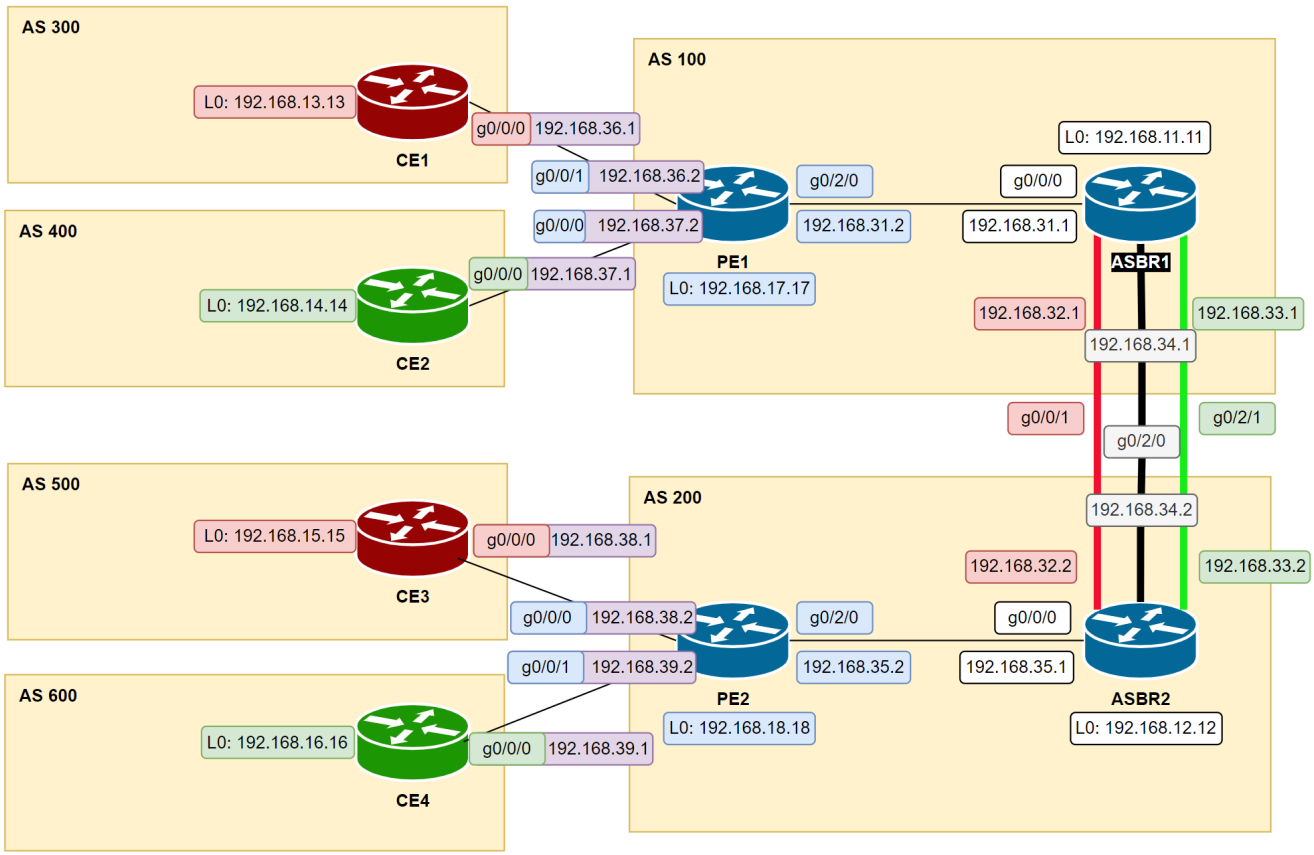

Figure 1: Inter-AS Option D - lab topology

4 Configuration

This section presents configuration steps that allowed us to obtain Inter-AS option D connectivity between particular sites presented in Figure 1.

4.1 Customer Edges

```
!
hostname CE1
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
interface Loopback0
 ip address 192.167.13.13 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 192.168.36.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.12.1 255.255.255.0
 negotiation auto
!
router ospf 1
```

```
 redistribute connected subnets
 network 192.168.13.13 0.0.0.0 area 0
!
router bgp 300
 bgp log-neighbor-changes
 neighbor 192.168.36.2 remote-as 100
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.36.2 activate
 exit-address-family
!
end
```

Listing 1: Customer Edge 1

```
!
hostname CE2
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
interface Loopback0
 ip address 192.168.14.14 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 192.168.37.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.34.1 255.255.255.0
 negotiation auto
!
router ospf 1
 redistribute connected subnets
 network 192.168.14.14 0.0.0.0 area 0
!
router bgp 400
 bgp log-neighbor-changes
 neighbor 192.168.37.2 remote-as 100
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.37.2 activate
 exit-address-family
!
end
```

Listing 2: Customer Edge 2

```
!
hostname CE3
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
interface Loopback0
 ip address 192.168.15.15 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 192.168.38.1 255.255.255.0
 negotiation auto
!
router ospf 1
 redistribute connected subnets
 network 192.168.15.15 0.0.0.0 area 0
!
router bgp 500
 bgp log-neighbor-changes
```

```
 neighbor 192.168.38.2 remote-as 200
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.38.2 activate
 exit-address-family
!
end
```

Listing 3: Customer Edge 3

```
!
hostname CE4
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
interface Loopback0
 ip address 192.168.16.16 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 192.168.39.1 255.255.255.0
 negotiation auto
!
router ospf 1
 redistribute connected subnets
 network 192.168.16.16 0.0.0.0 area 0
!
router bgp 600
 bgp log-neighbor-changes
 neighbor 192.168.39.2 remote-as 200
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.39.2 activate
 exit-address-family
!
end
```

Listing 4: Customer Edge 4

4.2 Provider Edges

```
!
hostname PE1
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 route-target import 200:1
!
ip vrf vpn2
 rd 100:2
 route-target export 100:2
 route-target import 100:2
 route-target import 200:2
!
mpls label protocol ldp
mpls ldp graceful-restart
!
!
interface Loopback0
 ip address 192.168.17.17 255.255.255.255
!
interface GigabitEthernet0/0/0
```

```
 ip vrf forwarding vpn2
 ip address 192.168.37.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip vrf forwarding vpn1
 ip address 192.168.36.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/2/0
 ip address 192.168.31.2 255.255.255.0
 negotiation auto
 mpls ip
 mpls label protocol ldp
!
router ospf 1
 redistribute connected subnets
 network 192.168.17.17 0.0.0.0 area 0
 network 192.168.31.0 0.0.0.255 area 0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.11.11 remote-as 100
 neighbor 192.168.11.11 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.11.11 activate
  neighbor 192.168.11.11 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
  redistribute connected
  neighbor 192.168.36.1 remote-as 300
  neighbor 192.168.36.1 activate
 exit-address-family
 !
 address-family ipv4 vrf vpn2
  redistribute connected
  neighbor 192.168.37.1 remote-as 400
  neighbor 192.168.37.1 activate
 exit-address-family
!
mpls ldp router-id Loopback0 force
!
end
```

Listing 5: Provider Edge 1

```
!
hostname PE2
!
ip vrf vpn1
 rd 200:1
 route-target export 200:1
 route-target import 100:1
 route-target import 200:1
!
ip vrf vpn2
 rd 200:2
 route-target export 200:2
 route-target import 100:2
 route-target import 200:2
!
mpls label protocol ldp
mpls ldp graceful-restart
!
interface Loopback0
 ip address 192.168.18.18 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip vrf forwarding vpn1
 ip address 192.168.38.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip vrf forwarding vpn2
 ip address 192.168.39.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/2/0
```

```
 ip address 192.168.35.2 255.255.255.0
 negotiation auto
 mpls ip
 mpls label protocol ldp
!
router ospf 1
 redistribute connected subnets
 network 192.168.18.18 0.0.0.0 area 0
 network 192.168.35.0 0.0.0.255 area 0
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.12.12 remote-as 200
 neighbor 192.168.12.12 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.12.12 activate
  neighbor 192.168.12.12 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
  redistribute connected
  neighbor 192.168.38.1 remote-as 500
  neighbor 192.168.38.1 activate
 exit-address-family
 !
 address-family ipv4 vrf vpn2
  neighbor 192.168.39.1 remote-as 600
  neighbor 192.168.39.1 activate
 exit-address-family
!
mpls ldp router-id Loopback0 force
!
end
```

Listing 6: Provider Edge 2

4.3 ASBRs

```
!
hostname ASBR1
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 route-target import 200:1
 inter-as-hybrid next-hop 192.168.32.2
!
ip vrf vpn2
 rd 100:2
 route-target export 100:2
 route-target import 100:2
 route-target import 200:2
 inter-as-hybrid next-hop 192.168.33.2
!
mpls label protocol ldp
mpls ldp graceful-restart
!
interface Loopback0
 ip address 192.168.11.11 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 192.168.31.1 255.255.255.0
 negotiation auto
 mpls ip
 mpls label protocol ldp
!
interface GigabitEthernet0/0/1
 ip vrf forwarding vpn1
 ip address 192.168.32.1 255.255.255.0
```

```
 negotiation auto
!
interface GigabitEthernet0/2/0
 ip address 192.168.34.1 255.255.255.0
 negotiation auto
 mpls bgp forwarding
!
interface GigabitEthernet0/2/1
 ip vrf forwarding vpn2
 ip address 192.168.33.1 255.255.255.0
 negotiation auto
!
router ospf 1
 network 192.168.11.11 0.0.0.0 area 0
 network 192.168.31.0 0.0.0.255 area 0
!
router bgp 100
 bgp router-id 192.168.11.11
 bgp log-neighbor-changes
 neighbor 192.168.17.17 remote-as 100
 neighbor 192.168.17.17 update-source Loopback0
 neighbor 192.168.34.2 remote-as 200
 !
 address-family ipv4
  neighbor 192.168.17.17 activate
  neighbor 192.168.34.2 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.17.17 activate
  neighbor 192.168.17.17 send-community extended
  neighbor 192.168.34.2 activate
  neighbor 192.168.34.2 send-community both
  neighbor 192.168.34.2 inter-as-hybrid
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 exit-address-family
!
ip route vrf vpn1 192.168.12.12 255.255.255.255 GigabitEthernet0/0/1 192.168.32.2
ip route vrf vpn2 192.168.12.12 255.255.255.255 GigabitEthernet0/2/1 192.168.33.2
!
mpls ldp router-id Loopback0 force
!
end
```

Listing 7: ASBR 1

```
!
hostname ASBR2
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
ip vrf vpn1
 rd 200:1
 route-target export 200:1
 route-target import 100:1
 route-target import 200:1
 inter-as-hybrid next-hop 192.168.32.1
!
ip vrf vpn2
 rd 200:2
 route-target export 200:2
 route-target import 100:2
 route-target import 200:2
 inter-as-hybrid next-hop 192.168.33.1
!
mpls label protocol ldp
mpls ldp graceful-restart
!
interface Loopback0
```

```
 ip address 192.168.12.12 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 192.168.35.1 255.255.255.0
 negotiation auto
 mpls ip
 mpls label protocol ldp
!
interface GigabitEthernet0/0/1
 ip vrf forwarding vpn1
 ip address 192.168.32.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/2/0
 ip address 192.168.34.2 255.255.255.0
 negotiation auto
 mpls bgp forwarding
!
interface GigabitEthernet0/2/1
 ip vrf forwarding vpn2
 ip address 192.168.33.2 255.255.255.0
 negotiation auto
!
router ospf 1
 redistribute connected subnets
 network 192.168.11.11 0.0.0.0 area 0
 network 192.168.35.0 0.0.0.255 area 0
!
router bgp 200
 bgp router-id 192.168.12.12
 bgp log-neighbor-changes
 neighbor 192.168.18.18 remote-as 200
 neighbor 192.168.18.18 update-source Loopback0
 neighbor 192.168.34.1 remote-as 100
 !
 address-family ipv4
  neighbor 192.168.18.18 activate
  neighbor 192.168.18.18 send-community extended
  neighbor 192.168.34.1 activate
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.18.18 activate
  neighbor 192.168.18.18 send-community extended
  neighbor 192.168.34.1 activate
  neighbor 192.168.34.1 send-community both
  neighbor 192.168.34.1 inter-as-hybrid
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 exit-address-family
!
ip route vrf vpn1 192.168.11.11 255.255.255.255 GigabitEthernet0/0/1 192.168.32.1
ip route vrf vpn2 192.168.11.11 255.255.255.255 GigabitEthernet0/2/1 192.168.33.1
!
end
```

Listing 8: ASBR 2

5 Results

With the proper configurations set on each of the devices the connections between the CEs are expected to be as follows:

- CE4 can only reach CE2 and vice versa
- CE3 can only reach CE1 and vice versa

There should be no working connection between CEs marked with different colors. A few commands were ran to check whether the configuration allows only for the desired connections, results can be seen in Figures 2 and Figures 3.

```
CE4#ping 192.168.13.13
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.13.13, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CE4#
```

Figure 2: ping between CE4 and CE1

```
CE3#ping 192.168.14.14
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.14.14, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CE3#
```

Figure 3: ping between CE3 and CE2

Starting with the connections that should not work, it can be seen in the Figures 2 and 3 that the success rate of a ping command between two CEs belonging to different VPNs is 0 percent, so the configuration is correct.

```
CE2>
CE2>traceroute 192.168.16.16
Type escape sequence to abort.
Tracing the route to 192.168.16.16
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.37.2 1 msec 1 msec 0 msec
  2 192.168.33.1 [MPLS: Label 22 Exp 0] 2 msec 1 msec 1 msec
  3 192.168.33.2 1 msec 2 msec 1 msec
  4 192.168.39.2 [AS 600] [MPLS: Label 18 Exp 0] 1 msec 1 msec 2 msec
  5 192.168.39.1 [AS 600] 1 msec 1 msec *
CE2>
```

Figure 4: traceroute between CE2 and CE4

```
CE3#traceroute 192.168.13.13
Type escape sequence to abort.
Tracing the route to 192.168.13.13
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.38.2 1 msec 1 msec 1 msec
  2 192.168.32.2 [MPLS: Label 24 Exp 0] 1 msec 1 msec 1 msec
  3 192.168.32.1 1 msec 1 msec 1 msec
  4 192.168.36.2 [AS 100] [MPLS: Label 21 Exp 0] 2 msec 2 msec 1 msec
  5 192.168.36.1 [AS 100] 1 msec 2 msec *
CE3#
```

Figure 5: traceroute between CE3 and CE1

Let us focus on the traceroute between CE2 and CE4 seen in figure 4. The packet enters AS100 by PE1, then reaches ASBR1 with an MPLS label 22. It moves to AS200 and ASBR2 and no MPLS label is present. Then, the MPLS label 18 is imposed again as the packet transports from ASBR2 to PE2. The label is off when it reaches the target CE4. The traceroute works as it is supposed to and similar steps can be noticed when running the command between CE3 and CE1 (figure 5).

Taking a look at the forwarding tables on the ASBRs, we can see that they match what traceroute has shown. For example. when it comes to ASBR1, the local label is 22 when the next hop is 192.168.33.2 (CE2-CE4, figure 4) and for ASBR2, the local label is 24 when the next hop is 192.168.32.1 (CE3-CE1, figure 5). Both situations indicate *No Label* when it comes to the outgoing one.

```
ASBR1#show mpls forwarding-table
Local      Outgoing    Prefix           Bytes Label   Outgoing    Next Hop
Label      Label       or Tunnel Id     Switched      interface
16         Pop Label   192.168.17.17/32 0             Gi0/0/0     192.168.31.2
17         Pop Label   IPv4 VRF[V]      0             aggregate/vpn2
18         Pop Label   IPv4 VRF[V]      0             aggregate/vpn1
19         Pop Label   192.168.34.2/32  0             Gi0/2/0     192.168.34.2
20         No Label    192.168.15.15/32[V]    \
                                        1518          Gi0/0/1     192.168.32.2
21         No Label    192.168.38.0/24[V]     \
                                        686           Gi0/0/1     192.168.32.2
22         No Label    192.168.16.16/32[V]    \
                                        1326          Gi0/2/1     192.168.33.2
23         No Label    192.168.39.0/24[V]     \
                                        686           Gi0/2/1     192.168.33.2
ASBR1#
```

Figure 6: Forwarding table - ASBR1

```
ASBR2#sh mpls forwarding-table
Local      Outgoing    Prefix           Bytes Label   Outgoing    Next Hop
Label      Label       or Tunnel Id     Switched      interface
16         Pop Label   192.168.18.18/32 0             Gi0/0/0     192.168.35.2
17         Pop Label   192.168.34.1/32  0             Gi0/2/0     192.168.34.1
18         No Label    192.168.36.0/24[V]     \
                                        1826          Gi0/0/1     192.168.32.1
19         No Label    192.168.37.0/24[V]     \
                                        1942          Gi0/2/1     192.168.33.1
21         No Label    192.168.14.14/32[V]    \
                                        378           Gi0/2/1     192.168.33.1
22         Pop Label   IPv4 VRF[V]      0             aggregate/vpn1
23         Pop Label   IPv4 VRF[V]      0             aggregate/vpn2
24         No Label    192.168.13.13/32[V]    \
                                        378           Gi0/0/1     192.168.32.1
ASBR2#
```

Figure 7: Forwarding table - ASBR2

To understand the last two labels visible in the traceroute, it is necessary to take a look at the Provider Edge forwarding tables:

```
PE1#show mpls forwa
PE1#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  192.168.11.11/32 0             Gi0/2/0    192.168.31.1
18         No Label   192.168.14.14/32[V]   \
                                       126           Gi0/0/0    192.168.37.1
19         No Label   192.168.36.0/24[V]    \
                                       2582          aggregate/vpn1
20         No Label   192.168.37.0/24[V]    \
                                       3454          aggregate/vpn2
21         No Label   192.168.13.13/32[V]   \
                                       126           Gi0/0/1    192.168.36.1
PE1#
```

Figure 8: Forwarding table - PE1

```
PE2#sh mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         No Label   192.168.38.0/24[V]    \
                                       1442          aggregate/vpn1
17         No Label   192.168.15.15/32[V]   \
                                       1266          Gi0/0/0    192.168.38.1
18         No Label   192.168.16.16/32[V]   \
                                       822           Gi0/0/1    192.168.39.1
19         No Label   192.168.39.0/24[V]    \
                                       1442          aggregate/vpn2
20         Pop Label  192.168.12.12/32  0            Gi0/2/0    192.168.35.1
21         Pop Label  192.168.34.0/24   0            Gi0/2/0    192.168.35.1
22         17         192.168.34.1/32   0            Gi0/2/0    192.168.35.1
PE2#
```

Figure 9: Forwarding table - PE2

Looking at the PE2, the local label is 18, when the next hop is 192.168.39.1, so the incoming interface of the CE4. It can be seen during the traceroute between CE2 and CE4, pictured in figure 4. For the traceroute between CE3 and CE1 (figure 5), the local label is 21. It matches the table entry, for the next hop of 192.168.36.1, which is the incoming interface of the CE1.

Next, the command *show ip bgp vpnv4 all* was run on PE1 and PE2.

```
PE1#
PE1#sh ip bgp vpnv4 all
BGP table version is 15, local router ID is 192.168.17.17
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,
              t secondary path,
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 vpn1)
 *>   192.168.13.13/32 192.168.36.1             0             0 300 ?
 *>i  192.168.15.15/32 192.168.11.11            0    100      0 200 500 ?
 *    192.168.36.0     192.168.36.1             0             0 300 ?
 *>                    0.0.0.0                  0         32768 ?
 *>i  192.168.38.0     192.168.11.11            0    100      0 200 ?
Route Distinguisher: 100:2 (default for vrf vpn2)
 *>   192.168.14.14/32 192.168.37.1             0             0 400 ?
 *>i  192.168.16.16/32 192.168.11.11            0    100      0 200 600 ?
 *    192.168.37.0     192.168.37.1             0             0 400 ?
 *>                    0.0.0.0                  0         32768 ?
 *>i  192.168.39.0     192.168.11.11            0    100      0 200 600 ?
PE1#
PE1#
```

Figure 10: BGP table - PE1

```
PE2#sh ip bgp vpnv4 all
BGP table version is 24, local router ID is 192.168.18.18
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,
              t secondary path,
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: 200:1 (default for vrf vpn1)
 *>i  192.168.13.13/32 192.168.12.12            0    100      0 100 300 ?
 *>   192.168.15.15/32 192.168.38.1             0             0 500 ?
 *>i  192.168.36.0     192.168.12.12            0    100      0 100 ?
 *    192.168.38.0     192.168.38.1             0             0 500 ?
 *>                    0.0.0.0                  0         32768 ?
Route Distinguisher: 200:2 (default for vrf vpn2)
 *>i  192.168.14.14/32 192.168.12.12            0    100      0 100 400 ?
 *>   192.168.16.16/32 192.168.39.1             0             0 600 ?
 *>i  192.168.37.0     192.168.12.12            0    100      0 100 ?
 r>   192.168.39.0     192.168.39.1             0             0 600 ?
PE2#
```

Figure 11: BGP table - PE2

In the figures 10 and 11 we can see the routes available for each of the VPNs. For example, for the VPN2 and Route Distinguisher 100:2, if the route starts at the loopback address of CE2, the next hop is the outgoing port of the CE2 to the PE1. The entry is similar for the network of 192.168.36.0. The BGP table on the PE only has knowledge about the Customer Edge nodes in the same VPN (as we can see the route for the network 192.168.15.15, which is the loopback of CE3) and the networks that connect them to the PEs. The routes are the same on tables for both of the PEs (as they are symmetric), only the next hops differ.

Last but not least, the command *show ip route vrf* was run on ASBRs and PEs to see the routing tables of each of the VPNs on all VPN-aware devices.

```
Routing Table: vpn1
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, p - overrides from PfR

Gateway of last resort is not set

      192.168.12.0/32 is subnetted, 1 subnets
S        192.168.12.12 [1/0] via 192.168.32.2, GigabitEthernet0/0/1
      192.168.13.0/32 is subnetted, 1 subnets
B        192.168.13.13 [200/0] via 192.168.17.17, 00:06:59
      192.168.15.0/32 is subnetted, 1 subnets
B        192.168.15.15 [20/0] via 192.168.32.2, 00:28:54
      192.168.32.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.32.0/24 is directly connected, GigabitEthernet0/0/1
L        192.168.32.1/32 is directly connected, GigabitEthernet0/0/1
B     192.168.36.0/24 [200/0] via 192.168.17.17, 00:32:06
B     192.168.38.0/24 [20/0] via 192.168.32.2, 00:28:54

Routing Table: vpn2
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, p - overrides from PfR

Gateway of last resort is not set

      192.168.12.0/32 is subnetted, 1 subnets
S        192.168.12.12 [1/0] via 192.168.33.2, GigabitEthernet0/2/1
      192.168.14.0/32 is subnetted, 1 subnets
B        192.168.14.14 [200/0] via 192.168.17.17, 00:32:06
      192.168.16.0/32 is subnetted, 1 subnets
B        192.168.16.16 [20/0] via 192.168.33.2, 00:28:54
      192.168.33.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.33.0/24 is directly connected, GigabitEthernet0/2/1
L        192.168.33.1/32 is directly connected, GigabitEthernet0/2/1
B     192.168.37.0/24 [200/0] via 192.168.17.17, 00:32:06
B     192.168.39.0/24 [20/0] via 192.168.33.2, 00:28:54
ASBR1#
```

Figure 12: IP route VRF - ASBR1

Figure 13: IP route VRF - ASBR2

Starting with the ASBR1 (figure 12), the routing table for VPN1 consists of:

- loopback address of ASBR2
- loopback address of CE1
- loopback address of CE3
- address of the VPN1-specific network between ASBR1 and ASBR2
- address of the network between CE1 and PE1
- address of the network between PE2 and CE3

The scenario is very similar for VPN2. When it comes to ASBR2 (figure 13), only one address is different in the VPN1 section - the loopback of the device (now it is ASBR1).

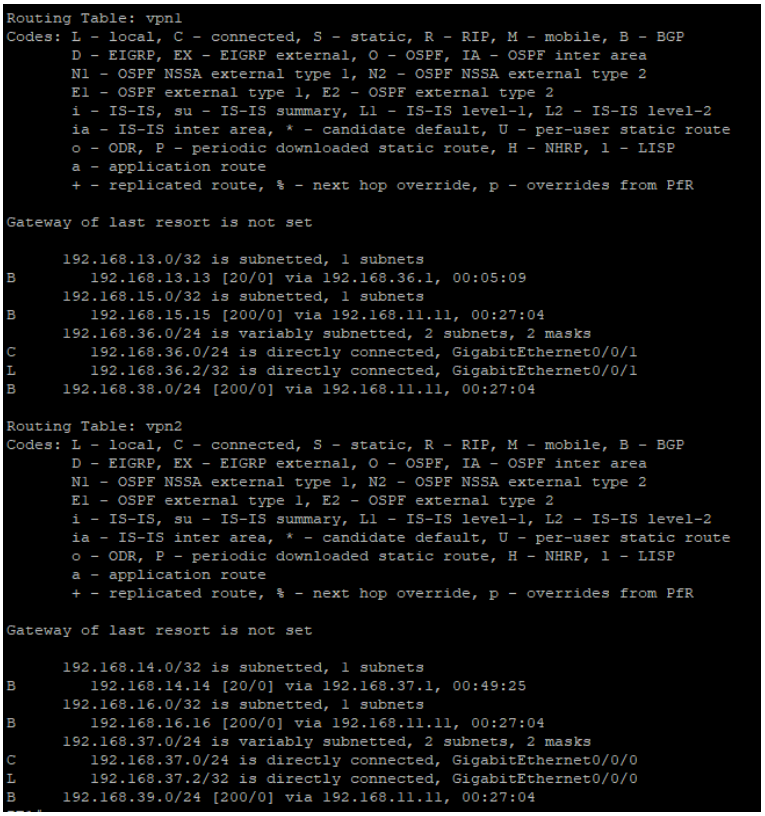

Figure 14: IP route VRF - PE1

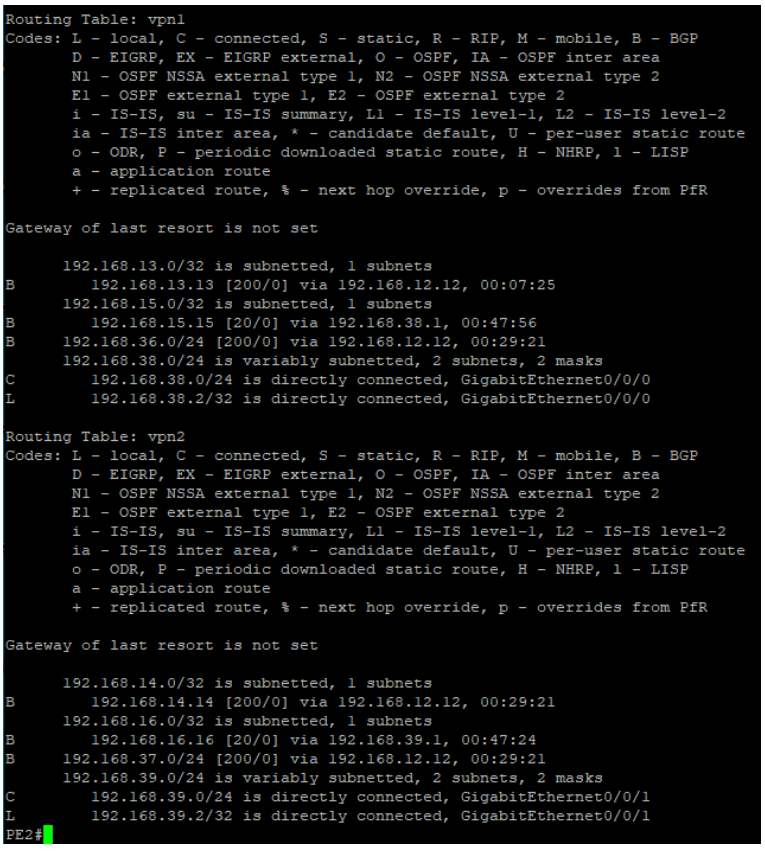

Figure 15: IP route VRF - PE2

The same command ran on the PEs (figures 14 and 15) show similar network, minus the loopback address of the device and the network connecting the ASBRs for a specific VPN. Meaning, the VRF routing table shows almost all networks necessary to connect from one VPN to another. Only one is missing - the connection between the PE and the ASBR. It can be seen by running the simple *show ip route* on the PE. The example can be seen in the figure 16, with the network 192.168.31.0 connecting the PE1 to ASBR1 clearly visible.

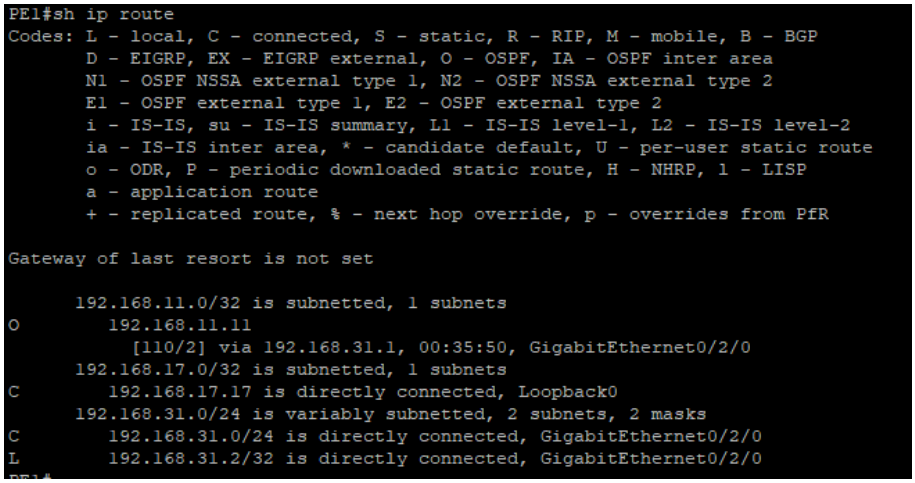

Figure 16: IP route - PE1

6 Summary

The project was a success as the connection between the customers in the same VPN was reached, while at the same time the connection between different VPNs was forbidden. The document provides all the results of our configurations and testing, proving the small network works correctly. It also explores the compatibility of Inter-AS Option AB with existing protocols and guides the reader to their own configuration of Inter-AS Option AB.