Conception d'un réseau opérateur et client multi-sites sécurisé (MPLS, BGP, IPsec)

Fabien MOINE

le 03/02/2025

1- Préparation du projet

A. Structure générale du réseau

1. Backbone opérateur

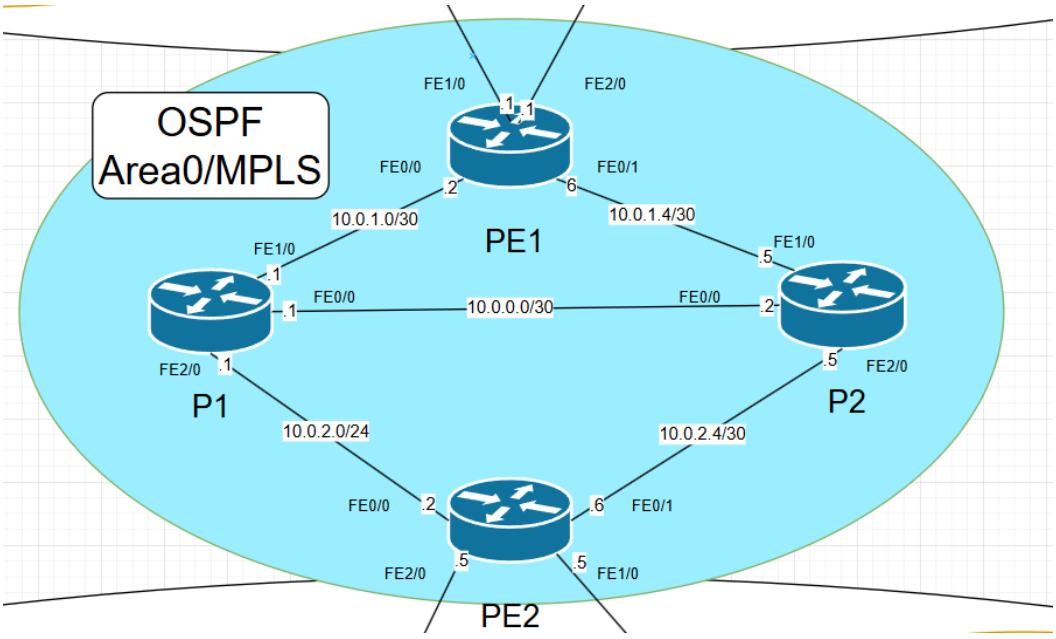
C'est le cœur du réseau, jouant un rôle de transit entre les différentes entités. Il est constitué de plusieurs routeurs interconnectés, assurant la redondance et la continuité des communications.

a) OSPF/MPLS

Dans cette topologie nous avons donc les Router P1 et P2 (Provider), qui discutent en OSPF et MPLS entre eux et avec PE1 et PE2 (Provider Edge). Les router PE peuvent donc se joindre via les routeurs Provider sans être directement connectés entre eux.

Cela rend le réseau très flexible : il suffit de configurer MPLS ou OSPF sur n'importe quel routeur PE que l'on rajoute au réseau et de le connecter à un routeur Provider pour qu'il joigne tous les autres routeurs du backbone.

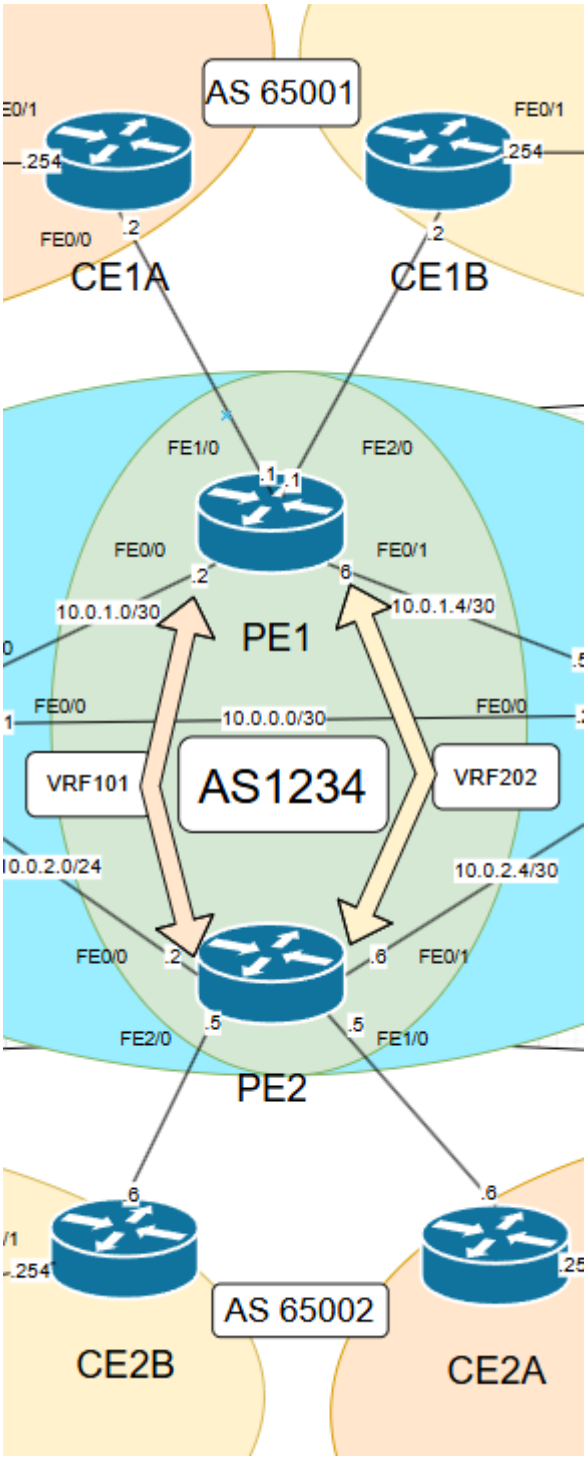
MPLS**(Multiprotocol Label Switching)** est une technologie de transport réseau qui optimise et accélère le routage des paquets en utilisant des étiquettes (labels) au lieu des adresses IP classiques. Il est souvent utilisé par les fournisseurs d'accès Internet (FAI) et les grandes entreprises pour assurer un transport efficace et sécurisé des données.



b) BGP et VRF

Les routeurs PE sont dans un même AS. Ils sont directement connectés à des routeur CE (Customer Edge), eux même dans des AS distinct :

- AS 1234 pour le coeur opérateur
- AS 65001 pour les routeur CE1A et CE1B
- AS 65002 pour les routeurs CE2A et CE2B



Nous utiliserons deux VRF pour échanger les routes annoncées en eBGP entre les deux routeurs PE. Cette organisation permet d'éviter que les routes du client A et celle du client B se confondent.

- VRF101 : CE1A et CE2A
- VRF202 : CE1B et CE2B

De cette manière CE1A et CE2A annoncent leurs routes en eBGP dans la VRF 101 et CE1B et CE2B échanges les leurs dans la VRF 202.

VRF(Virtual Routing and Forwarding) sont une technologie permettant la segmentation du routage sur un même équipement réseau. Elles permettent de créer plusieurs tables de routage indépendantes, chacune isolée des autres, comme si plusieurs routeurs virtuels coexistaient sur un même matériel.

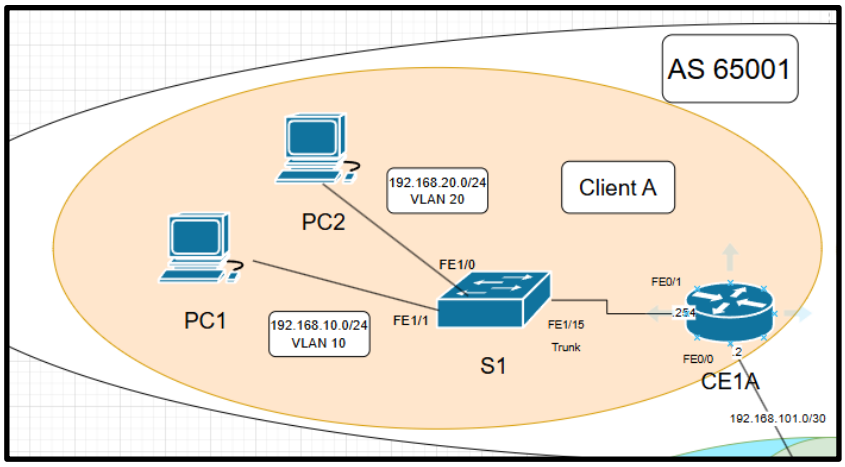
BGP (Border Gateway Protocol) est un protocole de routage dynamique utilisé pour échanger des routes entre autonomous systems (AS) sur Internet et dans certains réseaux privés. Il est fondamental pour l'interconnexion des réseaux à grande échelle, notamment chez les fournisseurs d'accès à Internet (FAI) et les grandes entreprises.

2. Sites clients

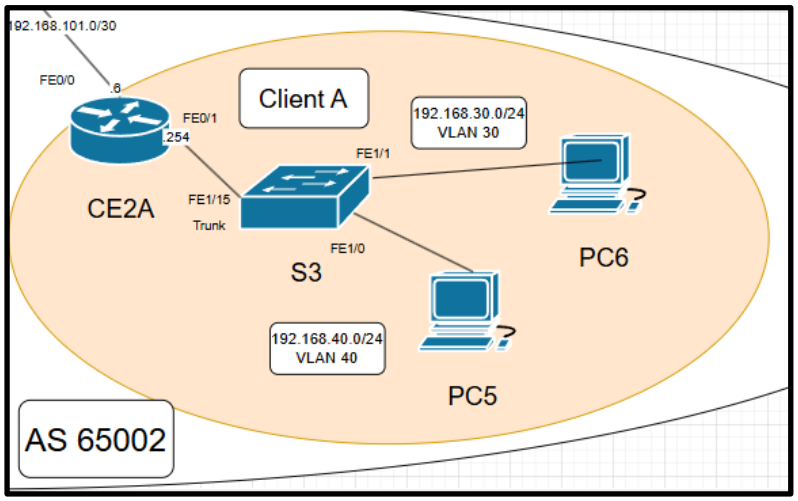
Chaque client a deux sites : un site 1 et un site 2. Les sites sont distant il ne sont donc pas dans le même AS. Nous devons donc traverser le réseau opérateur pour les faire communiquer. On mettra en place un VPN IPSEC pour cela.
Les sites sont configurés comme suit :

- Un routeur CE(Customer Edge) qui fait office de passerelle vers le Backbone
- Un switch connecté au routeur via un lien Trunk
- Deux PC connectés au Switch dans des Vlans différents pour séparer les services de l'entreprise
- Routage Inter-VLAN
- Serveur DHCP et ACl dans le routeur CE

Site 1 Client A :



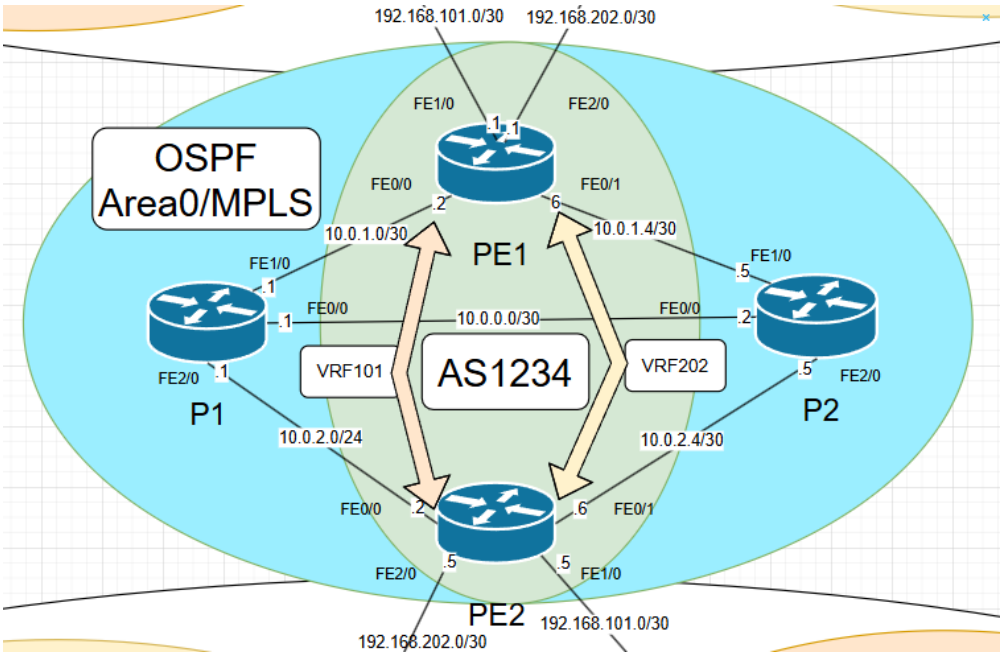
Site 2 Client A :

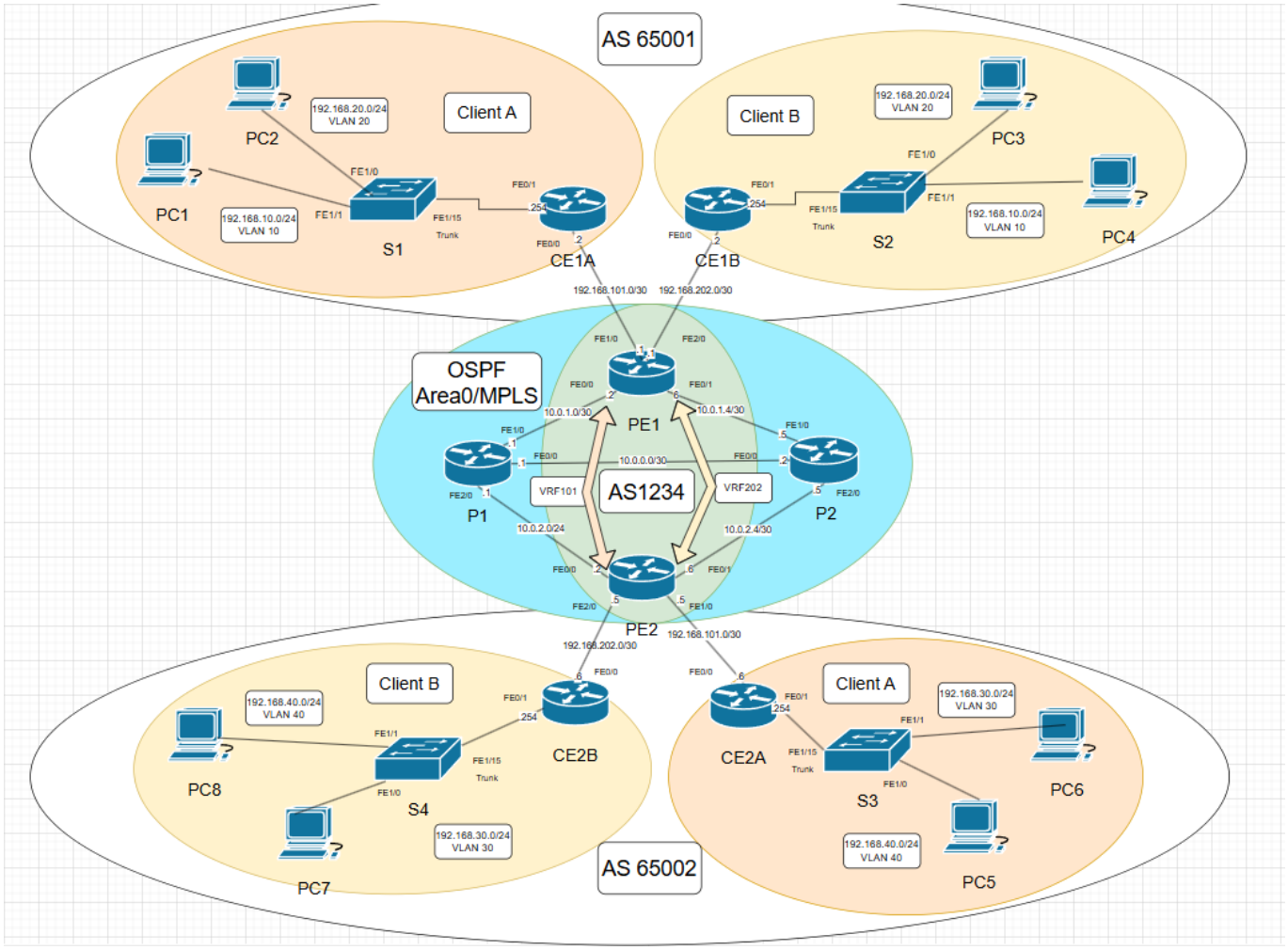


Note : Nous pouvons effectuer la même configuration pour les sites du Client 2 car nous avons séparé les tables de routage clients avec les VRF.

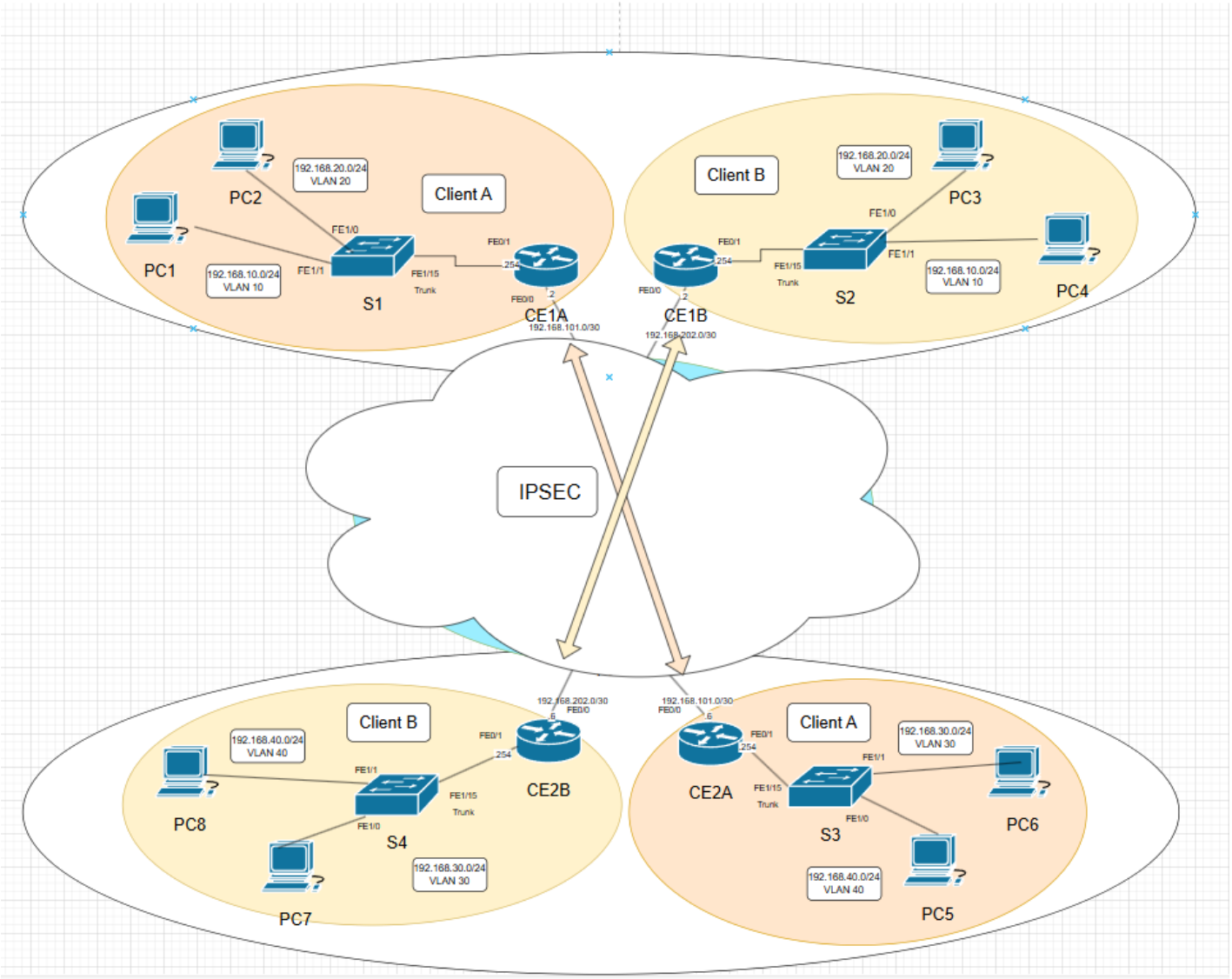
3. Architecture Finale (Opérateur et client)

Opérateur :





Client :


PC2
192.168.20.0/24
VLAN 20
Client A
PC1
192.168.10.0/24
VLAN 10
FE1/0
FE1/1
S1
FE1/15
Trunk
FE0/1
.254
FE0/0
.2
CE1A
192.168.101.0/30
Client B
192.168.20.0/24
VLAN 20
PC3
FE1/0
FE0/1
.254
FE1/15
Trunk
FE1/1
S2
192.168.10.0/24
VLAN 10
PC4
FE0/0
.2
CE1B
192.168.202.0/30
IPSEC
192.168.202.0/30
FE0/0
.6
FE0/1
.254
CE2B
Client B
192.168.40.0/24
VLAN 40
PC8
FE1/1
FE1/15
Trunk
FE1/0
S4
PC7
192.168.30.0/24
VLAN 30
192.168.101.0/30
FE0/0
.6
FE0/1
.254
CE2A
Client A
192.168.30.0/24
VLAN 30
FE1/1
FE1/15
Trunk
FE1/0
S3
PC6
192.168.40.0/24
VLAN 40
PC5

L'architecture est divisée en plusieurs zones :

- AS1234 (backbone opérateur) : C'est le cœur du réseau, jouant un rôle de transit entre les différentes entités. Il est constitué de plusieurs routeurs interconnectés, assurant la redondance et la continuité des communications.
- AS65001 & AS65002 (Zones client) : Ces AS représentent des sites clients distincts, chacun comprenant plusieurs équipements réseau :
    - Clients A et B : Chaque site contient un client organisés en plusieurs services qui se situent dans des VLAN distincts, séparant les domaines de diffusion et améliorant la sécurité. Les PC obtiennent leurs adresses IP grâce au serveur DHCP présent dans les routeurs CE (distribue aussi les IPV6). Il y a également du routage entre les VLAN.
    - Switching et Routage : Chaques services (présent sur chaques sites dans des VLANs différents) sont connectés à un commutateur de niveau 2 (ESW) et un routeur d'accès (CE), permettant la segmentation et l'optimisation du trafic.

L'architecture intègre plusieurs mécanismes de sécurité et d'optimisation du réseau :

- Routage : Implémentation de MPLS, OSPF ainsi que de deux VRF (qui échangent les routes entre les deux routeurs PE) au sein du cœur opérateur afin d'éviter que les routes du client A se confondent avec celles du client B.
- Filtrage inter-VLAN : Implémentation de VLANs pour segmenter le trafic des différents services présents sur les sites clients et éviter d'éventuelles menaces.
- VPN & Sécurisation des tunnels : Des tunnels sécurisés (VPN) garantissent la confidentialité et l'intégrité des échanges entre les sites.

Cette infrastructure vise à garantir :

- Une communication fluide et sécurisée entre les différentes entités.
- Une évolutivité permettant l'ajout futur de nouveaux sites sans compromettre la sécurité ni la performance.

2- Mise en oeuvre sur GNS3

A. Backbone

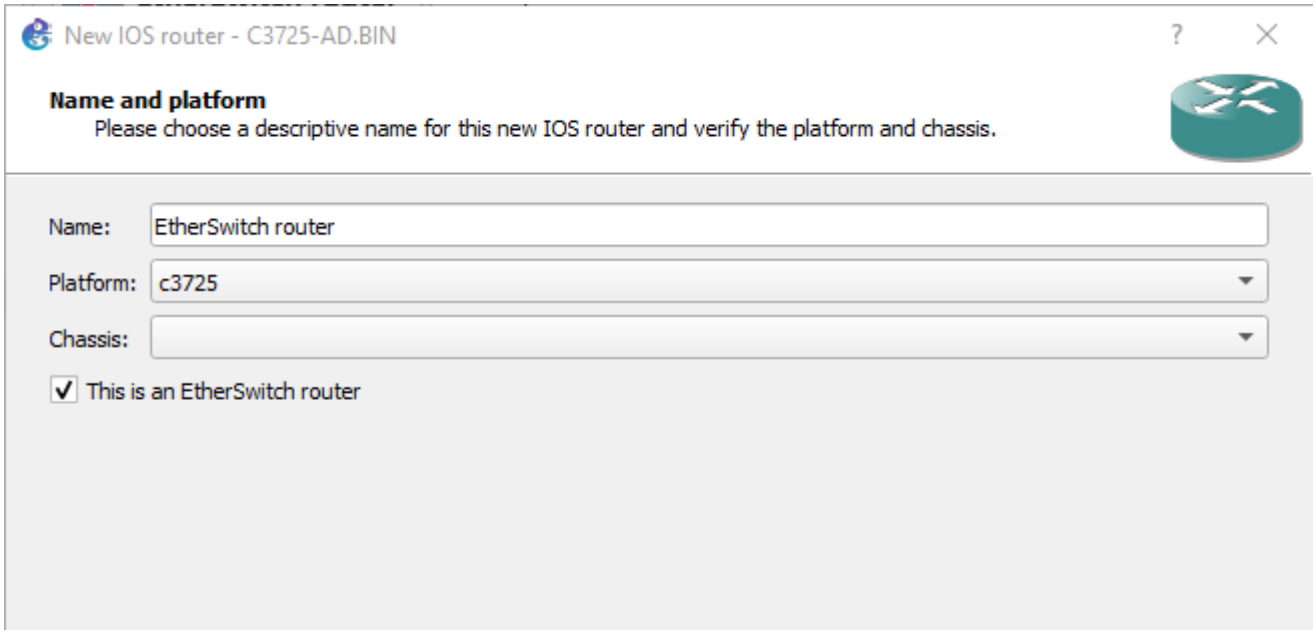
1. Ajout des images

Routeur :

File → Preferences → IosRouter → New → Fichier C3725-AD.BIN

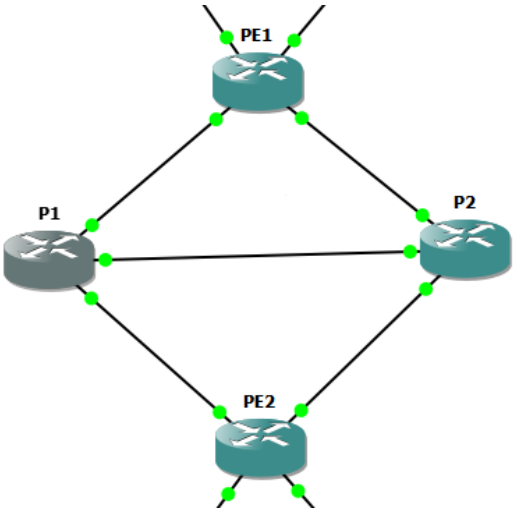
Switch :

Même chose et cocher la case suivante :



2. Ajout des Interfaces

On commence par ajouter toutes nos interfaces sur les routeurs de notre backbone en suivant notre schéma.



a) PE1

```
PE1#sh ip int br
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            10.0.1.2        YES NVRAM  up                    up

FastEthernet0/1            10.0.1.6        YES NVRAM  up                    up

FastEthernet1/0            192.168.101.1   YES NVRAM  up                    up

FastEthernet2/0            192.168.202.1   YES NVRAM  up                    up

Loopback0                  11.11.11.11     YES NVRAM  up                    up
```

b) PE2

```
PE2#sh ip int br
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            10.0.2.2        YES NVRAM  up                    up

FastEthernet0/1            10.0.2.6        YES NVRAM  up                    up

FastEthernet1/0            192.168.101.5   YES NVRAM  up                    up

FastEthernet2/0            192.168.202.5   YES NVRAM  up                    up

Loopback0                  22.22.22.22     YES NVRAM  up                    up
```

c) P1

```
P1#sh ip int br
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            10.0.0.1        YES NVRAM  up                    up

FastEthernet0/1            unassigned      YES NVRAM  administratively down down

FastEthernet1/0            10.0.1.1        YES NVRAM  up                    up

FastEthernet2/0            10.0.2.1        YES NVRAM  up                    up

Loopback0                  1.1.1.1         YES NVRAM  up                    up
```

d) P2

```
P2#sh ip  int br
Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            10.0.0.2        YES NVRAM  up                    up

FastEthernet0/1            unassigned      YES NVRAM  administratively down down

FastEthernet1/0            10.0.1.5        YES NVRAM  up                    up

FastEthernet2/0            10.0.2.5        YES NVRAM  up                    up

Loopback0                  2.2.2.2         YES NVRAM  up                    up
```

3. Configuration MPLS/ OSPF

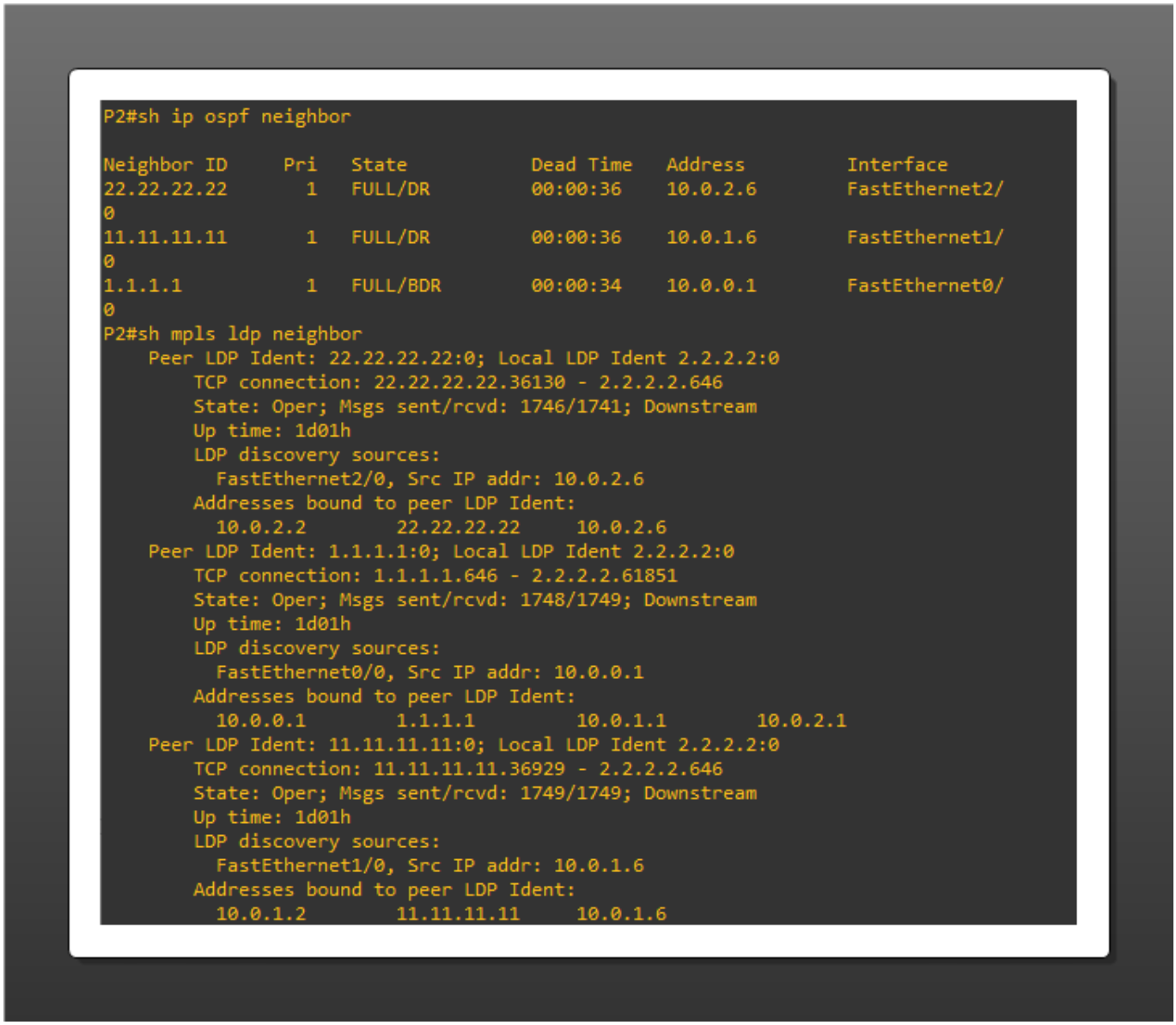
Sur tous les routeurs :

router ospf 1
network 0.0.0.0 0.0.0.0 area 0

Pour chaque interface en contact avec le Backbone :

Interface FastEthernetX/X
mpls ip

a) P2



```
P2#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
22.22.22.22       1   FULL/DR         00:00:36    10.0.2.6        FastEthernet2/
0
11.11.11.11       1   FULL/DR         00:00:36    10.0.1.6        FastEthernet1/
0
1.1.1.1           1   FULL/BDR        00:00:34    10.0.0.1        FastEthernet0/
0
P2#sh mpls ldp neighbor
    Peer LDP Ident: 22.22.22.22:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 22.22.22.22.36130 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 1746/1741; Downstream
        Up time: 1d01h
        LDP discovery sources:
          FastEthernet2/0, Src IP addr: 10.0.2.6
        Addresses bound to peer LDP Ident:
          10.0.2.2        22.22.22.22     10.0.2.6
    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 1.1.1.1.646 - 2.2.2.2.61851
        State: Oper; Msgs sent/rcvd: 1748/1749; Downstream
        Up time: 1d01h
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.0.0.1
        Addresses bound to peer LDP Ident:
          10.0.0.1        1.1.1.1         10.0.1.1        10.0.2.1
    Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 11.11.11.11.36929 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 1749/1749; Downstream
        Up time: 1d01h
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 10.0.1.6
        Addresses bound to peer LDP Ident:
          10.0.1.2        11.11.11.11     10.0.1.6
```

OSPF :

Neighbor ID	Pri	State	Dead Time	Address	Interface
22.22.22.22 (PE2)	1	FULL/DR	00:00:36	10.0.2.6	FastEthernet2/0
11.11.11.11 (PE1)	1	FULL/DR	00:00:36	10.0.1.6	FastEthernet1/0
1.1.1.1 (P1)	1	FULL/BDR	00:00:34	10.0.0.1	FastEthernet0/0

P2 est connecté à trois voisins OSPF (22.22.22.22, 11.11.11.11 et 1.1.1.1).
Deux voisins sont **DR**, et l'autre est **BDR**.
P2 échange bien des routes avec ces voisins.(état FULL)

MPLS :

Premier voisin (22.22.22.22)(PE2)
Peer LDP Ident: 22.22.22.22:0 → Le voisin MPLS a l'ID LDP 22.22.22.22.
Local LDP Ident: 2.2.2.2:0 → L'ID LDP local de P2 est 2.2.2.2.
TCP connection: 22.22.22.22.36130 - 2.2.2.2.646 → Session TCP entre les routeurs pour échanger des labels.
State: Open; Msgs sent/rcvd: 1746/1747 → La session LDP est active.
Uptime: 10d1h → LDP est actif depuis 10 jours et 1 heure.
LDP discovery source :

- Interface FastEthernet2/0, IP source 10.0.2.6.
- Les adresses associées sont 10.0.2.2 et 22.22.22.22.

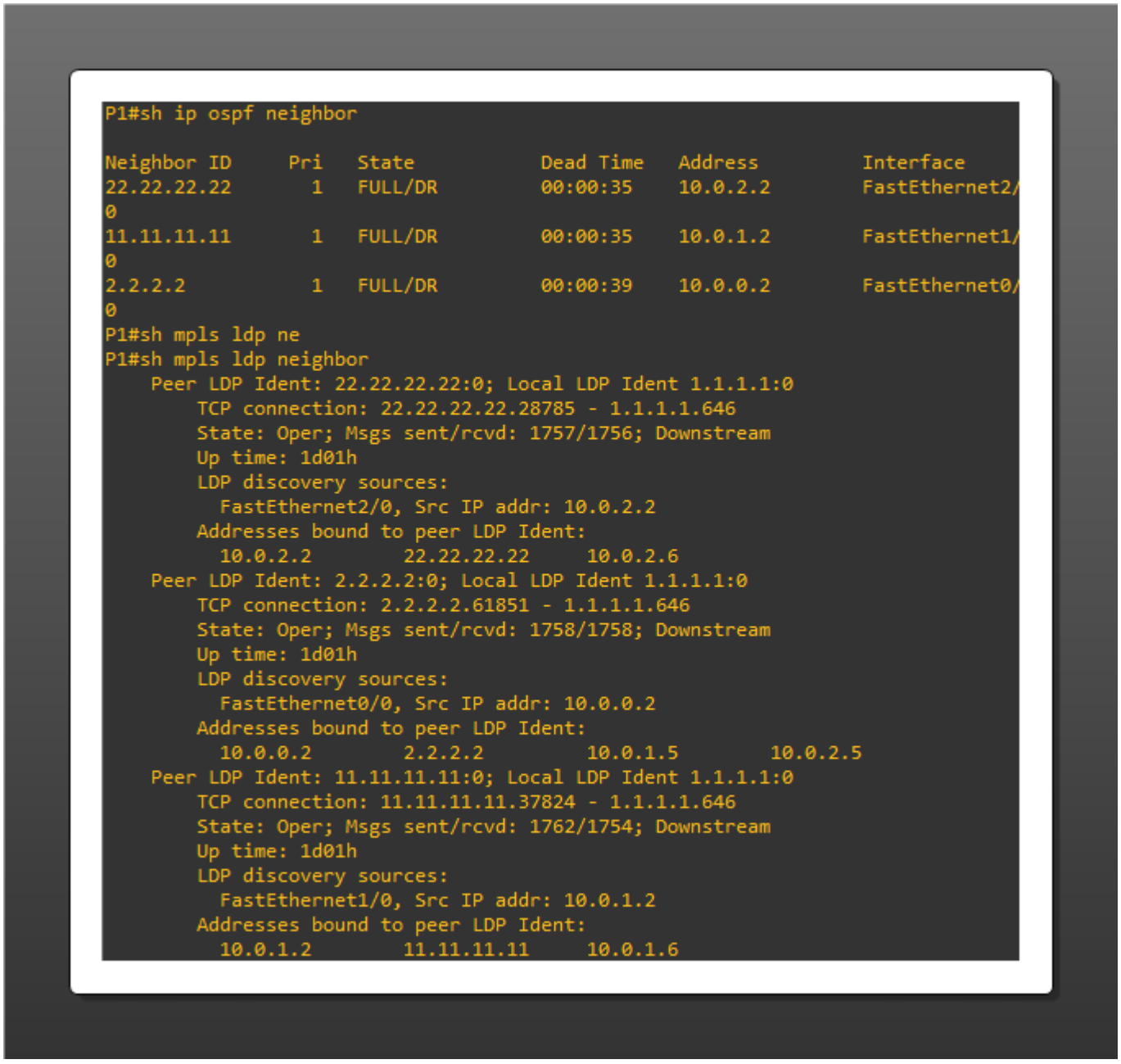
Deuxième voisin (1.1.1.1)
Peer LDP Ident: 1.1.1.1:0 → Le voisin MPLS a l'ID LDP 1.1.1.1.(P1)
LDP discovery source :

- Interface FastEthernet0/0, IP source 10.0.0.1.
- Les adresses associées sont 10.0.0.1, 1.1.1.1, 10.0.1.1.

Troisième voisin (11.11.11.11)
Peer LDP Ident: 11.11.11.11:0 → Le voisin MPLS a l'ID LDP 11.11.11.11.(PE1)
LDP discovery source :

- Interface FastEthernet1/0, IP source 10.0.1.6.
- Les adresses associées sont 10.0.1.2, 11.11.11.11, 10.0.1.6.

b) P1



```
P1#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
22.22.22.22       1   FULL/DR         00:00:35    10.0.2.2        FastEthernet2/
0
11.11.11.11       1   FULL/DR         00:00:35    10.0.1.2        FastEthernet1/
0
2.2.2.2           1   FULL/DR         00:00:39    10.0.0.2        FastEthernet0/
0
P1#sh mpls ldp ne
P1#sh mpls ldp neighbor
    Peer LDP Ident: 22.22.22.22:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 22.22.22.22.28785 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 1757/1756; Downstream
        Up time: 1d01h
        LDP discovery sources:
          FastEthernet2/0, Src IP addr: 10.0.2.2
        Addresses bound to peer LDP Ident:
          10.0.2.2        22.22.22.22     10.0.2.6
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 2.2.2.2.61851 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 1758/1758; Downstream
        Up time: 1d01h
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.0.0.2
        Addresses bound to peer LDP Ident:
          10.0.0.2        2.2.2.2         10.0.1.5        10.0.2.5
    Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 11.11.11.11.37824 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 1762/1754; Downstream
        Up time: 1d01h
        LDP discovery sources:
          FastEthernet1/0, Src IP addr: 10.0.1.2
        Addresses bound to peer LDP Ident:
          10.0.1.2        11.11.11.11     10.0.1.6
```

OSPF :

Neighbor ID	Pri	State	Dead Time	Address	Interface
22.22.22.22 (PE2)	1	FULL/DR	00:00:36	10.0.2.2	FastEthernet2/0
11.11.11.11 (PE1)	1	FULL/DR	00:00:36	10.0.1.2	FastEthernet1/0

2.2.2.2 (P2)	1	FULL/BDR	00:00:34	10.0.0.1	FastEthernet0/0

P1 est connecté à trois voisins OSPF (22.22.22.22, 11.11.11.11 et 1.1.1.1).
Deux voisins sont **DR**, et l'autre est **BDR**.
P1 échange bien des routes avec ces voisins.

MPLS :

Premier voisin (22.22.22.22) :
Peer LDP Ident: 22.22.22.22:0 → Le voisin MPLS a l'ID LDP 22.22.22.22. (Il s'agit de PE2)
Local LDP Ident: 1.1.1.1:0 → L'ID LDP local de P1 est 1.1.1.1.
TCP connection: 22.22.22.22.28785 - 1.1.1.1.646 → Session TCP active.
State: Open; Msgs sent/rcvd: 1757/1756 → L'échange de labels est actif.
Uptime: 10d1h → La session LDP fonctionne depuis 10 jours et 1 heure.
LDP discovery sources :

- Interface FastEthernet2/0, IP source 10.0.2.2.
- Adresses associées : 10.0.2.2, 22.22.22.22, 10.0.2.6.

Deuxième voisin (2.2.2.2) :
Peer LDP Ident: 2.2.2.2:0 → Le voisin MPLS a l'ID LDP 2.2.2.2. (Il s'agit de P2)
LDP discovery sources :

- Interface FastEthernet0/0, IP source 10.0.0.2.
- Adresses associées : 10.0.0.2, 2.2.2.2, 10.0.1.5, 10.0.2.5.

Troisième voisin (11.11.11.11) :
Peer LDP Ident: 11.11.11.11:0 → Le voisin MPLS a l'ID LDP 11.11.11.11. (Il s'agit de PE1)
LLDP discovery sources :

- Interface FastEthernet1/0, IP source 10.0.1.2.
- Adresses associées : 10.0.1.2, 11.11.11.11, 10.0.1.6.

c) PE2

```
PE2#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/BDR        00:00:38    10.0.2.5        FastEthernet0/
1
1.1.1.1           1   FULL/BDR        00:00:38    10.0.2.1        FastEthernet0/
0
PE2#sh mpls ldp ne
PE2#sh mpls ldp neighbor
    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 22.22.22.22:0
        TCP connection: 1.1.1.1.646 - 22.22.22.22.44514
        State: Oper; Msgs sent/rcvd: 15/14; Downstream
        Up time: 00:02:45
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.0.2.1
        Addresses bound to peer LDP Ident:
          10.0.0.1        1.1.1.1         10.0.1.1        10.0.2.1
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 22.22.22.22:0
        TCP connection: 2.2.2.2.646 - 22.22.22.22.54901
        State: Oper; Msgs sent/rcvd: 15/15; Downstream
        Up time: 00:02:44
        LDP discovery sources:
          FastEthernet0/1, Src IP addr: 10.0.2.5
        Addresses bound to peer LDP Ident:
          10.0.0.2        2.2.2.2         10.0.1.5        10.0.2.5
```

OSPF :

Neighbor ID	State	Address	Interface
2.2.2.2 (P2)	FULL/BDR	10.0.2.5	FastEthernet0/1
1.1.1.1 (P1)	FULL/DBR	10.0.2.1	FastEthernet0/0

PE2 a deux voisins OSPF (2.2.2.2 et 1.1.1.1) qui correspondent bien à P2 et P1, chacun connecté via une interface FastEthernet.

MPLS :

<u>Premier voisin :</u>

Peer LDP Ident: 1.1.1.1:0 → Le voisin MPLS a l'ID LDP 1.1.1.1.

Local LDP Ident: 22.22.22.22:0 → L'ID LDP local de PE2 est 22.22.22.22.

TCP connection: 1.1.1.1.646 - 22.22.22.22.44514 → Connexion TCP utilisée pour LDP.

State: Oper; Msgs sent/rcvd: 15/14 → LDP est opérationnel, avec 15 messages envoyés et 14 reçus.

LDP discovery source :

- Interface FastEthernet0/0 (IP source 10.0.2.1) est utilisée pour découvrir ce voisin LDP.
- Les adresses IP échangées sont 1.1.1.1, 10.0.1.1 et 10.0.2.1.

Deuxième voisin :

Peer LDP Ident: 2.2.2.2:0 → Le voisin MPLS a l'ID LDP 2.2.2.2.

LDP discovery source :

- Interface FastEthernet0/1 (IP source 10.0.2.5) utilisée pour découvrir ce voisin LDP.
- Les adresses échangées sont 2.2.2.2, 10.0.1.5 et 10.0.2.5.

d) PE1

```
PE1#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/BDR        00:00:34    10.0.1.5        FastEthernet0/
1
1.1.1.1           1   FULL/BDR        00:00:38    10.0.1.1        FastEthernet0/
0
PE1#sh mpls ldp ne
PE1#sh mpls ldp neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 11.11.11.11:0
        TCP connection: 2.2.2.2.646 - 11.11.11.11.65344
        State: Oper; Msgs sent/rcvd: 61/61; Downstream
        Up time: 00:43:42
        LDP discovery sources:
          FastEthernet0/1, Src IP addr: 10.0.1.5
        Addresses bound to peer LDP Ident:
          10.0.0.2        2.2.2.2         10.0.1.5        10.0.2.5
    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 11.11.11.11:0
        TCP connection: 1.1.1.1.646 - 11.11.11.11.62728
        State: Oper; Msgs sent/rcvd: 61/62; Downstream
        Up time: 00:43:41
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.0.1.1
        Addresses bound to peer LDP Ident:
          10.0.0.1        1.1.1.1         10.0.1.1        10.0.2.1
```

OSPF :

Neighbor ID	State	Address	Interface
2.2.2.2 (P2)	FULL/BDR	10.0.1.5	FastEthernet0/1
1.1.1.1 (P1)	FULL/DBR	10.0.1.1	FastEthernet0/0

PE1 a deux voisins OSPF (2.2.2.2 et 1.1.1.1) qui correspondent bien à P2 et P1, chacun connecté via une interface FastEthernet.

MPLS :

Premier voisin :

Peer LDP Ident: 2.2.2.2:0 → Le voisin MPLS a l'ID LDP 2.2.2.2.

Local LDP Ident: 11.11.11.11:0 → L'ID LDP local de PE1 est 11.11.11.11.

TCP connection: 2.2.2.2.646 - 11.11.11.11.65344 → Connexion TCP utilisée pour LDP.

State: Oper; Msgs sent/rcvd: 61/61 → LDP est opérationnel, avec 61 messages envoyés et 14 reçus.

LDP discovery source :

- Interface FastEthernet0/1 (IP source 10.0.1.5) est utilisée pour découvrir ce voisin LDP.
- Les adresses IP échangées sont 1.1.1.1, 10.0.1.1 et 10.0.2.1.

Deuxième voisin :

Peer LDP Ident: 1.1.1.1:0 → Le voisin MPLS a l'ID LDP 1.1.1.1

LDP discovery source :

- Interface FastEthernet0/0 (IP source 10.0.1.1) utilisée pour découvrir ce voisin LDP.
- Les adresses échangées sont 2.2.2.2, 10.0.1.5 et 10.0.2.5.

Toutes nos sessions OSPF et MPLS sont montées.

Test ping

On test donc le ping entre les routeurs PE :

PE2 vers PE1

```
PE2#ping 11.11.11.11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.11.11.11, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/43/56 ms
```

```
PE2#ping 10.0.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/56/64 ms
PE2#ping 10.0.1.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/59/68 ms
PE2#
```

Toutes les interfaces se ping bien.

PE1 vers PE2

```
PE1#ping 22.22.22.22

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 22.22.22.22, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/39/60 ms
```

```
PE1#ping 10.0.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/56/64 ms
PE1#ping 10.0.2.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/60/76 ms
PE1#
```

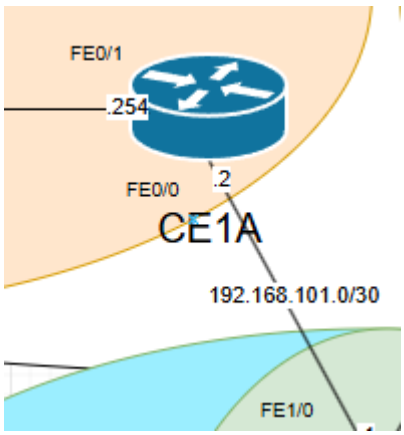
Toutes les interfaces se ping bien.

Notre cœur de réseau MPLS/OSPF est donc fonctionnel.

4. Configuration BGP et VRF

MP-BGP doit être configuré dans l'address-family vpnv4 pour établir les liaisons entre routeurs PE. L'AS choisi sera 1234

a) CE1A



Commandes :

```
router bgp 65001
 no synchronization
 bgp log-neighbor-changes
 network 1.1.1.1 mask 255.255.255.255
 network 101.101.101.101 mask 255.255.255.255

 neighbor 192.168.101.1 remote-as 1234
```

(Annonce PE1 comme neighbor dans l'AS 1234)

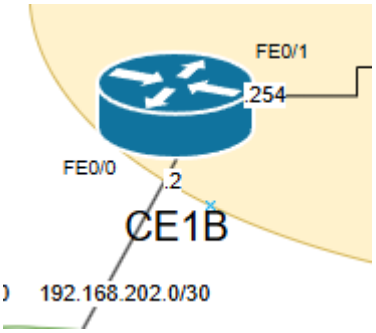
Vérification avec sh bgp all summary :

```
BGP activity 8/0 prefixes, 8/0 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.101.1   4  1234     111     111        9    0    0 01:47:52        4
```

La session BGP est montée avec PE1

b) CE1B

Commandes :




```
router bgp 65001
 no synchronization
 bgp log-neighbor-changes
 network 1.1.1.1 mask 255.255.255.255
 network 102.102.102.102 mask 255.255.255.255

 network 192.168.10.0 mask 255.255.255.0
 network 192.168.20.0 mask 255.255.255.0
```

(Annonce les réseau locaux en BGP)

```
 neighbor 192.168.202.1 remote-as 1234
```

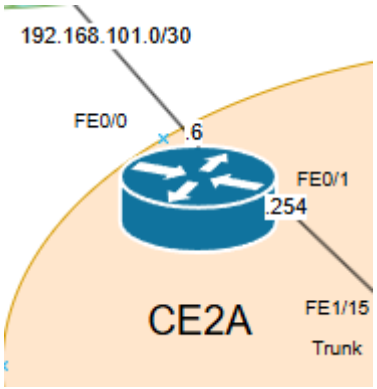
(Annonce PE1 comme neighbor dans l'AS 1234)

Vérification avec sh bgp all summary :

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.202.1   4  1234     123     124        9    0    0 01:59:03        4
CE1B#
```

La session BGP est bien montée avec PE1

c) CE2A



<u>Commandes :</u>

```
router bgp 65002
 no synchronization
 bgp log-neighbor-changes
 network 21.21.21.21 mask 255.255.255.255
 network 101.101.101.111 mask 255.255.255.255

 network 192.168.30.0 mask 255.255.255.0
 network 192.168.40.0 mask 255.255.255.0
```

(Annonce les réseau locaux en BGP)

```
 neighbor 192.168.101.5 remote-as 1234
```

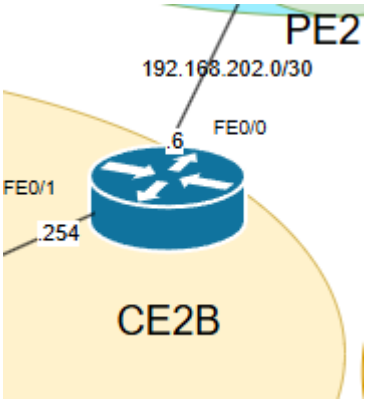
(Annonce PE2 comme neighbor dans l'AS 1234)

Vérification avec sh bgp all summary :

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.101.5   4  1234     137     137        9    0    0 02:13:31        4
```

La session BGP est bien montée avec PE2

d) CE2B



Commandes :

router bgp 65002
 network 22.22.22.22 mask 255.255.255.255
 network 102.102.102.112 mask 255.255.255.255

network 192.168.30.0
network 192.168.40.0
(Annonce les réseau locaux en BGP)

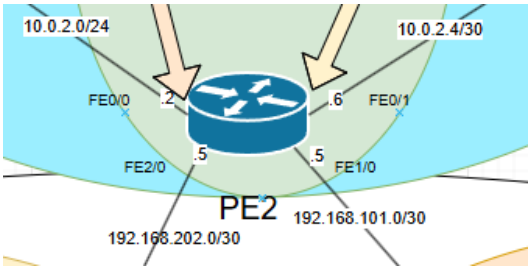
neighbor 192.168.202.5 remote-as 1234
(Annonce PE2 comme neighbor dans l'AS 1234)

Vérification avec sh bgp all summary :

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.202.5   4  1234     140     141        9    0    0 02:16:04        4
```

La session BGP est bien montée avec PE2

e) PE2



Commandes :

router bgp 1234
neighbor 11.11.11.11 remote-as 1234
(Définit PE1 comme voisin BGP avec l'adresse IP 11.11.11.11 et spécifie qu'il appartient au même AS (1234))
neighbor 11.11.11.11 update-source Loopback0
(Indique que les mises à jour BGP envoyées à ce voisin doivent utiliser l'interface Loopback0 comme source. Cela assure une meilleure stabilité, car Loopback0 reste toujours active même si un lien physique tombe.)
address-family vpnv4
(Commence la configuration de l'adresse de famille VPNv4, qui est utilisée pour le routage des VPN MPLS)
neighbor 11.11.11.11 activate
(Active le voisin BGP pour l'adresse de famille VPNv4, permettant l'échange de routes.)

neighbor 11.11.11.11 send-community extended

(Configure le voisin pour envoyer des communautés étendues avec les mises à jour de routage. Ceci est essentiel pour MPLS VPN, car les routes des clients doivent être taguées avec leurs VRF respectives..)

address-family ipv4 vrf ClientB

(Active BGP pour le VRF ClientB (une table de routage isolée pour le client B)

neighbor 192.168.202.6 remote-as 65002

(Definit CE2B comme voisin dans l'AS 65002)

exit-address-family

address-family ipv4 vrf ClientA

(Active BGP pour le VRF ClientA (une table de routage isolée pour le client A)

neighbor 192.168.101.6 remote-as 65002

neighbor 192.168.101.6 activate

(Definit CE2A comme voisin dans l'AS 65002)

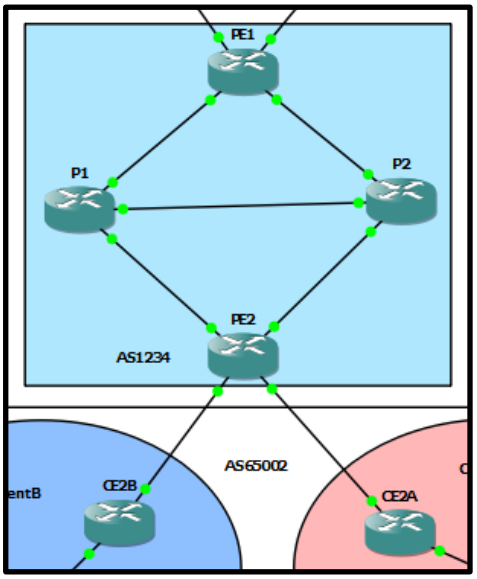
exit-address-family

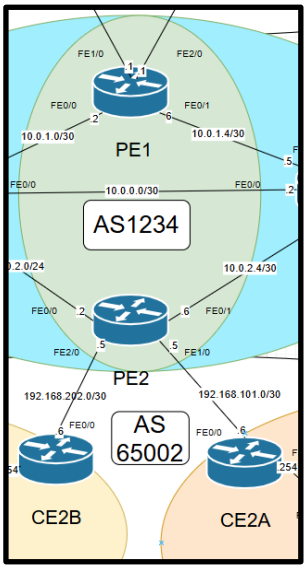
Vérification avec la commande **sh bgp all summary**:

```
BGP activity 16/0 prefixes, 16/0 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
11.11.11.11     4  1234      70      70       25    0    0 01:05:46        8
192.168.101.6   4 65002      69      69       25    0    0 01:05:39        4
192.168.202.6   4 65002      70      69       25    0    0 01:05:43        4
PE2#
```

On voit bien que PE1 : 11.11.11.11 est bien annoncé dans son AS 1234

On voit également les deux routeur CE2B et CE2A, chacun dans l'AS 65002





Vérification des VRF avec :

```
PE2#sh ip vrf
  Name                             Default RD          Interfaces
  ClientA                          1234:101            Fa1/0
  ClientB                          1234:202            Fa2/0
```

On voit bien nos deux VRF, une pour chaque client,

Affichage de la table de routage réserve au client A :

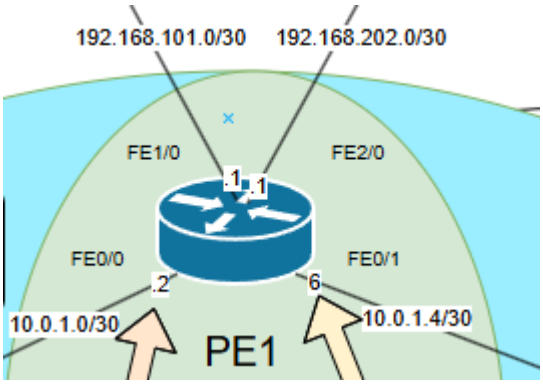
```
PE2#sh ip route vrf ClientA

      1.0.0.0/32 is subnetted, 1 subnets
B        1.1.1.1 [200/0] via 11.11.11.11, 02:28:24
B
      101.0.0.0/32 is subnetted, 2 subnets
B        101.101.101.101 [200/0] via 11.11.11.11, 02:28:24
B        101.101.101.111 [20/0] via 192.168.101.6, 02:28:32
      21.0.0.0/32 is subnetted, 1 subnets
B        21.21.21.21 [20/0] via 192.168.101.6, 02:28:32
B
B     192.168.40.0/24 [20/0] via 192.168.101.6, 02:28:34
B     192.168.20.0/24 [200/0] via 11.11.11.11, 02:28:26
      192.168.101.0/30 is subnetted, 1 subnets
C        192.168.101.4 is directly connected, FastEthernet1/0
PE2#
```

On voit bien les routes vers les réseaux locaux de chaque sites du client A. Échangé en iBGP [20/0] et eBGP [200/0]

Même principe pour le client B

f) PE1

Même configuration que PE2, seul l'adresse des neighbor change :



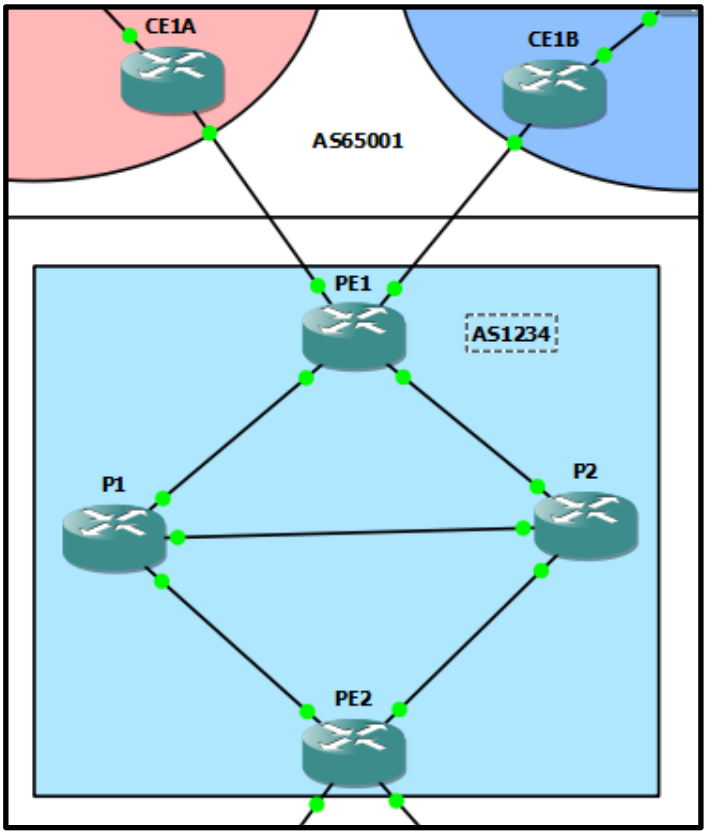

```
router bgp 1234
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 22.22.22.22 remote-as 1234
 neighbor 22.22.22.22 update-source Loopback0
 address-family vpnv4
  neighbor 22.22.22.22 activate
  neighbor 22.22.22.22 send-community extended
 exit-address-family

address-family ipv4 vrf ClientB
  neighbor 192.168.202.2 remote-as 65001
(Définit CE1B comme voisin dans l'AS 65001)
 exit-address-family

 address-family ipv4 vrf ClientA
  neighbor 192.168.101.2 remote-as 65001
(Définit CE1A comme voisin dans l'AS 65001)
 exit-address-family
```

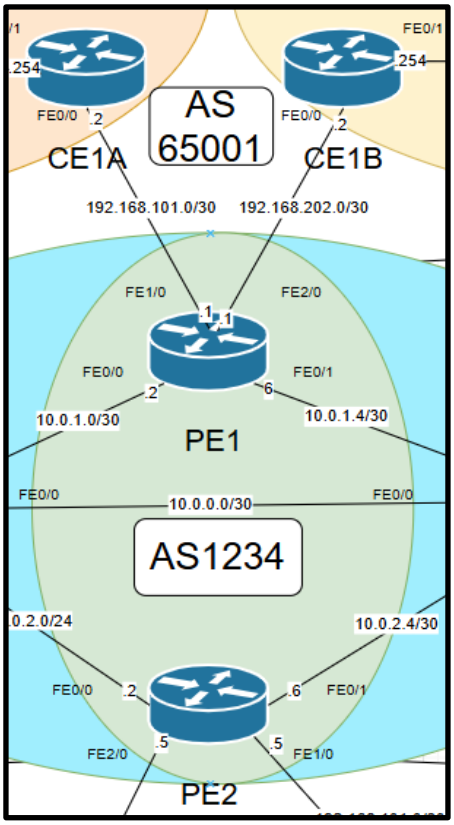
Vérification avec sh bgp all summary :

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
22.22.22.22     4  1234      69      69       25    0    0 01:04:03        8
192.168.101.2   4 65001      68      68       25    0    0 01:04:00        4
192.168.202.2   4 65001      68      67       25    0    0 01:03:59        4
PE1#
```

On voit bien que PE2 : 22.22.22.22 est bien annoncé dans son AS 1234

On voit également les deux routeur CE1B et CE1A, chacun dans l'AS 65001





Vérification des VRF avec :

```
PE1#sh ip vrf
  Name                             Default RD          Interfaces
  ClientA                          1234:101            Fa1/0
  ClientB                          1234:202            Fa2/0
```

Affichage de la table de routage réserve au client B :

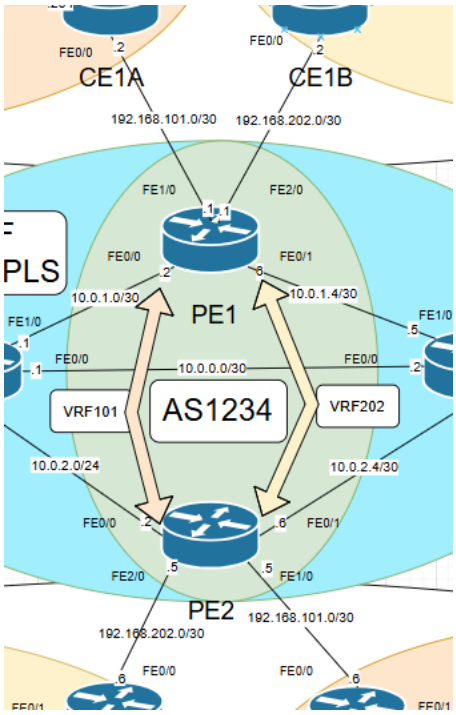
```
PE1#sh ip route vrf ClientB
```

```
      102.0.0.0/32 is subnetted, 2 subnets
B        102.102.102.102 [20/0] via 192.168.202.2, 02:56:25
B        102.102.102.112 [200/0] via 22.22.22.22, 02:56:15
      1.0.0.0/32 is subnetted, 1 subnets
B        1.1.1.1 [20/0] via 192.168.202.2, 02:56:25
B     192.168.30.0/24 [200/0] via 22.22.22.22, 02:56:15
B     192.168.10.0/24 [20/0] via 192.168.202.2, 02:56:25
B     192.168.40.0/24 [200/0] via 22.22.22.22, 02:56:15
      22.0.0.0/32 is subnetted, 1 subnets
B        22.22.22.22 [200/0] via 22.22.22.22, 02:56:17
B     192.168.20.0/24 [20/0] via 192.168.202.2, 02:56:27
      192.168.0.0/24 is subnetted, 1 subnets
C        192.168.202.0 is directly connected, FastEthernet2/0
PE1#
```

On voit bien les routes vers les réseaux locaux de chaque site du client B. Échangé en iBGP [20/0] et eBGP [200/0]

Même principe pour le client A

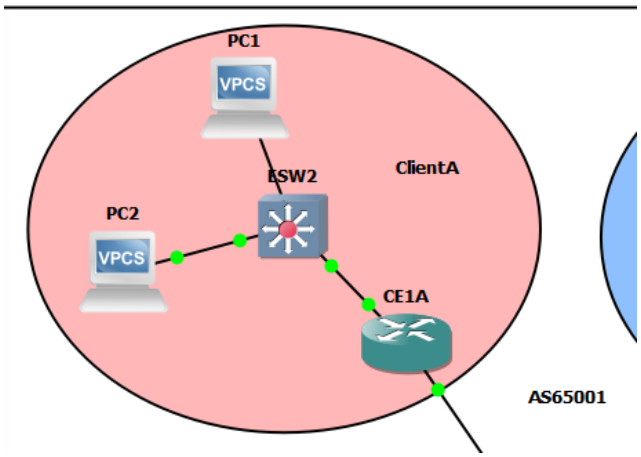
On a donc :



Les clients A et B ont chacun une VRF dédiée : 101 pour le client A et 202 pour le client B. Ces VRF contiennent des routes échangées en eBGP et iBGP. Permettant aux sites distants des clients de se joindre. Tout en garantissant une séparation entre le réseau du client A et celui du client B qui ne peuvent pas se joindre entre eux.

B. LAN Client

1. VLANs



A - Sur les routeurs CE

Pour l'interface FE0/1.10 du routeur CE1A :

```
interface FastEthernet0/1.10
 encapsulation dot1Q 10
 ip address 192.168.10.254 255.255.255.0
 ipv6 address 2001:DB8:10::1/64
!
```

- On crée une sous-interface (.10), qui sera utilisée pour le VLAN 10.
- Puis on active l'encapsulation 802.1Q et associe cette sous-interface au VLAN 10.
  Cela signifie que cette interface n'achemine que les trames marquées avec l'ID VLAN 10.
- On associe l'IP 192.168.10.254/24 à cette sous-interface, faisant d'elle la passerelle pour le VLAN 10.
- On configure l'adresse IPv6 2001:DB8:10::1/64 pour cette sous-interface.
- On utilise SLAAC (Stateless Address Autoconfiguration)Le routeur annonce un préfixe IPv6 via le Router Advertisement (RA) pour permettre aux hôtes de s'auto-configurer.

Pareil pour l'interface **FE0/1.20** du routeur **CE1A** :

```
interface FastEthernet0/1.20
 encapsulation dot1Q 20
 ip address 192.168.20.254 255.255.255.0
 ipv6 address 2001:DB8:20::1/64
!
```

- On crée une sous-interface (.20), qui sera utilisée cette fois pour le VLAN 20.
- Puis on active l'encapsulation 802.1Q et associe cette sous-interface au VLAN 20.
  Cela signifie que cette interface n'achemine que les trames marquées avec l'ID VLAN 20.
- On associe l'IP 192.168.20.254/24 à cette sous-interface, faisant d'elle la passerelle pour le VLAN 20.
- On configure l'adresse IPv6 2001:DB8:20::1/64 pour cette sous-interface.
- On utilise SLAAC (Stateless Address Autoconfiguration)Le routeur annonce un préfixe IPv6 via le Router Advertisement (RA) pour permettre aux hôtes de s'auto-configurer.

B - Sur les switchs clients

Configuration du port **FastEthernet1/0** du switch ESW2 :

```
interface FastEthernet1/0
 switchport access vlan 20
 duplex full
 speed 100
!
```

- On configure ce port en mode accès (access) et l'associe au VLAN 20.
  Cela signifie que tout appareil branché sur ce port sera automatiquement membre du VLAN 20.
- On force le port en mode duplex intégral (*full-duplex*), ce qui permet une communication simultanée en émission et en réception.
- Et on fixe la vitesse du port à 100 Mbps.

Pareil pour la configuration du port **FastEthernet1/1** du switch ESW2 :

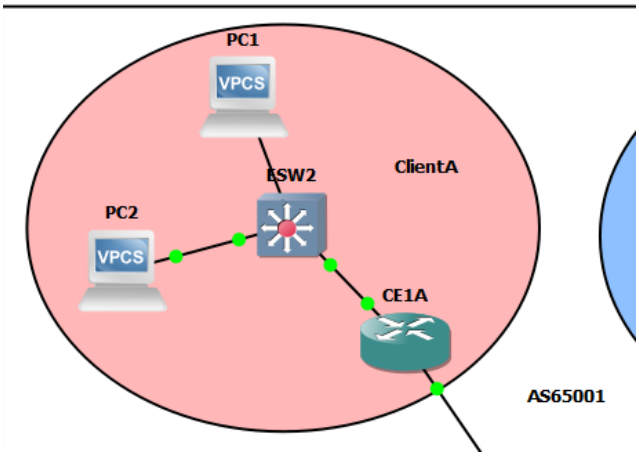
```
interface FastEthernet1/1
 switchport access vlan 10
 duplex full
 speed 100
```

- On configure ce port en mode accès (access) et l'associe au VLAN 10.
  Cela signifie que tout appareil branché sur ce port sera automatiquement membre du VLAN 10.
- On force le port en mode duplex intégral (*full-duplex*), ce qui permet une communication simultanée en émission et en réception.
- Et on fixe la vitesse du port à 100 Mbps.

Ce switch est également relié au routeur CE1A via un port trunk, permettant à l'interface de transporter plusieurs VLANs sur un même lien.

Ainsi, même si les deux VLANs existent sur le même client (le switch), ils fonctionnent comme deux réseaux séparés (pouvant tout de même communiquer entre eux via le routage Inter-VLAN), garantissant une segmentation et une isolation du trafic.

2. DHCP sur les routeurs CE



Sur CE1A :

Nous configurons un serveur DHCP sur ce routeur avec deux pool DHCP :

- VLAN10 (192.168.10.0/24) avec 192.168.10.254 comme passerelle
- VLAN20 (192.168.20.0/24) avec 192.168.20.254 comme passerelle
- Les adresses IP des passerelles sont exclues de l'attribution DHCP

Cela va permettre aux machines des VLANs 10 et 20 d'obtenir une adresse IP automatiquement via DHCP.

```
no ip dhcp use vrf connected
ip dhcp excluded-address 192.168.10.254
ip dhcp excluded-address 192.168.20.254
!
```

Les adresses 192.168.10.254 et 192.168.20.254 ne seront pas attribuées aux clients DHCP étant donné qu'il s'agit des passerelles par défauts.

Cette section définit un pool DHCP nommé "VLAN10" :

```
ip dhcp pool VLAN10
   network 192.168.10.0 255.255.255.0
   default-router 192.168.10.254
!
```

- Il attribue des adresses dans le sous-réseau **192.168.10.0/24**.
- La passerelle par défaut pour les clients DHCP est **192.168.10.254**

Même logique que pour VLAN10, mais cette fois pour **VLAN20** :

```
ip dhcp pool VLAN20
   network 192.168.20.0 255.255.255.0
   default-router 192.168.20.254
!
```

- Sous-réseau **192.168.20.0/24**
- Passerelle par défaut **192.168.20.254**

Sur PC1 :
Nous effectuons la commande "dhcp" afin qu'une adresse lui soit attribué (il faut que le routeur CE1A soit allumé en parallèle)

```
PC1> dhcp
DORA IP 192.168.10.1/24 GW 192.168.10.254
```

Nous observons qu'une adresse lui est bien attribuée, le serveur DHCP étant bien configuré.

Nous effectuons ensuite ce paramétrage sur tous les routeurs clients CE.

SLAAC :

```
PC1> ip auto
GLOBAL SCOPE      : 2001:db8:10:0:2050:79ff:fe66:6802/64
ROUTER LINK-LAYER : c2:01:51:6c:00:01
```

```
PC5> ip auto
GLOBAL SCOPE      : 2001:db8:40:0:2050:79ff:fe66:6801/64
ROUTER LINK-LAYER : c2:08:21:10:00:01
```

Les PC récupères bien une ipv6 avec le préfixe de leur VLAN

3. ACLs

A - Bloquer les protocoles non-sécurisés

a - Implémentations

Ici nous allons bloquer l'accès à tous les protocoles en autorisants ceux qui nous sont nécessaires un par un :

```
ip access-list extended SECURE_ONLY
 permit udp any any eq bootps
 permit udp any any eq bootpc
 permit tcp any any eq 22
 permit tcp any any eq 443
 permit icmp any any echo
 permit icmp any any echo-reply
 deny   ip any any
!
```

permit tcp any any eq 22 → Autorise SSH
permit tcp any any eq 443 → Autorise HTTPS
permit icmp any any echo → Autorise les requêtes de ping envoyées.
permit icmp any any echo-reply → Autorise les réponses aux pings.
permit udp any any eq 67 → Autorise DHCP (requêtes serveur)
permit udp any any eq 68 → Autorise DHCP (réponses client)

Le reste du trafic est toujours bloqué.

```
interface FastEthernet0/1.10
 encapsulation dot1Q 10
 ip address 192.168.10.254 255.255.255.0
 ip access-group SECURE_ONLY in
 ipv6 address 2001:DB8:10::1/64
!
interface FastEthernet0/1.20
 encapsulation dot1Q 20
 ip address 192.168.20.254 255.255.255.0
 ip access-group SECURE_ONLY in
 ipv6 address 2001:DB8:20::1/64
!
```

ip access-group SECURE_ONLY in → Permet d'attribuer les ACLs aux interfaces voulues.

Nous effectuons ces configurations ACLs sur toutes les interfaces des routeurs CE du réseau.

Résultat :

- **SSH (port 22) est autorisé**
- **HTTPS (port 443) est autorisé**
- **Les pings fonctionnent**
- **Tout le reste est bloqué**

b - Tests

```
PC4> dhcp
DORA IP 192.168.10.1/24 GW 192.168.10.254

PC4> ping 192.168.30.1
192.168.30.1 icmp_seq=1 timeout
192.168.30.1 icmp_seq=2 timeout
84 bytes from 192.168.30.1 icmp_seq=3 ttl=59 time=121.946 ms
84 bytes from 192.168.30.1 icmp_seq=4 ttl=59 time=135.570 ms
84 bytes from 192.168.30.1 icmp_seq=5 ttl=59 time=151.706 ms
```

ex : ping de pc4 vers pc8 (VLAN10 à VLAN30) sans bloquer les requêtes ICMP

```
C4> ping 192.168.30.1
192.168.10.254 icmp_seq=1 ttl=255 time=15.138 ms (ICMP type:3, code:13, Communication administratively prohibited)
192.168.10.254 icmp_seq=2 ttl=255 time=16.722 ms (ICMP type:3, code:13, Communication administratively prohibited)
192.168.10.254 icmp_seq=3 ttl=255 time=0.967 ms (ICMP type:3, code:13, Communication administratively prohibited)
192.168.10.254 icmp_seq=4 ttl=255 time=17.083 ms (ICMP type:3, code:13, Communication administratively prohibited)
192.168.10.254 icmp_seq=5 ttl=255 time=16.446 ms (ICMP type:3, code:13, Communication administratively prohibited)

C4>
```

ex : ping de pc4 vers pc8 (VLAN10 à VLAN30) en bloquant les requêtes ICMP via les ACLs

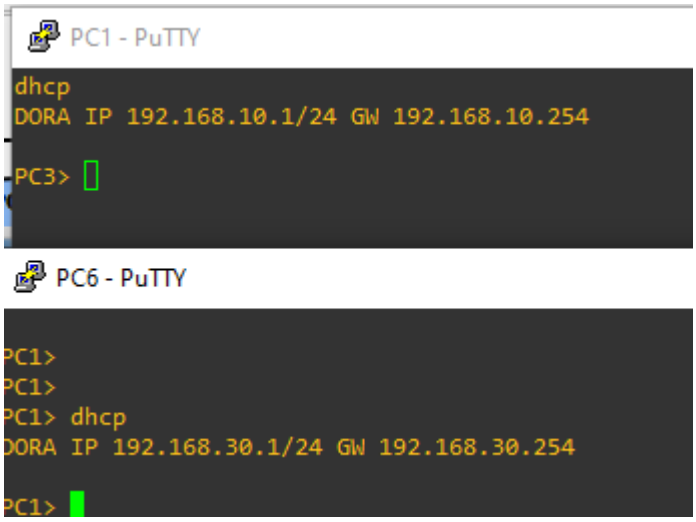
Après avoir appliquer l'acl sur toutes les interfaces des routeurs CE nous avons bloquer tous les ports non sécurisés, tels que Telnet,HTTP afin de ne laisser accessible que SSH et HTTPS, renforçant ainsi la sécurité de notre réseau en réduisant les risques liés à l'utilisation de protocoles obsolètes et vulnérables.

Note : Il est possible et facile de rajouter une ACL pour autoriser un protocole dont le client aurait besoin .

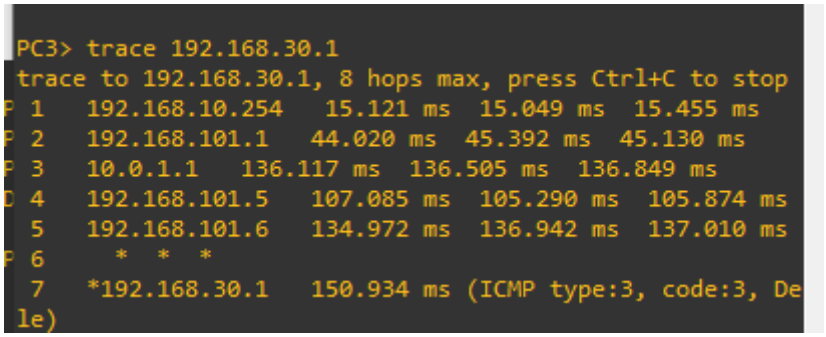
4. IPSEC

Pourquoi IPSEC ?

Pour le moment lorsque l'on effectue un ping entre 2 PC de 2 sites différents, on peut facilement voir son chemin dans le backbone . On va essayer ici de faire un trace de PC1 (192.168.10.1) vers PC6(192.168.30.1) :

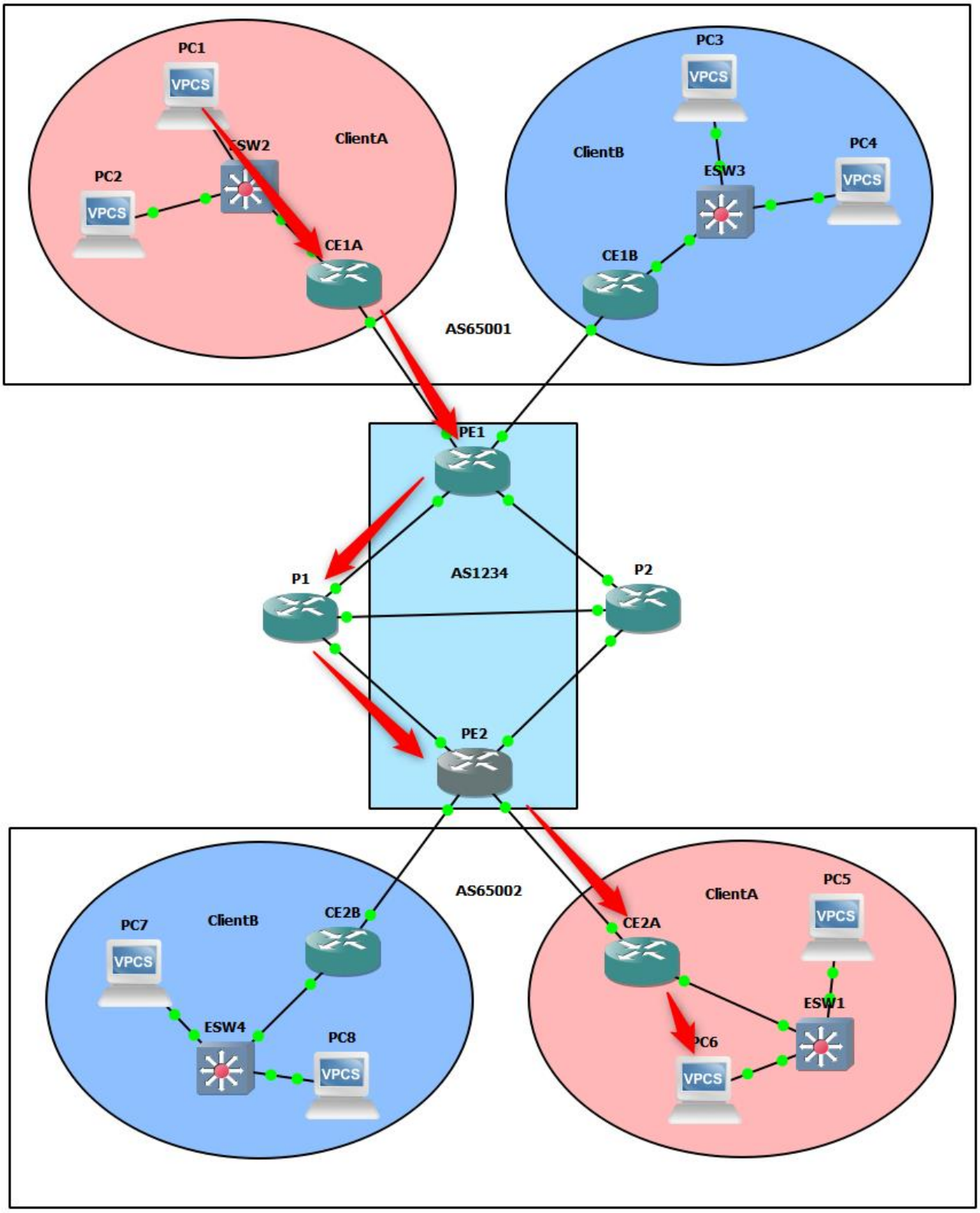


Trace de PC1 vers PC6 (ne pas prêter attention au nom du PC qui n'est pas à jour dans GNS)



1. Passerelle de CE1A
2. Interface de PE1 (passerelle de CE1A)
3. Interface de P1
4. Interface de PE2
5. Interface de CE2A
6. Adresse de PC6

Le chemin du paquet est le suivant :



Le problème est que l'on voit toutes les interfaces du backbone avec un simple trace. Il suffirait de capturer le trafic avec Wireshark pour tout voir. Il faudrait donc mettre en place un VPN IPSEC afin d'encapsuler cela. Cela permettrait également au client de masquer ces IP local sur le backbone.

IPSEC est un protocole de sécurisation des communications sur un réseau IP. Il fonctionne au niveau de la couche réseau (couche 3 du modèle OSI) et permet de sécuriser les échanges de données entre deux hôtes, deux réseaux ou un hôte et un réseau distant.

Intérêt d'IPsec

- Confidentialité : Grâce au chiffrement des paquets IP, IPsec empêche l'interception et la lecture des données transmises.
- Authentification : Il assure que les données proviennent d'une source fiable grâce à des mécanismes d'authentification.
- Intégrité : Il garantit que les données n'ont pas été altérées en cours de transmission.
- Protection contre les attaques : IPsec protège contre les attaques de type "replay" et le spoofing en utilisant des numéros de séquence et des clés de session dynamiques.

Configuration IPSEC entre CE1A et CE2A :

On vérifie que les Loopback se ping :

```
CE1A#ping 21.21.21.21 source Loopback1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 21.21.21.21, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 96/114/148 ms
CE1A#
```

```
CE2A#ping 1.1.1.1 source Loopback1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 21.21.21.21
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/113/136 ms
CE2A#
```

Commande sur CE2A et CE1A :

PHASE 1

Objectif : Authentifier les routeurs et établir un canal sécurisé pour échanger les clés de chiffrement.

Configuration sur les deux routeurs :

crypto isakmp policy 10
encr aes
(Chiffrement AES)
authentication pre-share
(Authentification avec clé pré-partagée)
group 5
(Groupe Diffie-Hellman 5 pour l'échange de clés sécurisé)

Configuration de l'identité et de la clé pré-partagée :

CE2A :
crypto isakmp identity address
crypto isakmp key vpnuser address 1.1.1.1
CE1A :
crypto isakmp identity address
crypto isakmp key vpnuser address 21.21.21.21

- Chaque routeur identifie son pair via son adresse IP Loopback et utilise la clé pré-partagée vpnuser.
- CE2A attend une connexion de 1.1.1.1 (CE1A).
- CE1A attend une connexion de 21.21.21.21 (CE2A).

PHASE 2

Objectif : Définir comment le trafic sera sécurisé après la Phase 1.(définit les algorithmes de chiffrement et d'intégrité)

Sur les deux routeurs :

crypto ipsec transform-set myset esp-aes esp-sha-hmac

- esp-aes : Chiffrement des données avec AES.
- esp-sha-hmac : Hachage SHA pour garantir l'intégrité.

CE1A:

crypto map mymap 10 ipsec-isakmp

set peer 21.21.21.21

(Définition du pair (routeur distant) avec qui établir le tunnel IPsec.)

set transform-set myset

(Application du transform-set `myset` pour définir les paramètres de chiffrement.)

match address 100

(Application de l'ACL 100 pour filtrer le trafic qui doit passer dans le tunnel.)

Sur CE2A le peer sera 1.1.1.1 (Loopback 1 de CE1A)

ACl :

access-list 100 permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255

access-list 100 permit ip 192.168.10.0 0.0.0.255 192.168.40.0 0.0.0.255

access-list 100 permit ip 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255

access-list 100 permit ip 192.168.20.0 0.0.0.255 192.168.40.0 0.0.0.255

(inversement des adresse pour l'autre routeur)

Cette ACL 100 permet uniquement le trafic entre les réseaux locaux à transiter dans le tunnel IPsec. Tout autre trafic ne sera pas chiffré.

Note : une interface Loopback ne peut pas être utilisée directement pour IPSEC, il faut utiliser ces commandes pour utiliser la Loopback en passant par la FastEthernet :

crypto map mymap local-address Loopback1

interface FastEthernet0/0

crypto map mymap

Cette configuration met en place un tunnel IPsec site-to-site entre CE2A et CE1A, garantissant la confidentialité, l'intégrité et l'authentification du trafic entre leurs réseaux locaux.

Test :

On ping de PC1 vers PC6 pour générer du trafic et initialiser notre tunnel. On vérifie ensuite l'établissement de celui ci :

```
CE1A#sh crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
21.21.21.21     1.1.1.1         QM_IDLE           1001    0 ACTIVE

IPv6 Crypto ISAKMP SA
```

```
CE2A#sh crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
1.1.1.1         21.21.21.21     QM_IDLE           1001    0 ACTIVE
```

On voit bien que la phase 1 est active,

Phase 2 :

```
CE1A#sh crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: mymap, local addr 1.1.1.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.10.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.30.0/255.255.255.0/0/0)
   current_peer 21.21.21.21 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 1.1.1.1, remote crypto endpt.: 21.21.21.21
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
     current outbound spi: 0xA750E634(2807096884)

     inbound esp sas:
      spi: 0x2C7F366C(746534508)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 1, flow_id: SW:1, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4443275/2245)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:
```

```
CE2A#sh crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: mymap, local addr 21.21.21.21

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.30.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.10.0/255.255.255.0/0/0)
   current_peer 1.1.1.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 21.21.21.21, remote crypto endpt.: 1.1.1.1
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
     current outbound spi: 0x2C7F366C(746534508)

     inbound esp sas:
      spi: 0xA750E634(2807096884)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 1, flow_id: SW:1, crypto map: mymap
        sa timing: remaining key lifetime (k/sec): (4596172/2261)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
```

On voit bien la local adresse de chacun(Loopback1), le local and remote indent (les Lan de chaque site)

Ainsi que le nombre de paquet encapsulé et décapsuler.

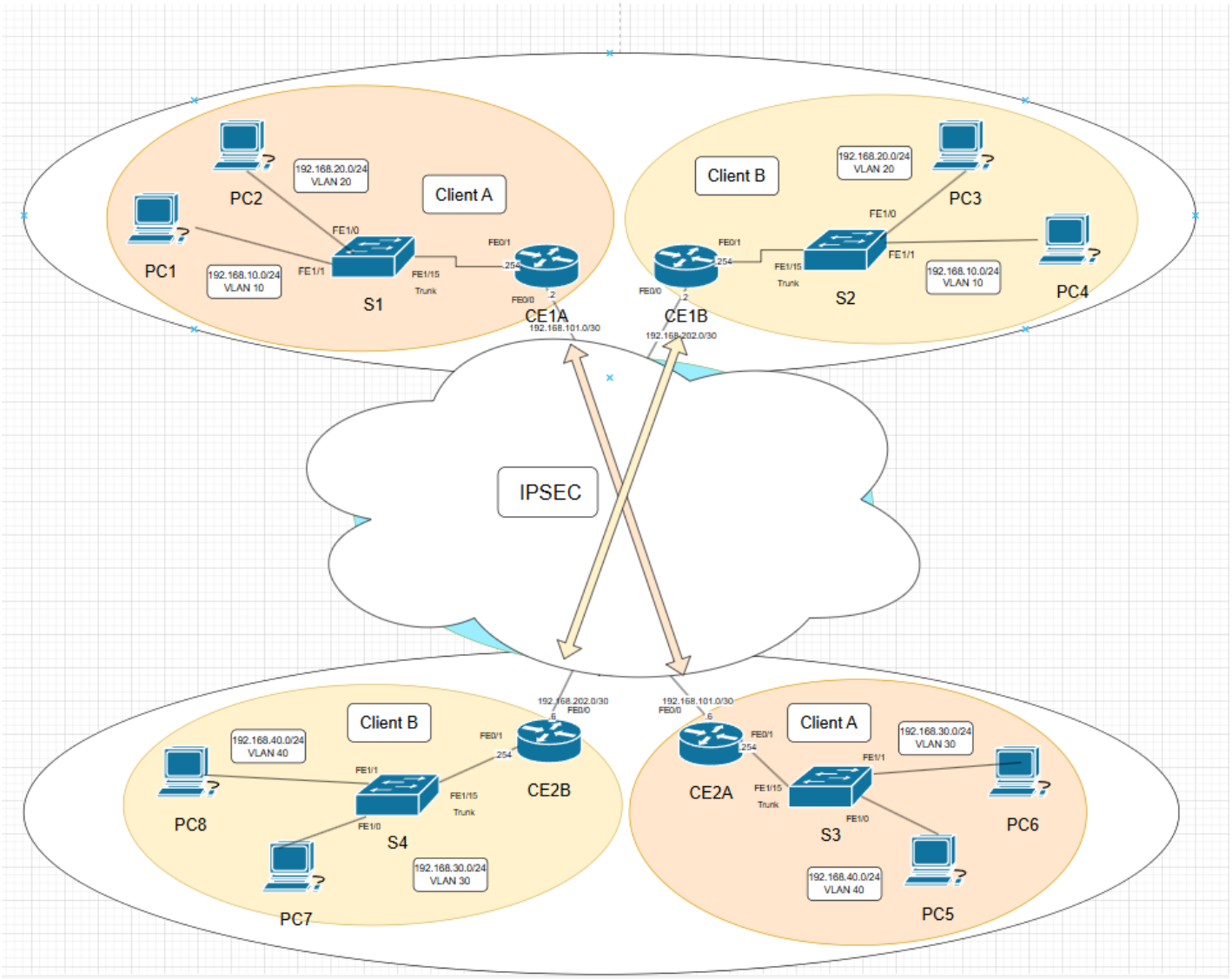
On va maintenant effectuer un trace depuis PC1 vers PC6 :

```
PC3> trace 192.168.30.1
trace to 192.168.30.1, 8 hops max, press Ctrl+C to stop
 1   192.168.10.254   14.555 ms  15.699 ms  15.135 ms
 2   192.168.101.6   135.824 ms  136.553 ms  135.832 ms
 3     *  *  *
 4   *192.168.30.1   152.403 ms (ICMP type:3, code:3, Destination port unreachab
le)
```

1. Passerelle de CE1A
2. Interface de CE2A
3. PC6

Le paquet effectue maintenant ce chemin :

Il nous suffit maintenant de faire la même chose entre CE1B et CE2B et nous avons la configuration final de notre réseau :


PC2
192.168.20.0/24
VLAN 20
Client A
PC1
192.168.10.0/24
VLAN 10
FE1/0
FE1/1
S1
FE1/15
Trunk
FE0/1
.254
FE0/0
.2
CE1A
192.168.101.0/30
Client B
192.168.20.0/24
VLAN 20
PC3
FE1/0
FE0/1
.254
FE1/15
Trunk
FE1/1
S2
FE0/0
.2
CE1B
192.168.202.0/30
192.168.10.0/24
VLAN 10
PC4
IPSEC
192.168.202.0/30
FE0/0
.6
Client B
192.168.40.0/24
VLAN 40
FE0/1
.254
FE1/1
FE1/15
Trunk
FE1/0
S4
CE2B
PC8
PC7
192.168.30.0/24
VLAN 30
192.168.101.0/30
FE0/0
.6
Client A
192.168.30.0/24
VLAN 30
FE0/1
.254
FE1/1
FE1/15
Trunk
FE1/0
S3
CE2A
PC6
192.168.40.0/24
VLAN 40
PC5

5. QOS

La **Q**oS (Quality of Service) est un ensemble de mécanismes permettant de garantir la qualité du service réseau en priorisant certains types de trafic, en limitant la bande passante pour d'autres et en réduisant la latence et la gigue. Elle est particulièrement utile pour les applications sensibles comme la VoIP, le streaming vidéo et les jeux en ligne.

La QoS repose sur plusieurs concepts clés :

- Classification et marquage du trafic : Identifier et étiqueter les paquets en fonction de leur type (ex. VoIP, FTP, HTTP).
- Priorisation du trafic : Assigner un niveau de priorité en fonction des besoins des applications.
- Gestion de la congestion : Utilisation de files d'attente et d'algorithmes pour éviter la saturation du réseau.
- Mécanismes de limitation et de réservation de bande passante : Allocation de bande passante spécifique pour certains flux critiques.
- Réduction de la latence et de la gigue : Essentiel pour les flux en temps réel comme la VoIP.

A. Modèles de QoS :

- Best Effort : Aucun mécanisme de QoS, tout le trafic est traité de la même manière.
- Integrated Services (IntServ) : Réservation de ressources par flux (ex. RSVP). Peu utilisé car lourd à mettre en place.
- Differentiated Services (DiffServ) : Classement du trafic avec des niveaux de priorité (DSCP, CoS).
- MPLS QoS : Utilisation de labels pour garantir la qualité du service sur des réseaux opérateurs.

Mécanismes de QoS :

1. Classification et marquage
    - Utilisation de DSCP (Differentiated Services Code Point) pour identifier les paquets.
    - Marquage des trames sur un switch avec CoS (Class of Service) dans un VLAN (802.1p).
2. Gestion de files d'attente (Queuing)
    - FIFO (First In First Out) : Traitement des paquets dans l'ordre d'arrivée.
    - PQ (Priority Queuing) : Files d'attente avec niveaux de priorité.
    - WFQ (Weighted Fair Queuing) : Répartition de la bande passante selon des poids définis.
    - CBWFQ (Class-Based Weighted Fair Queuing) : Extension de WFQ avec classes de trafic.

Implémentation de la QoS

Configuration QoS sur un routeur Custome Edge

Pour prioriser le trafic VoIP :

Étape 1 : Définir une classe de trafic

```
access-list 102 permit udp any any range 16384 32767  // RTP traffic
access-list 102 permit tcp any any eq 5060          // SIP traffic

class-map match-any VOIP
 match access-group 102
```

Étape 2 : Définir une politique QoS

```
policy-map QOS-POLICY
 class VOIP
  priority percent 20       // Reserve 20% de bande passant a la VOIP
 class class-default
  bandwidth percent 80      // Alloue le reste au reste du traffic
```

Étape 3 : Appliquer la politique sur une interface

```
interface FastEthernet0/0
 service-policy output QoS_POLICY
```

De cette manière les clients pourront installer des logiciels permettant d'émuler des téléphones IP. Le trafic voix bénéficiera donc d'une politique de QOS que l'on pourra facilement adapter

Configuration QoS sur un switch du LAN :

Le trafic vidéo est marqué DSCP AF41 (Assured Forwarding, Class 4, Low Drop Probability).
Le trafic de données (par exemple, transferts de fichiers, navigation Web) est considéré comme une priorité inférieure.
L'objectif est de garantir que le trafic vidéo bénéficie d'une priorité plus élevée et d'une bande passante suffisante, tandis que le trafic de données est traité au mieux.

DSCP (Differentiated Services Code Point):
Un champ de 6 bits dans l'en-tête IP utilisé pour classer et prioriser le trafic au niveau de la couche 3.
La valeur DSCP AF41 correspond à 34 en décimal.

CoS (Class of Service):
Un champ de 3 bits dans l'en-tête de trame Ethernet (balise 802.1Q) utilisé pour la priorisation de couche 2.
Les valeurs CoS vont de 0 (priorité la plus basse) à 7 (priorité la plus élevée).

File d'attente :
Le trafic est placé dans des files d'attente en fonction de la priorité et la bande passante est allouée en conséquence.

Etape 1 : Activer la QOS globalement
mls qos

Etape 2 : Map DSCP to CoS
mls qos map dscp-to-cos 34 to 4

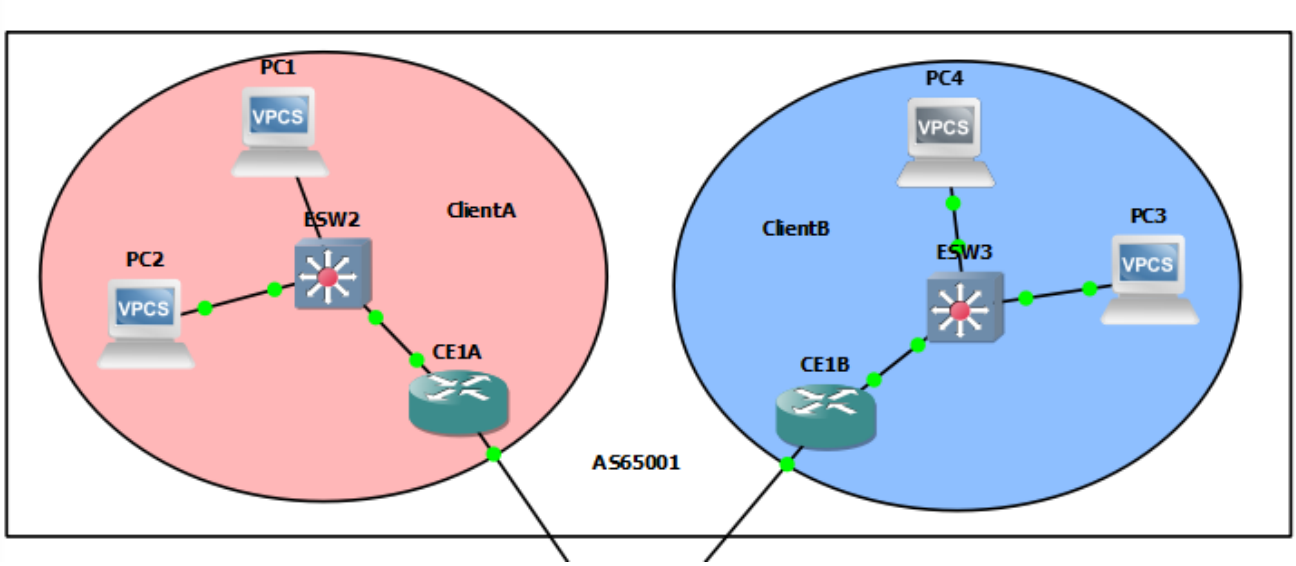
Le trafic vidéo est marqué DSCP AF41, nous devons donc maper cette valeur a une valeur de CoS de couche 2. (CoS 4 correspond a une priorité élevée)

Etape 3 : Configurer le switch pour "faire confiance" au marquage DSCP et assigner une priorité dans la queu :

```
Interface FastEthernet0/1.10
mls qos trust dscp
priority-queue out
Interface FastEthernet0/1.20
mls qos trust dscp
priority-queue out
```

3 - Tests sur GNS3

A. Test de connectivité entre Client A et Client B



Les adresses IP attribuées dynamiquement par les routeurs **CE1A** et **CE1B** aux **PC1,PC2** et **PC3,PC4** sont identiques et appartiennent à la même plage d'adresses.

Cependant, cela ne pose aucun conflit d'adresses, car les sites Clients A et Clients B sont isolés et ne peuvent pas communiquer entre eux.

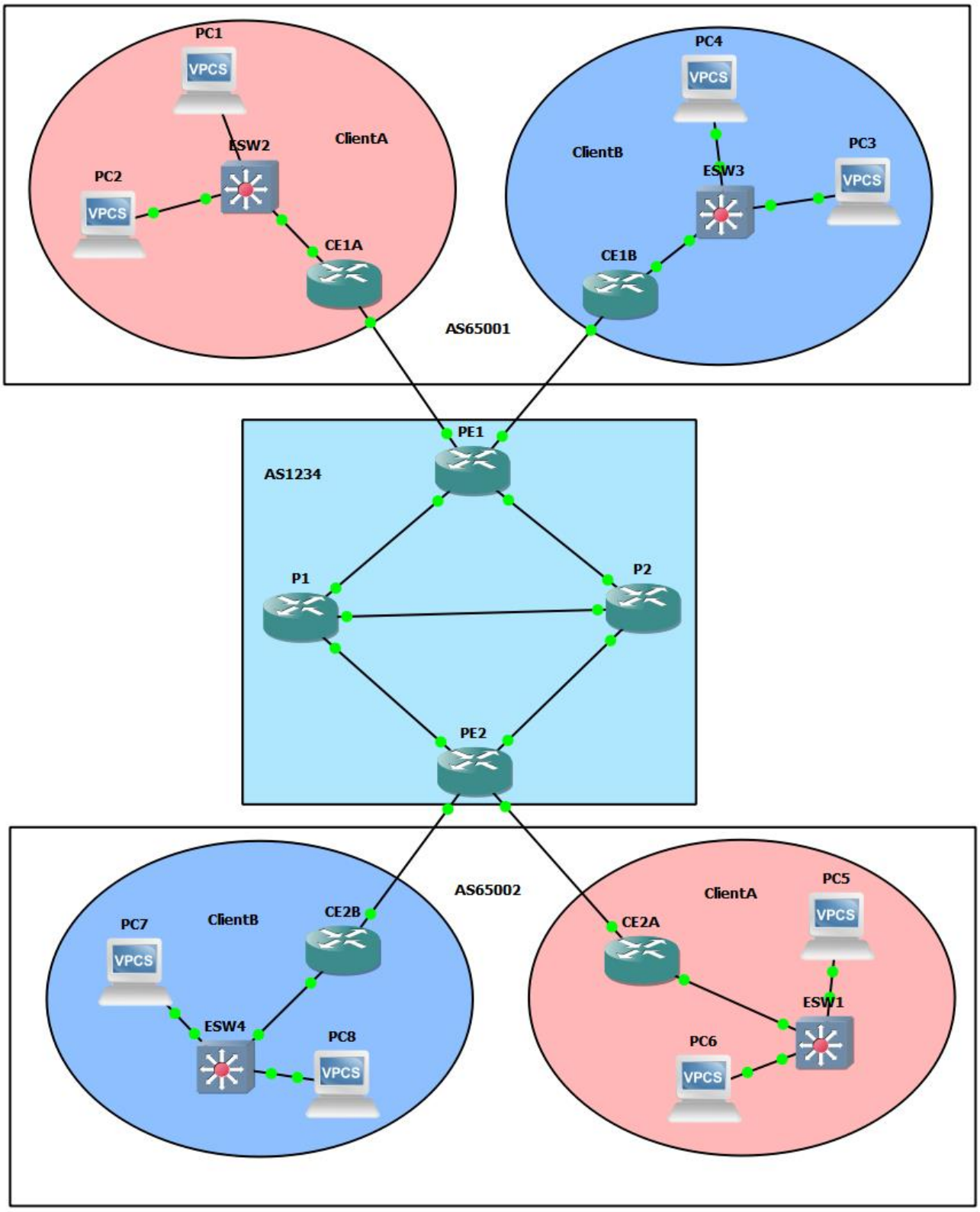
En d'autres termes, chaque site fonctionne comme un réseau indépendant, avec son propre plan d'adressage. Même si des adresses IP identiques sont utilisées des deux côtés, cela n'entraîne aucune interférence, car les paquets ne traversent pas entre les deux clients.

On voit donc par exemple que PC1 et PC4 ont la même IP et cela ne pose aucun problèmes

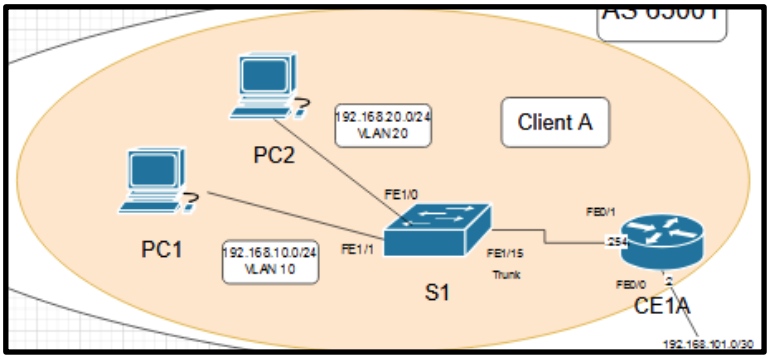
```
PC1> dhcp
DDORA IP 192.168.10.1/24 GW 192.168.10.254

PC1>
```

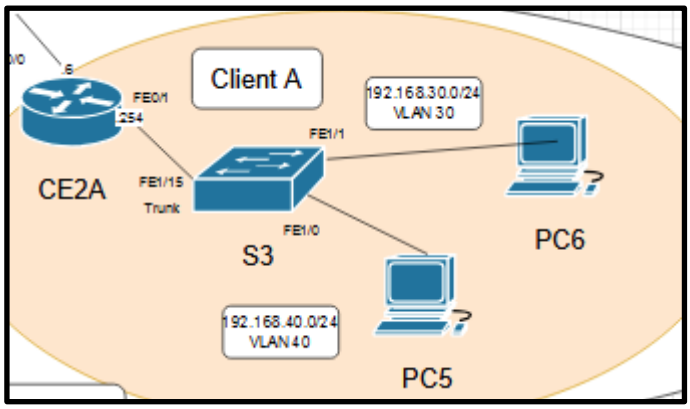
```
PC4> dhcp
DDORA IP 192.168.10.1/24 GW 192.168.10.254

PC4>
```

B. Test de ping entre deux sites du même client



a - Client A :





Le **Client A** possède **deux sites distincts** :

- **Site 1** avec les VLANs **10 et 20**
- **Site 2** avec les VLANs **30 et 40**

Ces **deux sites sont interconnectés** et peuvent **communiquer entre eux** via une liaison fournie par le backbone opérateur.

Chaque site est segmenté en plusieurs **VLANs**, ce qui permet de séparer logiquement les réseaux locaux :

- Sur **le Site 1**, les VLANs **10 et 20** peuvent échanger des données si un **routage inter-VLAN** est en place via le routeur CE1A.
- Sur **le Site 2**, les VLANs **30 et 40** peuvent échanger des données si un **routage inter-VLAN** est en place via le routeur CE2A.
- **Entre les deux sites**, un tunnel IPSEC relie les deux sites

Test ping entre PC1 et PC2 :

```
PC1> ping 192.168.20.1
192.168.20.1 icmp_seq=1 timeout
192.168.20.1 icmp_seq=2 timeout
84 bytes from 192.168.20.1 icmp_seq=3 ttl=63 time=31.450 ms
84 bytes from 192.168.20.1 icmp_seq=4 ttl=63 time=30.427 ms
84 bytes from 192.168.20.1 icmp_seq=5 ttl=63 time=29.273 ms
```

Test ping entre PC1 et PC5 :

```
PC1> ping 192.168.40.1
192.168.40.1 icmp_seq=1 timeout
192.168.40.1 icmp_seq=2 timeout
84 bytes from 192.168.40.1 icmp_seq=3 ttl=62 time=153.629 ms
84 bytes from 192.168.40.1 icmp_seq=4 ttl=62 time=151.345 ms
84 bytes from 192.168.40.1 icmp_seq=5 ttl=62 time=155.643 ms
```

Traceroute entre PC1 et PC5 :

```
PC1> trace 192.168.40.1
trace to 192.168.40.1, 8 hops max, press Ctrl+C to stop
 1   192.168.10.254   15.531 ms  16.167 ms  15.886 ms
 2   192.168.101.6   138.467 ms  136.258 ms  135.306 ms
 3   *192.168.40.1   152.955 ms (ICMP type:3, code:3, Des
le)
```

Nous pouvons observer le cheminement du paquet d'un site à l'autre du client A. Ce dernier est bien encapsulé dans un tunnel IPSEC car l'on passe directement de l'interface d'un CE à l'interface de l'autre CE. On ne voit pas les hop dans le backbone.

IPV6 :

```
PC2> ip auto
GLOBAL SCOPE      : 2001:db8:20:0:2050:79ff:fe66:6803/64
ROUTER LINK-LAYER : c2:01:51:6c:00:01
```

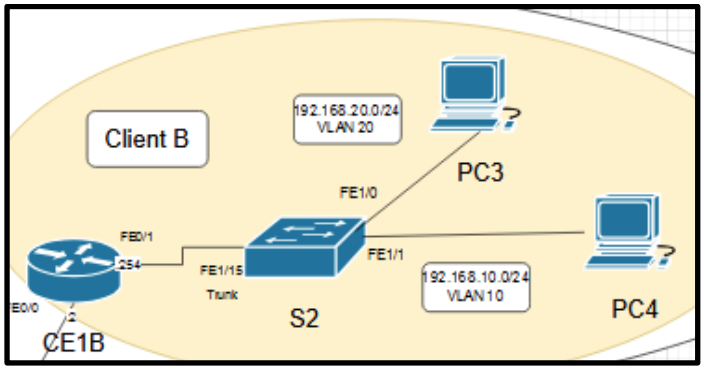
```
PC1> ip auto
GLOBAL SCOPE      : 2001:db8:10:0:2050:79ff:fe66:6802/64
ROUTER LINK-LAYER : c2:01:51:6c:00:01
```

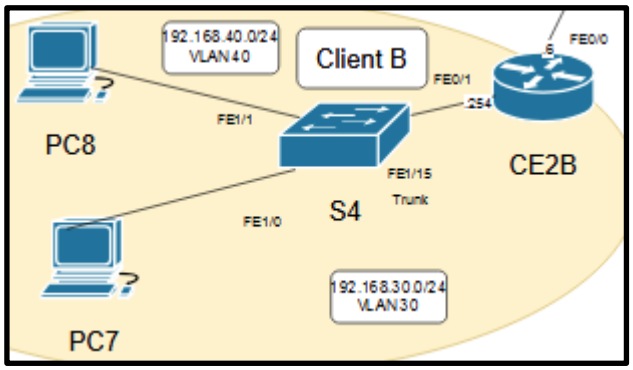
PC1 vers PC2 IPV6

```
PC1> ping 2001:db8:20:0:2050:79ff:fe66:6803/64

2001:db8:20:0:2050:79ff:fe66:6803 icmp6_seq=1 ttl=62 time=62.387 ms
2001:db8:20:0:2050:79ff:fe66:6803 icmp6_seq=2 ttl=62 time=29.084 ms
2001:db8:20:0:2050:79ff:fe66:6803 icmp6_seq=3 ttl=62 time=27.976 ms
2001:db8:20:0:2050:79ff:fe66:6803 icmp6_seq=4 ttl=62 time=29.727 ms
2001:db8:20:0:2050:79ff:fe66:6803 icmp6_seq=5 ttl=62 time=31.297 ms
```

Note : pour effectuer un ping entre deux site distant via IPV6 il faudrait configurer un tunel GRE sur IPSEC pour gérer l'encapsulation IPV6 dans V4

b - Client B :





Le **Client B** possède **deux sites distincts** :

- **Site 1** avec les VLANs **10 et 20**
- **Site 2** avec les VLANs **30 et 40**

Ces **deux sites sont interconnectés** et peuvent **communiquer entre eux** comme pour le client A

```
PC4> dhcp
DDORA IP 192.168.10.1/24 GW 192.168.10.254
```

```
PC3> dhcp
DDORA IP 192.168.20.1/24 GW 192.168.20.254
```

```
PC8> dhcp
DDORA IP 192.168.40.1/24 GW 192.168.40.254

PC8>
```

Ping PC 4 vers PC3 (InterVLAN) :

```
PC4> ping 192.168.20.1
192.168.20.1 icmp_seq=1 timeout
192.168.20.1 icmp_seq=2 timeout
84 bytes from 192.168.20.1 icmp_seq=3 ttl=63 time=30.226 ms
84 bytes from 192.168.20.1 icmp_seq=4 ttl=63 time=31.052 ms
84 bytes from 192.168.20.1 icmp_seq=5 ttl=63 time=29.457 ms
```

Ping PC 4 vers PC8 (Entre site) :

```
PC4> ping 192.168.40.1
192.168.40.1 icmp_seq=1 timeout
192.168.40.1 icmp_seq=2 timeout
84 bytes from 192.168.40.1 icmp_seq=3 ttl=59 time=150.486 ms
84 bytes from 192.168.40.1 icmp_seq=4 ttl=59 time=155.334 ms
84 bytes from 192.168.40.1 icmp_seq=5 ttl=59 time=153.282 ms
```

Trace PC 4 vers PC8 :

```
PC4> trace 192.168.40.1
trace to 192.168.40.1, 8 hops max, press Ctrl+C to stop
 1   192.168.10.254   16.291 ms  14.443 ms  15.661 ms
 2   192.168.202.6   134.712 ms  138.546 ms  136.441 ms
 3   *192.168.40.1   152.698 ms (ICMP type:3, code:3, De
le)
```

Nous pouvons observer le cheminement du paquet d'un site à l'autre du client B. Ce dernier est bien encapsulé dans un tunnel IPSEC car l'on passe directement de l'interface d'un CE à l'interface de l'autre CE. On ne voit pas les hop dans le backbone.

IPV6

```
PC4> ip auto
GLOBAL SCOPE      : 2001:db8:10:0:2050:79ff:fe66:6804/64
ROUTER LINK-LAYER : c2:02:1d:f4:00:01
```

```
PC3> ip auto
GLOBAL SCOPE      : 2001:db8:20:0:2050:79ff:fe66:6805/64
ROUTER LINK-LAYER : c2:02:1d:f4:00:01
```

On a bien une adresse avec SLACC avec le préfixe du VLAN

```
PC4> ping 2001:db8:20:0:2050:79ff:fe66:6805/64

2001:db8:20:0:2050:79ff:fe66:6805 icmp6_seq=1 ttl=62 time=60.503 ms
2001:db8:20:0:2050:79ff:fe66:6805 icmp6_seq=2 ttl=62 time=30.410 ms
2001:db8:20:0:2050:79ff:fe66:6805 icmp6_seq=3 ttl=62 time=30.716 ms
2001:db8:20:0:2050:79ff:fe66:6805 icmp6_seq=4 ttl=62 time=30.357 ms
2001:db8:20:0:2050:79ff:fe66:6805 icmp6_seq=5 ttl=62 time=30.301 ms
```

Note : pour effectuer un ping entre deux site distant via IPV6 il faudrait configurer un tunnel GRE sur IPSEC pour gérer l'encapsulation IPV6 dans V4

4 - Conclusion

Nous avons mis en place une infrastructure multi-sites sécurisée comprenant deux clients composé de deux sites chacun.

Cette infrastructure repose sur les concepts fondamentaux du réseau : routage dynamique : BGP, MPLS, OSPF, Sécurisation et séparation avec VPN et VRF. Ainsi que les réseau LAN avec : VLAN, ACl, routage Inter VLAN, DHCP et QOS.

Nous avons détaillés la mise en place et la configuration des équipements et avons prouvé le bon fonctionnement de ceux-ci.

Pour améliorer ce réseau nous pourrions mettre en place la communication IPV6 entre les sites avec un tunnel GRE sur IPSEC, la communication IPV6 n'étant pour le moment disponible qu'à l'intérieur des sites.