Université de Strasbourg

UFR Mathématique-Informatique

Réseaux d'opérateur - TP3 - VPN BGP-MPLS

Guillaume LUCAS

Strasbourg, le 11 août 2013

Sommaire

1 Avant de commencer

1.1 Adressage

1.1.1 ASN

En ce qui concerne les numéros d'AS que l'on doit utiliser avec BGP, j'ai décidé d'en choisir un parmi la plage d'ASN réservée à l'IANA pour la documentation (64496-64511,65536-65551) : 64501.

1.1.2 IPv4

Concernant l'adressage IP, j'ai décidé de choisir mon préfixe dans la plage réservée à l'IANA pour les tests : 198.18.0.0/15. Pour une plus grande facilité, j'ai décidé d'utiliser le sous-préfixe 198.18.0.0/16.

Pour faciliter l'adressage, j'ai décidé de découper ce préfixe en plusieurs préfixes de longueur /20 :

1. 198.18.0.0/20 : services. Les loopback de mes routeurs seront dans ce sous-réseau.
2. 198.18.16.0/20 : VPN A
3. 198.18.32.0/20 : VPN B
4. Le milieu est réservé pour un usage futur (expansion des "services", expansion des interconnexions, proposer de nouveaux VPN, ...).
5. Le dernier /20, 198.18.240.0/20, est dédié aux interconnexions des routeurs internes. Il sera découpé en /31, chaque /31 étant dédié à une interconnexion.

Je ne prévois pas d'avoir plus de 2048 interconnexions internes mais si c'était le cas, je prendrais des préfixes dans le milieu en suivant mais en partant de la fin.

Les interconnexions se font en /31 car les IPv4 sont rares et il faut donc les économiser. Même si nous utilisons un préfixe de documentation/test, nous avons fait ce choix car, d'après nos renseignements, il s'agit d'une BCP donc pourquoi ne pas s'y conformer, même sur une maquette ? De plus, cela amène des avantages (pas de broadcast, ...).

Sur chaque routeur de l'opérateur, j'ai une loopback qui permet, entre autres, de raccrocher des démons (BGP) et de tester MPLS "de bout en bout". Les préfixes de mes loopback de services sont :

- P - L1 : 198.18.0.1/32
- PE1 - L1 : 198.18.0.2/32
- PE2 - L1 : 198.18.0.3/32
- PE3 - L1 : 198.18.0.4/32
- PE4 - L1 : 198.18.0.5/32

Malgrè que l'on puisse utiliser le même adressage pour plusieurs VPN (le RD permet de faire la différence), j'ai décidé d'attribuer un préfixe par VPN pour faciliter un éventuel débugage.

Chaque préfixe dédié à un VPN (/20) sera découpé en /22, un pour chaque site. Le dernier /22 sera dédié aux interconnexions entre PE et CE, si besoin.

En pratique, pour le VPN A (198.18.16.0/20), on obtient :

- 198.18.16.0/22 : site 1 (CE1)
- 198.18.20.0/22 : site 2 (CE2)
- 198.18.24.0/22 : site 3 (PE1)
- 198.18.28.0/22 : interconnexions PE-CE. Détails :
  - 198.18.28.0/31 : PE2 <-> CE1
  - 198.18.28.2/31 : PE3 <-> CE2

Pour le VPN B (198.18.32.0/20), on obtient :

- 198.18.32.0/22 : site 1 (PE4)
- 198.18.36.0/22 : site 2 (PE3)
- 198.18.40.0/22 : site 3 (CE3)
- 198.18.44.0/22 : interconnexions PE-CE. Détails :
  - 198.18.44.0/31 : PE2 <-> CE3

Note : pour le VPN B, j'ai décidé d'avoir un CE pour que l'utilisation d'OSPF demandée (et des redistributions associées) fasse sens.

Sur chaque CE, j'ai également une loopback qui permet de faire comme si j'avais le réseau de mon client directement connecté :

- CE1 : 198.18.19.254/22
- CE2 : 198.18.23.254/22
- CE3 : 198.18.43.254/22

Note : les sorties des commandes de vérification (show ...) seront allégées/tronquées pour ne garder que l'essentiel afin de ne pas surcharger ce rapport.

1.2 Schéma

En figure 1 se trouve le schéma donné dans l'énoncé complété avec mon plan d'adressage et les VPN.

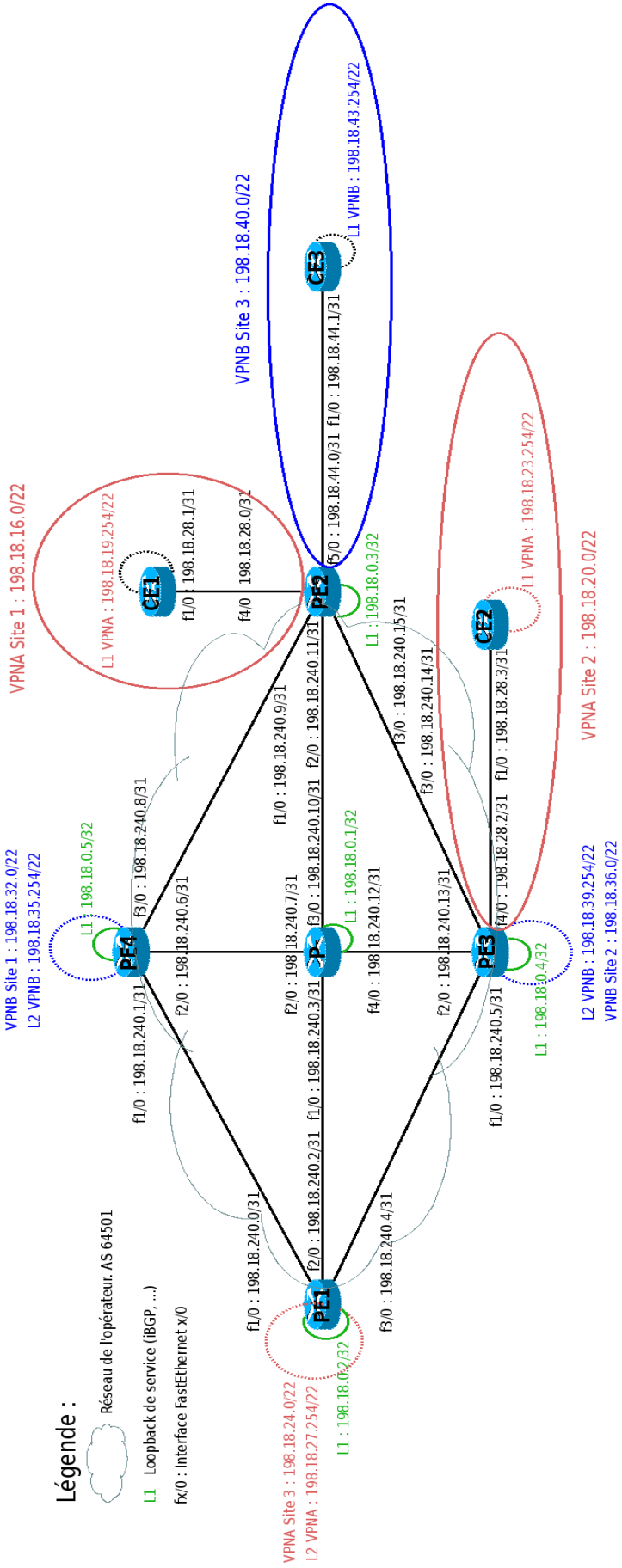


FIGURE 1 – Schéma de ma maquette. D'après l'énoncé.

2 VPN sans TE

2.1 Question 1

La configuration de dynagen pour ce TP est donnée en annexe 1.

La configuration des routeurs se fait assez facilement : il n'y a pas de commandes qui n'ont pas été données lors de la séance de TP. Les configurations intégrales de mes routeurs se trouvent en annexe 2.

D'abord, regardons la table de routage IP de P pour constater que le réseau de l'opérateur est opérationnel :

```
P#sh ip route
     198.18.240.0/31 is subnetted, 8 subnets
O       198.18.240.4 [110/2] via 198.18.240.13, 04:52:09, FastEthernet4/0
                     [110/2] via 198.18.240.2, 04:52:09, FastEthernet1/0
C       198.18.240.6 is directly connected, FastEthernet2/0
O       198.18.240.0 [110/2] via 198.18.240.6, 04:52:09, FastEthernet2/0
                     [110/2] via 198.18.240.2, 04:52:09, FastEthernet1/0
C       198.18.240.2 is directly connected, FastEthernet1/0
C       198.18.240.12 is directly connected, FastEthernet4/0
O       198.18.240.14 [110/2] via 198.18.240.13, 04:52:09, FastEthernet4/0
                      [110/2] via 198.18.240.11, 04:52:09, FastEthernet3/0
O       198.18.240.8 [110/2] via 198.18.240.11, 04:52:09, FastEthernet3/0
                     [110/2] via 198.18.240.6, 04:52:09, FastEthernet2/0
C       198.18.240.10 is directly connected, FastEthernet3/0
     198.18.0.0/32 is subnetted, 5 subnets
O       198.18.0.4 [110/2] via 198.18.240.13, 04:52:12, FastEthernet4/0
O       198.18.0.5 [110/2] via 198.18.240.6, 04:52:12, FastEthernet2/0
C       198.18.0.1 is directly connected, Loopback1
O       198.18.0.2 [110/2] via 198.18.240.2, 04:52:12, FastEthernet1/0
O       198.18.0.3 [110/2] via 198.18.240.11, 04:52:12, FastEthernet3/0
```

Ensuite, regardons les VRF de chaque PE pour vérifier que les VPN sont bien opérationnels :

```
PE1#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
B       198.18.28.0 [200/0] via 198.18.0.3, 04:09:26
```

```
B       198.18.28.2 [200/0] via 198.18.0.4, 04:09:26
C    198.18.24.0/22 is directly connected, Loopback2
B    198.18.16.0/22 [200/1] via 198.18.0.3, 04:09:26
B    198.18.20.0/22 [200/1] via 198.18.0.4, 04:09:26

PE2#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
C       198.18.28.0 is directly connected, FastEthernet4/0
B       198.18.28.2 [200/0] via 198.18.0.4, 04:12:53
B    198.18.24.0/22 [200/0] via 198.18.0.2, 04:12:53
R    198.18.16.0/22 [120/1] via 198.18.28.1, 00:00:07, FastEthernet4/0
B    198.18.20.0/22 [200/1] via 198.18.0.4, 04:12:53

PE2#sh ip route vrf vpnb
     198.18.43.0/32 is subnetted, 1 subnets
O       198.18.43.254 [110/2] via 198.18.44.1, 04:13:59, FastEthernet5/0
     198.18.44.0/31 is subnetted, 1 subnets
C       198.18.44.0 is directly connected, FastEthernet5/0
B    198.18.32.0/22 [200/0] via 198.18.0.5, 04:12:58
B    198.18.36.0/22 [200/0] via 198.18.0.4, 04:12:58

PE3#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
B       198.18.28.0 [200/0] via 198.18.0.3, 04:13:58
C       198.18.28.2 is directly connected, FastEthernet4/0
B    198.18.24.0/22 [200/0] via 198.18.0.2, 04:13:58
B    198.18.16.0/22 [200/1] via 198.18.0.3, 04:13:58
R    198.18.20.0/22 [120/1] via 198.18.28.3, 00:00:02, FastEthernet4/0

PE3#sh ip route vrf vpnb
     198.18.43.0/32 is subnetted, 1 subnets
B       198.18.43.254 [200/2] via 198.18.0.3, 04:13:59
     198.18.44.0/31 is subnetted, 1 subnets
B       198.18.44.0 [200/0] via 198.18.0.3, 04:13:59
B    198.18.32.0/22 [200/0] via 198.18.0.5, 04:13:59
C    198.18.36.0/22 is directly connected, Loopback2
```

```
PE4#sh ip route vrf vpnb
     198.18.43.0/32 is subnetted, 1 subnets
B       198.18.43.254 [200/2] via 198.18.0.3, 04:14:48
     198.18.44.0/31 is subnetted, 1 subnets
B       198.18.44.0 [200/0] via 198.18.0.3, 04:14:48
C    198.18.32.0/22 is directly connected, Loopback2
B    198.18.36.0/22 [200/0] via 198.18.0.4, 04:14:48
```

Enfin, regardons la table de routage IP de nos CE pour vérifier que les redistributions internes à nos VPN sont opérationnelles :

```
CE1#sh ip route
     198.18.28.0/31 is subnetted, 2 subnets
C       198.18.28.0 is directly connected, FastEthernet1/0
R       198.18.28.2 [120/1] via 198.18.28.0, 00:00:24, FastEthernet1/0
R    198.18.24.0/22 [120/1] via 198.18.28.0, 00:00:24, FastEthernet1/0
C    198.18.16.0/22 is directly connected, Loopback1
R    198.18.20.0/22 [120/1] via 198.18.28.0, 00:00:24, FastEthernet1/0

CE2#sh ip route
     198.18.28.0/31 is subnetted, 2 subnets
R       198.18.28.0 [120/1] via 198.18.28.2, 00:00:18, FastEthernet1/0
C       198.18.28.2 is directly connected, FastEthernet1/0
R    198.18.24.0/22 [120/1] via 198.18.28.2, 00:00:18, FastEthernet1/0
R    198.18.16.0/22 [120/1] via 198.18.28.2, 00:00:18, FastEthernet1/0
C    198.18.20.0/22 is directly connected, Loopback1

CE3#sh ip route
     198.18.44.0/31 is subnetted, 1 subnets
C       198.18.44.0 is directly connected, FastEthernet1/0
C    198.18.40.0/22 is directly connected, Loopback1
O IA 198.18.32.0/22 [110/2] via 198.18.44.0, 04:19:58, FastEthernet1/0
O IA 198.18.36.0/22 [110/2] via 198.18.44.0, 04:19:58, FastEthernet1/0
```

Pour conclure, faisons quelques tests de connectivité à l'intérieur de chacun de nos VPN.

On effectue un traceroute à destination du site 1 du VPN A (CE1) depuis le site 3 du VPN A (L2 de PE1). On effectue ensuite un ping à destination du site 2 du VPN A (CE2) depuis le site 3 du VPN A (L2 de PE1) :

```
PE1#traceroute vrf vpna 198.18.19.254 source 198.18.27.254
  1 198.18.240.5 [MPLS: Labels 23/25 Exp 0] 12 msec
    198.18.240.3 [MPLS: Labels 22/25 Exp 0] 12 msec
    198.18.240.1 [MPLS: Labels 23/25 Exp 0] 12 msec
  2 198.18.28.0 [MPLS: Label 25 Exp 0] 12 msec 12 msec 12 msec
  3 198.18.28.1 20 msec *  24 msec


PE1#ping  vrf vpna 198.18.23.254 source 198.18.27.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/12/16 ms
```

Notre VPN A est totalement fonctionnel.

On effectue un ping à destination du site 1 du VPN B (L2 de PE4) depuis le site 3 du VPN B (CE3). On effectue ensuite un traceroute à destination du site 2 du VPN B (L2 de PE3) depuis le site 3 du VPN B (CE3) :

```
CE3#ping 198.18.35.254 source 198.18.43.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/8/12 ms


CE3#traceroute 198.18.39.254 source 198.18.43.254
  1 198.18.44.0 8 msec 8 msec 32 msec
  2 198.18.39.254 8 msec *  32 msec
```

Notre VPN B est totalement fonctionnel.

2.2 Question 2

Le routeur P effectue la commutation uniquement en se basant sur les labels MPLS de sommet de la pile. Il n'a pas conscience de l'empilement des labels MPLS. Il n'a donc pas connaisance des préfixes IP attribués aux différents VPN dans sa table de routage IP ni dans sa table de commutation MPLS.

Notons néanmoins que P reçoit l'intégralité des paquets qu'il doit commuter. Une capture du trafic réseau mettrait donc en évidence les préfixes IPs des VPN dans les couches supérieures.

Pour la vérification, voir la sortie de la commande show ip route donnée dans la réponse à la question 1.

2.3 Question 3

Pour illustrer mon raisonnement, j'ai décidé de faire un ping/traceroute à destination du site 1 du VPN A (CE1) depuis le site 3 de ce même VPN (L2 de PE1).

Pour commencer, regardons les tables de commutation pour en déduire le comportement :

```
PE1#show mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
17     Pop tag     198.18.0.1/32    0          Fa2/0      198.18.240.3
18     Pop tag     198.18.0.5/32    0          Fa1/0      198.18.240.1
23     23          198.18.0.3/32    0          Fa3/0      198.18.240.5
       22          198.18.0.3/32    0          Fa2/0      198.18.240.3
       23          198.18.0.3/32    0          Fa1/0      198.18.240.1
24     Pop tag     198.18.0.4/32    37495      Fa3/0      198.18.240.5

PE1#show ip bgp vpnv4 all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 64501:1 (vpna)
   198.18.16.0/22   198.18.0.3      nolabel/25
   198.18.20.0/22   198.18.0.4      nolabel/25
   198.18.24.0/22   0.0.0.0         25/aggregate(vpna)

PE2#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
20     Pop tag     198.18.0.1/32    0          Fa2/0      198.18.240.10
21     22          198.18.0.2/32    0          Fa3/0      198.18.240.14
       16          198.18.0.2/32    0          Fa2/0      198.18.240.10
       20          198.18.0.2/32    0          Fa1/0      198.18.240.8
22     Pop tag     198.18.0.5/32    28695      Fa1/0      198.18.240.8
24     Pop tag     198.18.0.4/32    56         Fa3/0      198.18.240.14

PE2#sh ip bgp vpnv4 all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 64501:1 (vpna)
   198.18.16.0/22   198.18.28.1     25/nolabel
```

```
198.18.20.0/22    198.18.0.4      nolabel/25
198.18.24.0/22    198.18.0.2      nolabel/25
```

Vu les tables ci-dessus, on devrait obtenir :

Aller : PE1 [22 ou 23,25] -> PE4, P ou PE3 [25] -> PE2 [plus de MPLS ici] -> CE1

Retour : CE1 [pas de MPLS ici] -> PE2 [22 ou 16 ou 20,25] -> PE4, P ou PE3 [25] -> PE1

Le fonctionnement de MPLS ne change pas. En conséquence, il y aura du Penultimate Hop Popping entre les PE. Seul le label qui identifie un VPN/une VRF/une route restera d'un bout à l'autre (du PE source au PE destination).

Vérification :

```
PE1#traceroute vrf vpna 198.18.19.254 source 198.18.27.254
  1 198.18.240.5 [MPLS: Labels 23/25 Exp 0] 16 msec
    198.18.240.3 [MPLS: Labels 22/25 Exp 0] 12 msec
    198.18.240.1 [MPLS: Labels 23/25 Exp 0] 16 msec
  2 198.18.28.0 [MPLS: Label 25 Exp 0] 12 msec 12 msec 12 msec
  3 198.18.28.1 12 msec *  24 msec
```

Vérification, sur PE1, avec un ping vrf vpna 198.18.19.254 source 198.18.27.254 :

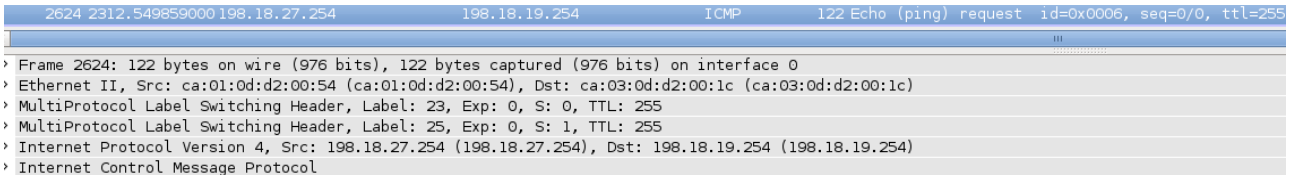


FIGURE 2 – En sortie de PE1, pour l'echo request, nous avons bien deux labels : le premier, 23, pour la commutation dans le réseau de l'opérateur, le deuxième, 25, pour identifier le VPN.

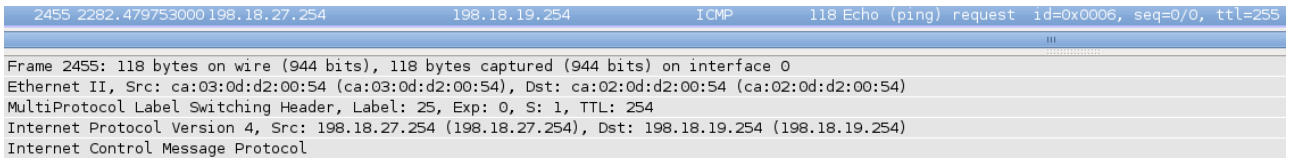


FIGURE 3 – En entrée de PE2, pour l'echo request, nous n'avons plus qu'un seul label : celui de fond de pile, 25, qui permet d'idenfitier le VPN. Le label de sommet de pile a été poper par PE3.

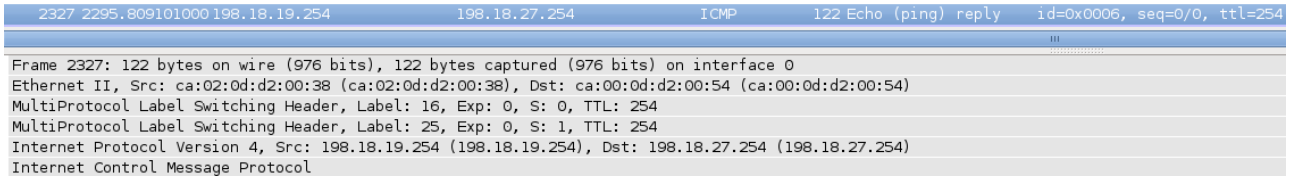


FIGURE 4 – En sortie de PE2, pour l'echo reply, nous avons deux labels : le premier, 16, pour la commutation dans le réseau de l'opérateur, le deuxième, 25, pour identifier le VPN.

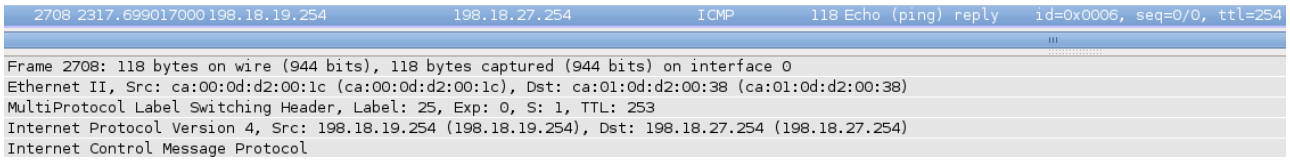


FIGURE 5 – En entrée de PE1, pour l'echo reply, nous n'avons plus que le label de fond de pile, 25, qui permet d'identifier le VPN. Le label de sommet de pile a été poper par P.

2.4 Question 4

2.4.1 Ajout d'un PE

En théorie, lors de l'ajout d'un PE (sans VPN, sans liaisons iBGP, rien), les PE voisins (directement connectés), vont annoncer les nouveaux préfixes d'interconnexion au reste du réseau de l'opérateur. Puis, lorsque son initialisation sera complète, le nouveau PE s'annoncera lui-même, via OSPF, à ses voisins, qui relayeront cette annonce. Puis, de nouveaux labels seront assignés et distribués via LDP.

Si l'on configure aussi les sessions iBGP avec les autres PE sans qu'il n'y ait aucun site sur ce PE, alors on aura aussi des messages iBGP : ouverture des sessions et keepalive.

Vérification : j'introduis PE4 dans ma maquette.

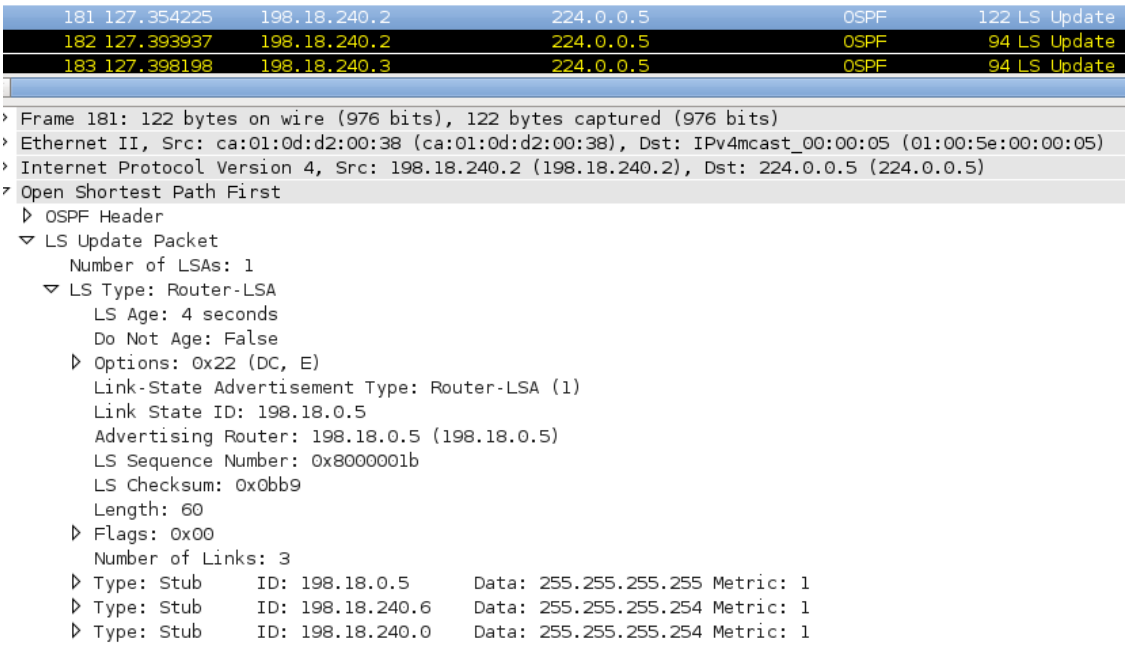


FIGURE 6 – PE1 annonce (à P dans notre cas), le nouveau préfixe d'interconnexion 198.18.240.0 entre lui et PE4.

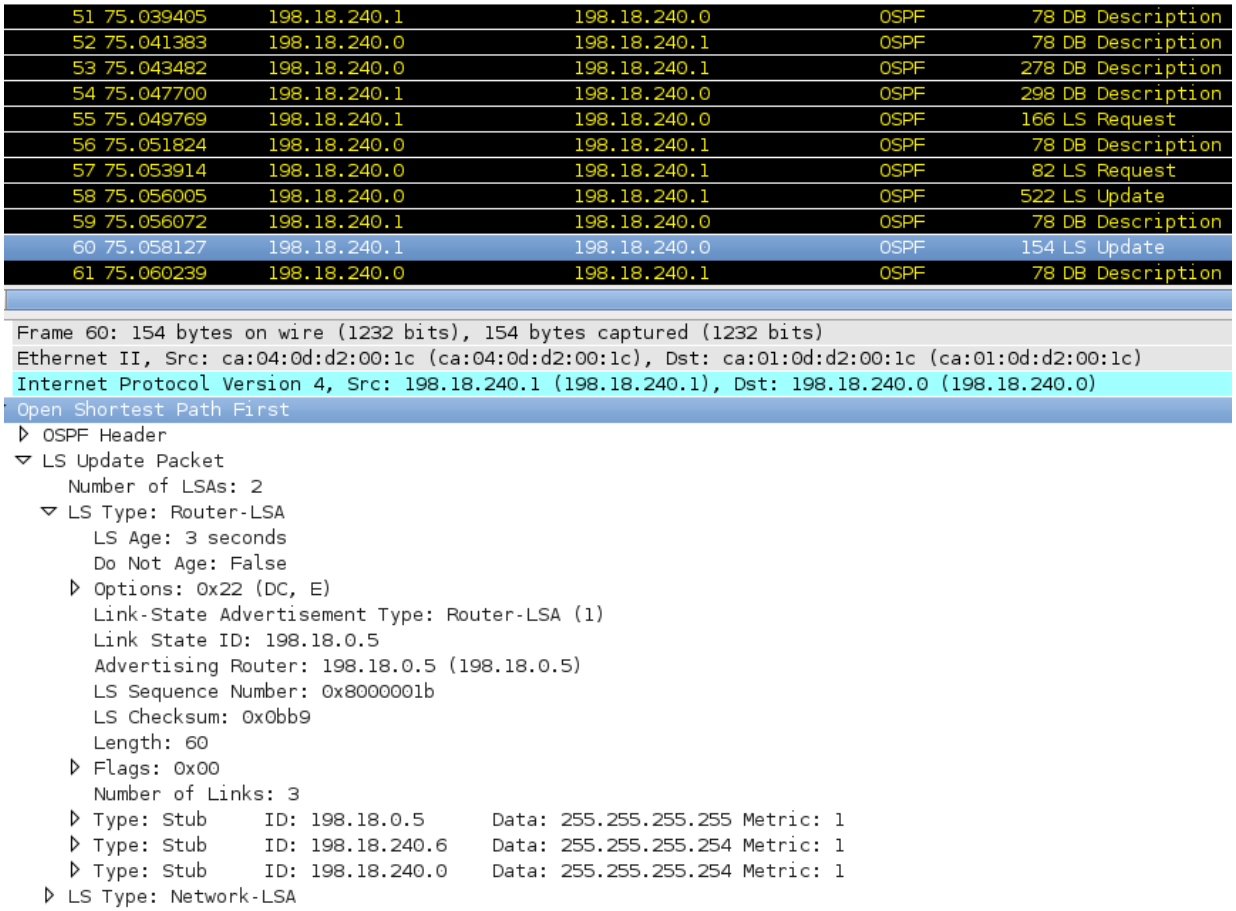


FIGURE 7 – À la fin de son initialisation, PE4 ouvre des sessions OSPF avec ses voisins (PE1 dans notre cas).

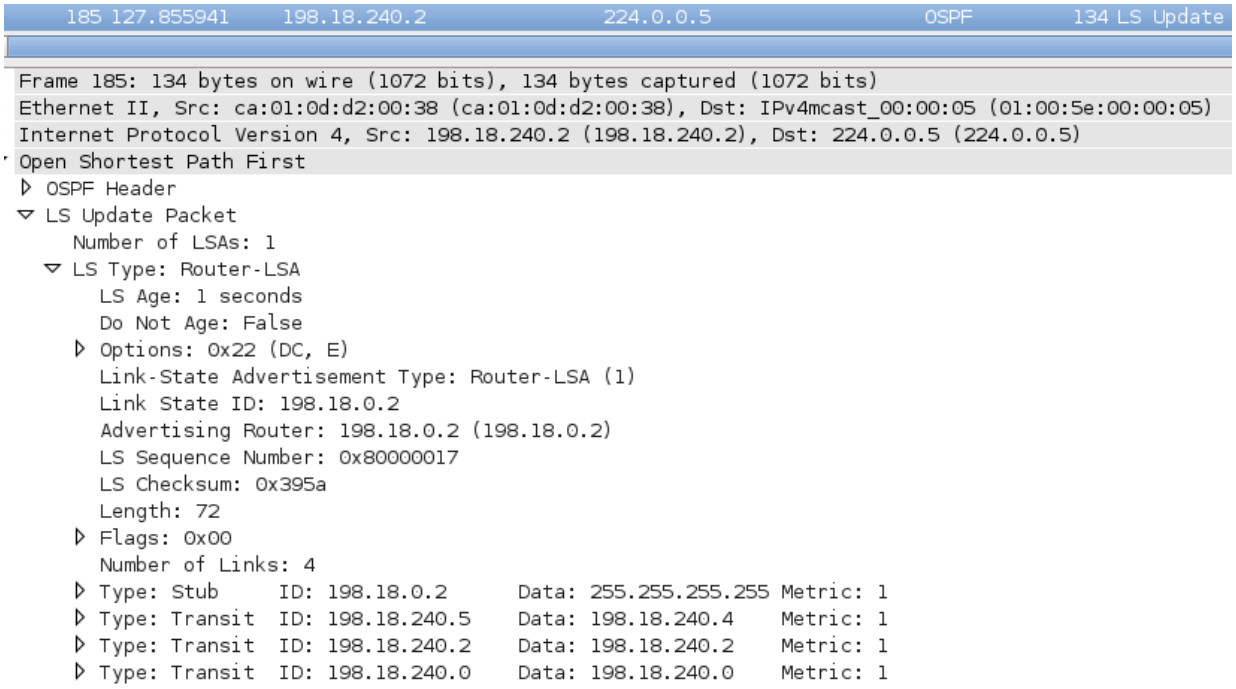
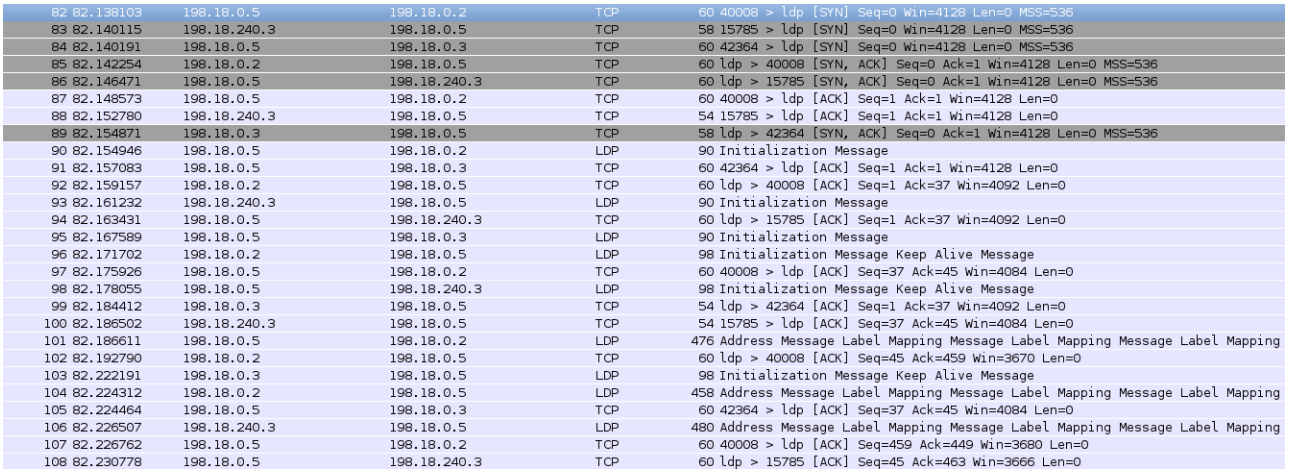
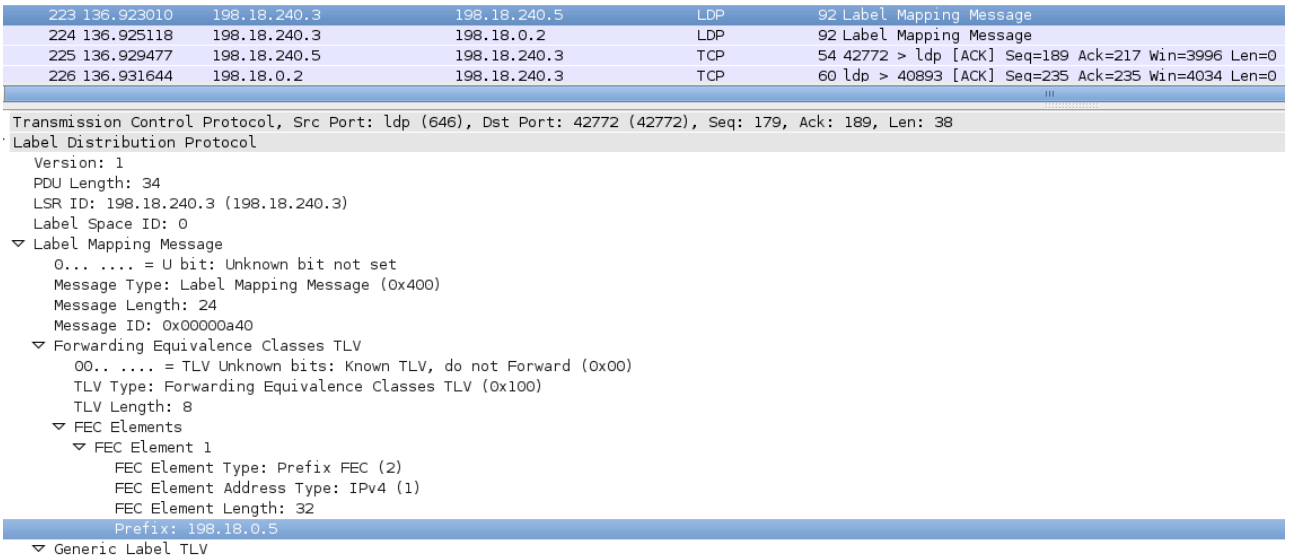


FIGURE 8 – Les voisins de PE4 (PE1 dans notre cas) relayent les annonces de PE4 et marquent le préfixe d'interconnexion PE1<->PE4 comme un lien de transit vers d'autres préfixes.



82 82.138103	198.18.0.5	198.18.0.2	TCP	60 40008 > ldp [SYN] Seq=0 Win=4128 Len=0 MSS=536
83 82.140115	198.18.240.3	198.18.0.5	TCP	58 15785 > ldp [SYN] Seq=0 Win=4128 Len=0 MSS=536
84 82.140191	198.18.0.5	198.18.0.3	TCP	60 42364 > ldp [SYN] Seq=0 Win=4128 Len=0 MSS=536
85 82.142254	198.18.0.2	198.18.0.5	TCP	60 ldp > 40008 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
86 82.146471	198.18.0.5	198.18.240.3	TCP	60 ldp > 15785 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
87 82.148573	198.18.0.5	198.18.0.2	TCP	60 40008 > ldp [ACK] Seq=1 Ack=1 Win=4128 Len=0
88 82.152780	198.18.240.3	198.18.0.5	TCP	54 15785 > ldp [ACK] Seq=1 Ack=1 Win=4128 Len=0
89 82.154871	198.18.0.3	198.18.0.5	TCP	58 ldp > 42364 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
90 82.154946	198.18.0.5	198.18.0.2	LDP	90 Initialization Message
91 82.157083	198.18.0.5	198.18.0.3	TCP	60 42364 > ldp [ACK] Seq=1 Ack=1 Win=4128 Len=0
92 82.159157	198.18.0.2	198.18.0.5	TCP	60 ldp > 40008 [ACK] Seq=1 Ack=37 Win=4092 Len=0
93 82.161232	198.18.240.3	198.18.0.5	LDP	90 Initialization Message
94 82.163431	198.18.0.5	198.18.240.3	TCP	60 ldp > 15785 [ACK] Seq=1 Ack=37 Win=4092 Len=0
95 82.167589	198.18.0.5	198.18.0.3	LDP	90 Initialization Message
96 82.171702	198.18.0.2	198.18.0.5	LDP	98 Initialization Message Keep Alive Message
97 82.175926	198.18.0.5	198.18.0.2	TCP	60 40008 > ldp [ACK] Seq=37 Ack=45 Win=4084 Len=0
98 82.178055	198.18.0.5	198.18.240.3	LDP	98 Initialization Message Keep Alive Message
99 82.184412	198.18.0.3	198.18.0.5	TCP	54 ldp > 42364 [ACK] Seq=1 Ack=37 Win=4092 Len=0
100 82.186502	198.18.240.3	198.18.0.5	TCP	54 15785 > ldp [ACK] Seq=37 Ack=45 Win=4084 Len=0
101 82.186611	198.18.0.5	198.18.0.2	LDP	476 Address Message Label Mapping Message Label Mapping Message Label Mapping
102 82.192790	198.18.0.2	198.18.0.5	TCP	60 ldp > 40008 [ACK] Seq=45 Ack=459 Win=3670 Len=0
103 82.222191	198.18.0.3	198.18.0.5	LDP	98 Initialization Message Keep Alive Message
104 82.224312	198.18.0.2	198.18.0.5	LDP	458 Address Message Label Mapping Message Label Mapping Message Label Mapping
105 82.224464	198.18.0.5	198.18.0.3	TCP	60 42364 > ldp [ACK] Seq=37 Ack=45 Win=4084 Len=0
106 82.226507	198.18.240.3	198.18.0.5	LDP	480 Address Message Label Mapping Message Label Mapping Message Label Mapping
107 82.226762	198.18.0.5	198.18.0.2	TCP	60 40008 > ldp [ACK] Seq=459 Ack=449 Win=3680 Len=0
108 82.230778	198.18.0.5	198.18.240.3	TCP	60 ldp > 15785 [ACK] Seq=45 Ack=463 Win=3666 Len=0

FIGURE 9 – PE4 et ses voisins initient des sessions LDP. Ici on voit PE4->PE1, P->PE4, PE4->PE3.



223 136.923010	198.18.240.3	198.18.240.5	LDP	92 Label Mapping Message
224 136.925118	198.18.240.3	198.18.0.2	LDP	92 Label Mapping Message
225 136.929477	198.18.240.5	198.18.240.3	TCP	54 42772 > ldp [ACK] Seq=189 Ack=217 Win=3996 Len=0
226 136.931644	198.18.0.2	198.18.240.3	TCP	60 ldp > 40893 [ACK] Seq=235 Ack=235 Win=4034 Len=0

```
Transmission Control Protocol, Src Port: ldp (646), Dst Port: 42772 (42772), Seq: 179, Ack: 189, Len: 38
Label Distribution Protocol
   Version: 1
   PDU Length: 34
   LSR ID: 198.18.240.3 (198.18.240.3)
   Label Space ID: 0
 ▽ Label Mapping Message
      0... .... = U bit: Unknown bit not set
      Message Type: Label Mapping Message (0x400)
      Message Length: 24
      Message ID: 0x00000a40
   ▽ Forwarding Equivalence Classes TLV
        00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
        TLV Type: Forwarding Equivalence Classes TLV (0x100)
        TLV Length: 8
     ▽ FEC Elements
       ▽ FEC Element 1
            FEC Element Type: Prefix FEC (2)
            FEC Element Address Type: IPv4 (1)
            FEC Element Length: 32
            Prefix: 198.18.0.5
   ▽ Generic Label TLV
```

FIGURE 10 – La loopback de PE4 (et les préfixes d'interconnexion) se voit attribuer un label. Ici, on voit les échanges entre P<->PE4 et P<->PE1.

2.4.2 Ajout d'un nouveau VPN sur un PE

En théorie, lors de l'ajout d'un nouveau VPN sur un PE, on va créer la VRF, mettre en place le routage interne à ce site VPN et annoncer ce site via BGP (avec les extensions Multiprotocol). On aura donc la signalisation du routage interne (OSPF ou RIP dans notre cas) puis BGP entre les PE puis à nouveau le routage interne sur l'autre site (redistribution depuis BGP). Évidemment, si les autres sites ne sont pas configurés, les tentatives BGP échoueront. Évidemment, il n'y a pas d'échanges LDP, le label associé au VPN étant échangé via BGP.

Vérification : j'ajoute le VPN B sur PE2 qui, pour l'instant, est configuré uniquement pour le VPN A.

1	0.000000000	198.18.44.1	224.0.0.5	OSPF	94 Hello Packet
2	0.006154000	198.18.44.0	198.18.44.1	OSPF	78 DB Description
3	0.008288000	198.18.44.1	198.18.44.0	OSPF	78 DB Description
4	0.010431000	198.18.44.1	198.18.44.0	OSPF	98 DB Description
5	0.012455000	198.18.44.0	198.18.44.1	OSPF	98 DB Description
6	0.014598000	198.18.44.0	198.18.44.1	OSPF	70 LS Request
7	0.014623000	198.18.44.1	198.18.44.0	OSPF	78 DB Description
8	0.016693000	198.18.44.0	198.18.44.1	OSPF	78 DB Description
9	0.016732000	198.18.44.1	198.18.44.0	OSPF	70 LS Request
10	0.018817000	198.18.44.1	198.18.44.0	OSPF	110 LS Update
11	0.020854000	198.18.44.0	198.18.44.1	OSPF	98 LS Update
12	0.020879000	198.18.44.1	198.18.44.0	OSPF	78 DB Description
13	0.504289000	198.18.44.0	224.0.0.5	OSPF	98 LS Update
14	0.506516000	198.18.44.1	224.0.0.5	OSPF	110 LS Update
15	0.543813000	198.18.44.1	224.0.0.5	OSPF	94 LS Update
16	2.534310000	198.18.44.0	224.0.0.5	OSPF	98 LS Acknowledge
17	2.536574000	198.18.44.1	224.0.0.5	OSPF	78 LS Acknowledge
18	5.490557000	198.18.44.0	198.18.44.1	OSPF	98 LS Update
19	5.494881000	198.18.44.1	198.18.44.0	OSPF	110 LS Update
20	7.993141000	198.18.44.1	224.0.0.5	OSPF	78 LS Acknowledge
21	7.997285000	198.18.44.0	224.0.0.5	OSPF	78 LS Acknowledge

FIGURE 11 – Le routage propre au VPN (ici, OSPF) se met en marche entre PE2 et CE3. PE2 va désormais avoir le nouveau site VPN B (L2 de CE3) et les préfixes d'interconnexions dans ses tables.

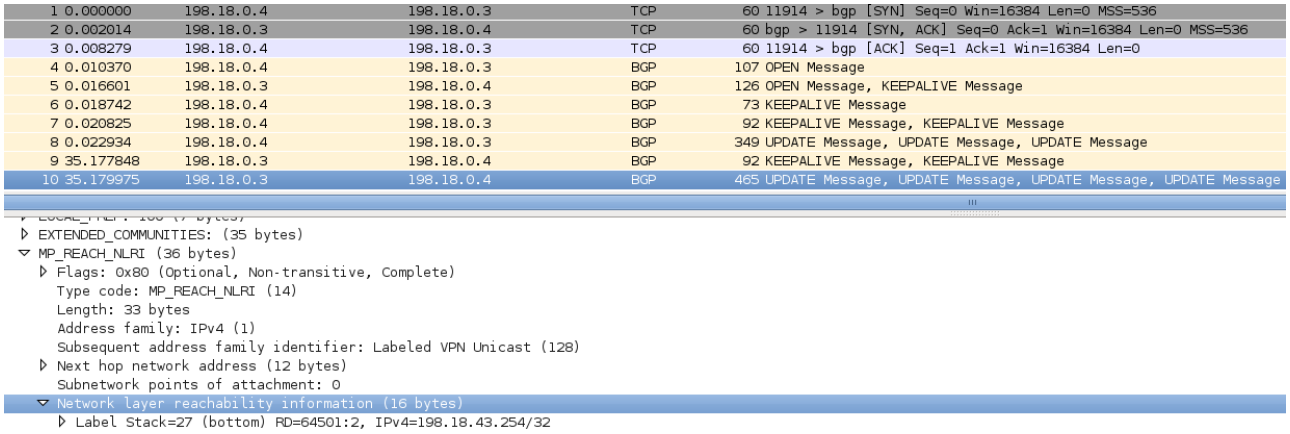


FIGURE 12 – PE2 annonce le nouveau site VPN B (L2 de CE3) aux autres sites du VPN B (PE3 dans cette illustration) et apprend les préfixes des autres sites du VPN B.

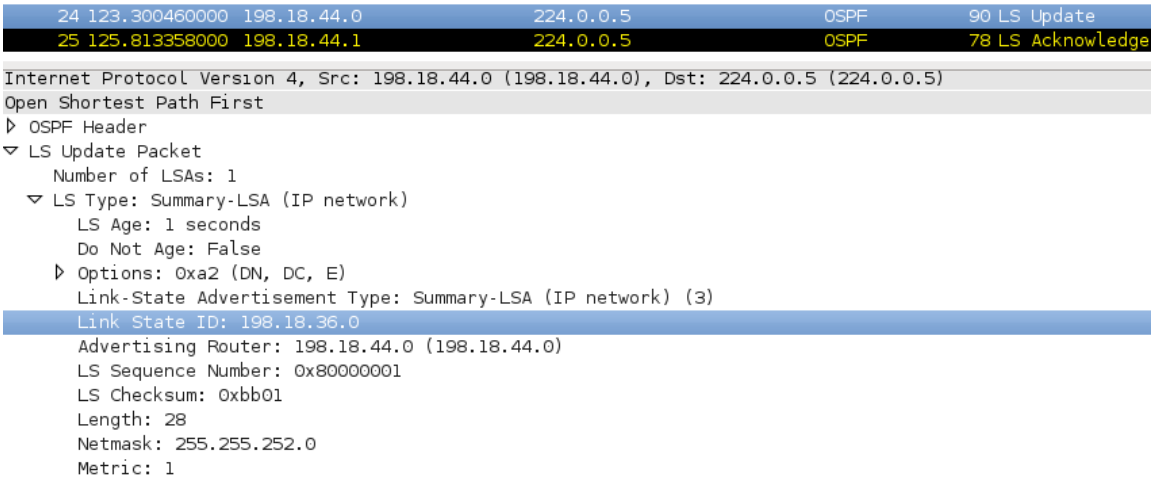


FIGURE 13 – PE2 propage, à l'intérieur du site, les autres préfixes des sites distants du VPN B. Ici : le site 2 appris via PE3.

2.4.3 Ajout d'un nouveau préfixe dans un site existant

En théorie, lors de l'ajout d'un nouveau préfixe dans un site existant, le routage interne à ce site VPN (OSPF ou RIP dans notre cas) va propager le nouveau préfixe. Puis le PE, via BGP (et les extensions Multiprotocol), va propager ce nouveau préfixe entre les sites. Sur les autres sites, le préfixe sera propagé via le protocole de routage interne propre à ce site (OSPF/RIP). Évidemment, il n'y a pas d'échanges LDP, le label associé au préfixe du VPN étant échangé via BGP.

Vérification : on crée une nouvelle loopback sur CE3, L3 : 198.18.43.253/22

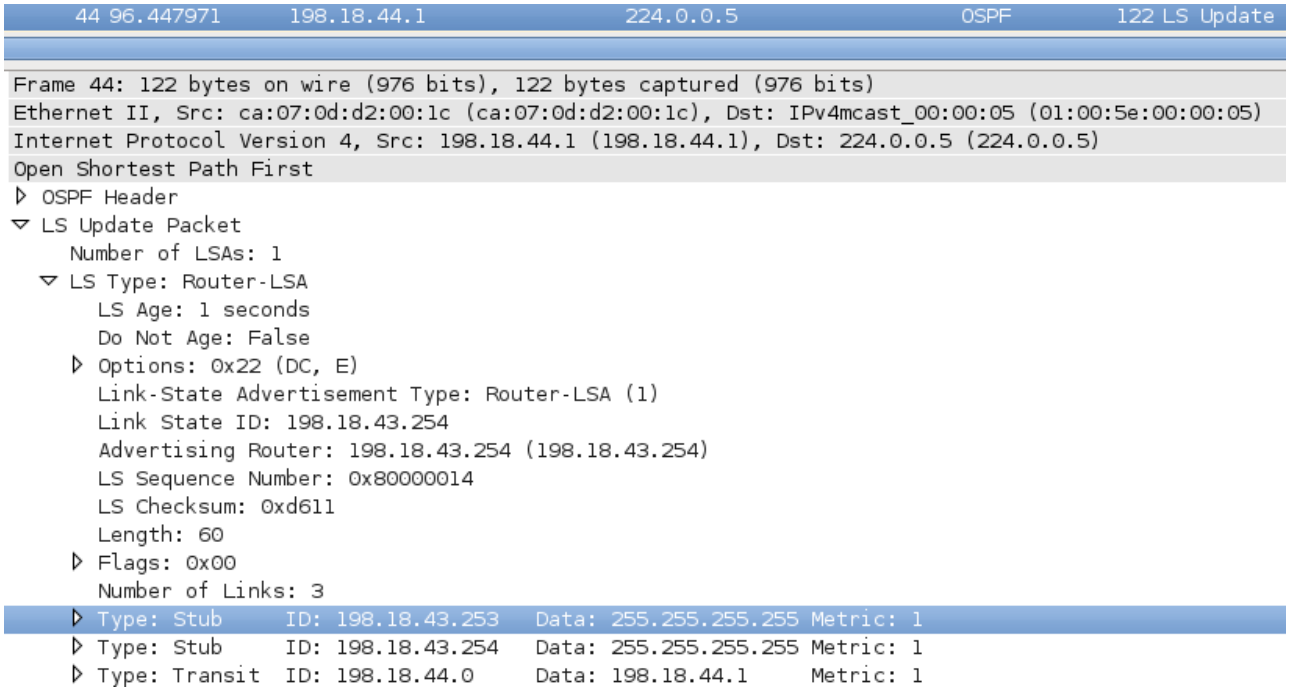


FIGURE 14 – CE3 annonce, via OSPF, sa nouvelle loopback.

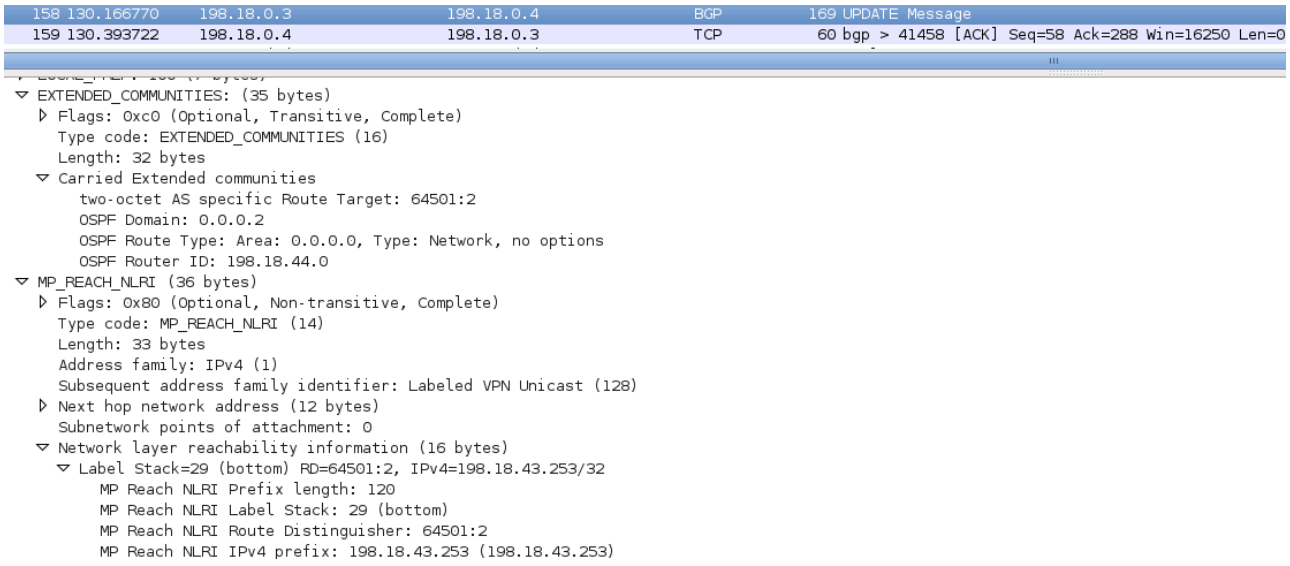


FIGURE 15 – PE2 annonce, via iBGP, le nouveau préfixe existant sur le site VPN aux autres sites du VPN (ici : PE3).

2.5 Question 5

Note : entre chaque topologie, je reviens à la configuration initiale de ma maquette telle que présentée à la question 1.

2.5.1 Mesh

Avec cette topologie, chaque site peut communiquer directement avec tous les autres sites d'un même VPN. Illustration :

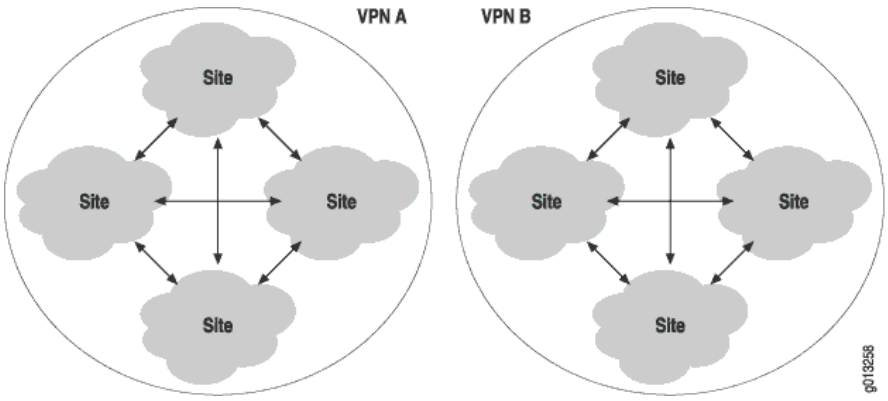


FIGURE 16 – Une topologie full-mesh. Source : `https://www.juniper.net/techpubs/software/erx/junose93/swconfig-bgp-mpls/using-route-targets-to-configure-vpn-topologies.html`.

Principaux avantages/inconvénients :

- \+ Toujours le chemin le plus court entre deux PE (donc entre deux sites)
- \+ Redondance "par design"

La topologie actuelle de ma maquette est du full-mesh : pour un même VPN, chaque site (chaque PE) est relié directement à chaque autre site (à chaque autre PE). Pour chaque site, on utilise la même valeur de route-target en import et en export.

2.5.2 Hub

Avec cette topologie, chaque site peut communiquer directement uniquement avec le site central (le hub) d'un même VPN mais il ne peut pas communiquer directement (= sans passer par le hub) avec les autres sites d'un même VPN. Illustration :

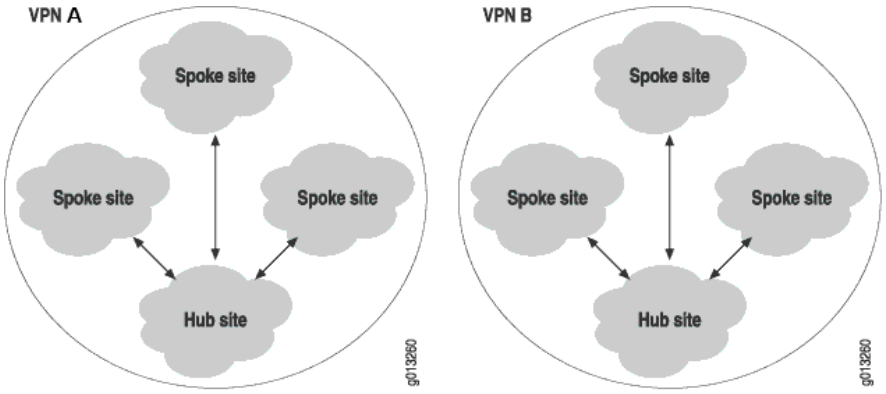


FIGURE 17 – Une topologie Hub-and-Spoke. D'après : `https://www.juniper.net/techpubs/software/erx/junose93/swconfig-bgp-mpls/using-route-targets-to-configure-vpn-topologies.html`.

Cette topologie peut se présenter sous deux sous-types distincts : soit on isole les sites "spoke", soit on leur permet de communiquer à la condition que les communications entre "spoke" passent par le hub.

Pour comprendre l'intérêt du premier sous-type, prenons un exemple (repris de `http://www.brimbelle.org/mattieu/projects/bgpmpls/etat-art.htm` et complété) : une société commerciale a son siège, son entrepôt, son usine, ses magasins, ... répartis partout en France. Depuis le siège social, où se trouve la direction informatique, on souhaite pouvoir se connecter à tous les sites (usine, entrepôts, chaque magasin, ...) pour recueillir des informations (monitoring) ou pour dépanner les utilisateurs. Chaque site doit donc pouvoir communiquer avec le siège mais pas avec les autres sites.

L'intérêt du deuxième sous-type est de permettre, plus que le premier sous-type, la centralisation des politiques (de communication entre sites, de sécurité, ...).

Principaux avantages/inconvénients :

- \+ Centralisation des politiques
- \+ Passage à l'échelle (full-mesh : explosion du nombre de liens entre PE)
- \- Pas toujours le plus court chemin entre deux PE

Pour obtenir cette topologie, il faut manipuler les route-target. Il nous faut deux communautés bien distinctes : 64501 :1 et 64501 :3. Le hub importe selon la communauté 64501 :1 et exporte la communauté 64501 :3. Les sites spoke importent selon la communauté 64501 :3 et exportent selon la communauté 64501 :1. Ainsi, deux sites spoke ne peuvent communiquer entre eux : ils vont refuser l'annonce de l'autre car ce n'est pas la bonne communauté (celle présentée : 64501 :1, attendue : 64501 :3). Seules les annonces du hub seront acceptées par les sites spoke.

Pour mettre cela en pratique, j'ai décidé que PE2 sera le hub du VPN A et qu'il n'y aura aucune communication entre les sites spoke.

Il faut donc adapter la configuration actuelle. Sur PE2 :

```
ip vrf vpna
  no route-target export 64501:1
  route-target export 64501:3
```

Sur PE1 et PE3 :

```
ip vrf vpna
  no route-target import 64501:1
  route-target import 64501:3
```

On obtient le comportement désiré, le hub possède tous les préfixes des sites du VPN A dans sa table :

```
PE2#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
C       198.18.28.0 is directly connected, FastEthernet4/0
B       198.18.28.2 [200/0] via 198.18.0.4, 00:00:13
B    198.18.24.0/22 [200/0] via 198.18.0.2, 00:00:28
R    198.18.16.0/22 [120/1] via 198.18.28.1, 00:00:10, FastEthernet4/0
B    198.18.20.0/22 [200/1] via 198.18.0.4, 00:00:13
```

Les sites spoke possèdent uniquement les préfixes du site 1 (PE2-CE1, hub) et ceux de leur propre site :

```
PE1#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 1 subnets
B       198.18.28.0 [200/0] via 198.18.0.3, 00:02:00
C    198.18.24.0/22 is directly connected, Loopback2
B    198.18.16.0/22 [200/1] via 198.18.0.3, 00:02:00


PE3#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
B       198.18.28.0 [200/0] via 198.18.0.3, 00:01:20
C       198.18.28.2 is directly connected, FastEthernet4/0
B    198.18.16.0/22 [200/1] via 198.18.0.3, 00:01:20
R    198.18.20.0/22 [120/1] via 198.18.28.3, 00:00:23, FastEthernet4/0
```

Le site 1 (hub) peut être joint depuis n'importe quel autre site (spoke) :

```
PE1#ping vrf vpna 198.18.19.254 source 198.18.27.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/18/20 ms

CE2#ping 198.18.19.254 source 198.18.23.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/12/16 ms
```

Mais la communication entre les deux sites spoke est interdite :

```
PE1#ping vrf vpna 198.18.23.254 source 198.18.27.254
.....
Success rate is 0 percent (0/5)

CE2#ping 198.18.27.254 source 198.18.23.254
.....
Success rate is 0 percent (0/5)
```

Pour que nos spokes puissent communiquer entre eux (en passant par le hub), il faudrait rajouter un PE et un CE qui seraient les deux composantes de notre hub. Selon la documentation de Cisco, la plupart du temps, on utilise (e|i)BGP entre le CE hub et le PE hub. La configuration semble se résumer à une configuration BGP puis à la création de VRF supplémentaires (une pour recevoir les préfixes des spoke, l'autre les préfixes du hub (ceux des spoke redistribués en réalité)). Je n'ai pas effectué cette configuration car c'est répétitif vis-à-vis des questions précédentes.

2.5.3 Configuration originale

Il existe un grand nombre de topologies (multi-VPN, multilevel Hub-and-spoke, managed network, extranet, ...) ainsi que des déclinaisons (mesh partiel, hub-and-spoke redondé, ...) ainsi que des combinaisons (hub-and-spoke + full-mesh, ...). Il est donc difficile de choisir quelle topologie choisir et illustrer.

J'ai d'abord éliminé les topologies complexes (combinaisons) et les topologies nécessitant plus de machines pour montrer l'intérêt. Parmi les topologies restantes, j'ai choisi l'overlapping VPN, site commun à plusieurs VPN simplement parce qu'il répondait à la question que je me posais : comment offrir un même service à tous les sites de tous (ou partie) des VPN ?

Avec cette topologie, on peut offrir un ou plusieurs services centralisés (connectivité vers Internet, résolveur DNS, station de monitoring, ...) à tous les sites de plusieurs VPN. Illustration :

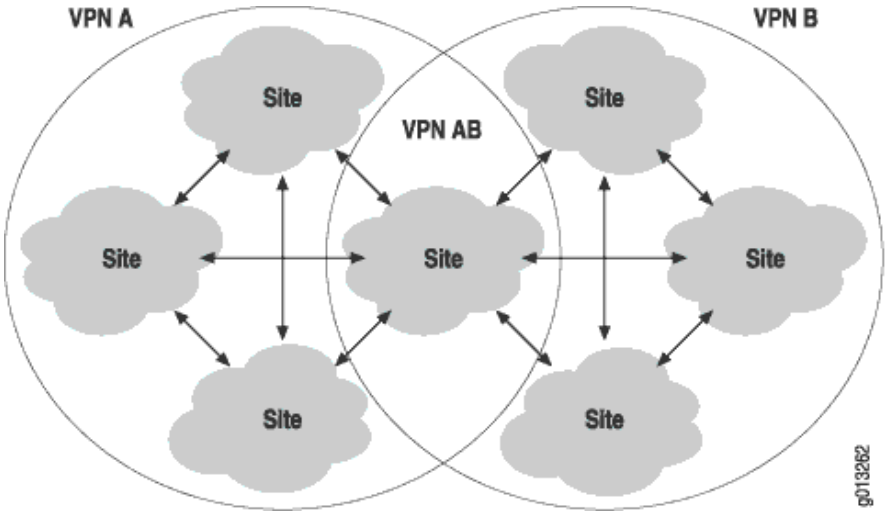


FIGURE 18 – Une topologie avec un site commun (Overlapping VPN). Source : `https://www.juniper.net/techpubs/software/erx/junose93/swconfig-bgp-mpls/using-route-targets-to-configure-vpn-topologies.html`.

Pour obtenir cette topologie, il faut manipuler les route-target. Le site commun doit importer et exporter les deux communautés : celle du VPN A et celle du VPN B.

Pour mettre cela en pratique, j'ai décidé que PE2 sera le PE commun. On créera une nouvelle loopback depuis laquelle on annoncera le préfixe qui devra être commun (198.18.48.0/22). On ne déploie pas de protocole de routage dynamique (RIP, OSPF) sur ce nouveau site car c'est répétitif vis-à-vis des questions précédentes.

Voici le bout de configuration à ajouter sur PE2 :

```
ip vrf vpnab
 rd 64501:12
 route-target export 64501:1
 route-target export 64501:2
 route-target import 64501:1
 route-target import 64501:2

interface Loopback2
 ip vrf forwarding vpnab
 ip address 198.18.48.1 255.255.252.0

router bgp 64501
 address-family ipv4 vrf vpnab
  redistribute connected
 exit-address-family
```

On obtient le comportement désiré : notre site commun reçoit tous les préfixes de tous les autres sites, qu'il fasse partie du VPN A ou B.

```
PE2#sh ip route vrf vpnab
     198.18.43.0/32 is subnetted, 1 subnets
B       198.18.43.254 [20/2] via 198.18.44.1 (vpnb), 00:00:10, FastEthernet5/0
     198.18.44.0/31 is subnetted, 1 subnets
B       198.18.44.0 is directly connected, 00:00:10, FastEthernet5/0
     198.18.28.0/31 is subnetted, 2 subnets
B       198.18.28.0 is directly connected, 00:00:10, FastEthernet4/0
B       198.18.28.2 [200/0] via 198.18.0.4, 00:00:10
B    198.18.24.0/22 [200/0] via 198.18.0.2, 00:00:10
B    198.18.32.0/22 [200/0] via 198.18.0.5, 00:00:10
C    198.18.48.0/22 is directly connected, Loopback2
B    198.18.16.0/22 [20/1] via 198.18.28.1 (vpna), 00:00:10, FastEthernet4/0
B    198.18.36.0/22 [200/0] via 198.18.0.4, 00:00:11
B    198.18.20.0/22 [200/1] via 198.18.0.4, 00:00:11
```

Le préfixe du site commun est propagé dans tous les sites, VPN A ou B :

```
PE1#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
B       198.18.28.0 [200/0] via 198.18.0.3, 00:05:55
B       198.18.28.2 [200/0] via 198.18.0.4, 00:05:55
C    198.18.24.0/22 is directly connected, Loopback2
B    198.18.48.0/22 [200/0] via 198.18.0.3, 00:05:55
B    198.18.16.0/22 [200/1] via 198.18.0.3, 00:05:55
B    198.18.20.0/22 [200/1] via 198.18.0.4, 00:05:55


PE3#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
B       198.18.28.0 [200/0] via 198.18.0.3, 00:03:57
C       198.18.28.2 is directly connected, FastEthernet4/0
B    198.18.24.0/22 [200/0] via 198.18.0.2, 00:03:57
B    198.18.48.0/22 [200/0] via 198.18.0.3, 00:03:57
B    198.18.16.0/22 [200/1] via 198.18.0.3, 00:03:57
R    198.18.20.0/22 [120/1] via 198.18.28.3, 00:00:22, FastEthernet4/0
```

```
PE3#sh ip route vrf vpnb
     198.18.43.0/32 is subnetted, 1 subnets
B       198.18.43.254 [200/2] via 198.18.0.3, 00:04:03
     198.18.44.0/31 is subnetted, 1 subnets
B       198.18.44.0 [200/0] via 198.18.0.3, 00:04:03
B    198.18.32.0/22 [200/0] via 198.18.0.5, 00:04:03
B    198.18.48.0/22 [200/0] via 198.18.0.3, 00:04:03
C    198.18.36.0/22 is directly connected, Loopback2

PE4#sh ip route vrf vpnb
     198.18.43.0/32 is subnetted, 1 subnets
B       198.18.43.254 [200/2] via 198.18.0.3, 00:05:13
     198.18.44.0/31 is subnetted, 1 subnets
B       198.18.44.0 [200/0] via 198.18.0.3, 00:05:13
C    198.18.32.0/22 is directly connected, Loopback2
B    198.18.48.0/22 [200/0] via 198.18.0.3, 00:05:13
B    198.18.36.0/22 [200/0] via 198.18.0.4, 00:05:13
```

Chaque site, qu'il soit du VPN A ou B, peut joindre le préfixe commun :

```
CE1#ping 198.18.48.1 source 198.18.19.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/12/36 ms

PE1#ping vrf vpna 198.18.48.1 source 198.18.27.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/12 ms

CE3#ping 198.18.48.1 source 198.18.43.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/12/32 ms

PE4#ping vrf vpnb 198.18.48.1 source 198.18.35.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
```

3 VPN avec TE

3.1 Question 1

Les commandes supplémentaires (activation d'OSPF-TE, MPLS-TE, ...) pour répondre à cette question se trouvent en annexe. Nous ne configurerons pas la capacité réservable sur chaque lien pour l'instant. Comme nous l'avons vu lors du précédent TP, OSPF-TE annoncera donc des capacités réservables nulles.

Nous allons construire un tunnel explicite PE2 -> PE3. Voici les commandes à utiliser, sur PE2 :

```
interface Tunnel1
ip unnumbered Loopback1
tunnel destination 198.18.0.4
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng autoroute announce

ip explicit-path name explicitlongtunnel enable
next-address 198.18.240.8
next-address 198.18.240.0
next-address 198.18.240.5
exit

interface Tunnel1
tunnel mpls traffic-eng path-option 1 explicit name explicitlongtunnel
no sh
```

On constate que le tunnel est bien construit et suit le chemin que nous voulons :

```
sh mpls traffic-eng tunnels Tunnel1

Name: PE2_t1                              (Tunnel1) Destination: 198.18.0.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, type explicit explicitlongtunnel (Basis for Setup, path weight 3)
[...]
  InLabel  :  -
```

```
  OutLabel : FastEthernet1/0, 26
  RSVP Signalling Info:
       Src 198.18.0.3, Dst 198.18.0.4, Tun_Id 1, Tun_Instance 1
    RSVP Path Info:
      My Address: 198.18.240.9
      Explicit Route: 198.18.240.8 198.18.240.1 198.18.240.0 198.18.240.4
                      198.18.240.5 198.18.0.4
      Record Route:  NONE
[...]
  Shortest Unconstrained Path Info:
    Path Weight: 1 (TE)
    Explicit Route: 198.18.240.15 198.18.240.14 198.18.0.4
  History:
    Tunnel:
      Time since created: 13 seconds
      Time since path change: 14 seconds
    Current LSP:
      Uptime: 14 seconds

PE2#traceroute 198.18.0.4
  1 198.18.240.8 [MPLS: Label 26 Exp 0] 16 msec 12 msec 12 msec
  2 198.18.240.0 [MPLS: Label 26 Exp 0] 12 msec 12 msec 12 msec
  3 198.18.240.5 16 msec *  12 msec
```

On constate que tout le trafic qu'il est plus avantageux de faire passer par le tunnel passera par le tunnel. Pour les préfixes où le tunnel n'est pas plus avantageux, il est choisi comme chemin de secours :

```
PE2#sh ip route
     198.18.240.0/31 is subnetted, 8 subnets
O       198.18.240.4 [110/2] via 0.0.0.0, 00:11:45, Tunnel1
O       198.18.240.6 [110/2] via 198.18.240.10, 00:11:45, FastEthernet2/0
                     [110/2] via 198.18.240.8, 00:11:45, FastEthernet1/0
O       198.18.240.0 [110/2] via 198.18.240.8, 00:11:45, FastEthernet1/0
O       198.18.240.2 [110/2] via 198.18.240.10, 00:11:45, FastEthernet2/0
O       198.18.240.12 [110/2] via 198.18.240.10, 00:11:45, FastEthernet2/0
                      [110/2] via 0.0.0.0, 00:11:45, Tunnel1
C       198.18.240.14 is directly connected, FastEthernet3/0
```

```
C       198.18.240.8 is directly connected, FastEthernet1/0
C       198.18.240.10 is directly connected, FastEthernet2/0
     198.18.0.0/32 is subnetted, 5 subnets
O       198.18.0.4 [110/2] via 0.0.0.0, 00:11:45, Tunnel1
O       198.18.0.5 [110/2] via 198.18.240.8, 00:11:46, FastEthernet1/0
O       198.18.0.1 [110/2] via 198.18.240.10, 00:11:47, FastEthernet2/0
O       198.18.0.2 [110/3] via 198.18.240.10, 00:11:47, FastEthernet2/0
                   [110/3] via 198.18.240.8, 00:11:47, FastEthernet1/0
                   [110/3] via 0.0.0.0, 00:11:47, Tunnel1
C       198.18.0.3 is directly connected, Loopback1
```

Cela vaut aussi pour le trafic de nos VPN. Regardons la VRF associée au VPN A sur PE2 :

```
PE2#sh ip route vrf vpna
     198.18.28.0/31 is subnetted, 2 subnets
C       198.18.28.0 is directly connected, FastEthernet4/0
B       198.18.28.2 [200/0] via 198.18.0.4, 00:09:51
B    198.18.24.0/22 [200/0] via 198.18.0.2, 1d08h
R    198.18.16.0/22 [120/1] via 198.18.28.1, 00:00:09, FastEthernet4/0
B    198.18.20.0/22 [200/1] via 198.18.0.4, 00:09:51
```

Pour joindre 198.18.20.0/22, BGP nous a informés qu'il faut envoyer le trafic à 198.18.0.4. Qui est 198.18.0.4 ? Rien dans notre VRF. Regardons dans la table de routage IP globale :

```
PE2#sh ip route
O       198.18.0.4 [110/2] via 0.0.0.0, 00:11:45, Tunnel1
```

Donc pour joindre 198.18.0.4 qui nous permet de joindre un site distant du VPN A, il faut passer par Tunnel1. Vérifions :

```
PE2#traceroute vrf vpna 198.18.23.254
  1 198.18.240.8 [MPLS: Labels 26/25 Exp 0] 16 msec 16 msec 16 msec
  2 198.18.240.0 [MPLS: Labels 26/25 Exp 0] 12 msec 12 msec 16 msec
  3 198.18.28.2 [MPLS: Label 25 Exp 0] 12 msec 12 msec 12 msec
  4 198.18.28.3 16 msec *  20 msec
```

Néanmoins, la documentation de Cisco nous informe que les tunnels sont unidirectionnels. PE3 n'empruntera donc pas ce tunnel pour répondre à PE2. Cela se confirme en lisant la table de routage de PE3 :

```
PE3#sh ip route
(...]
O       198.18.0.3 [110/2] via 198.18.240.15, 03:08:08, FastEthernet3/0
```

3.2 Question 2

Pour répondre à cette question, nous supprimons le tunnel de la question précédente. Nous allons recréer ce tunnel PE2 -> PE3 mais cette fois-ci, non plus avec une exigence sur le chemin exact à suivre mais avec une exigence de capacité disponible : nous voulons une capacité disponible de 50 megas.

Comme nous avons déjà vu ce type de configuration lors du TP précédent et pour me simplifier la tâche, j'ai décidé que seul le chemin PE2->PE4->P->PE1->PE3 sera en mesure d'acheminer 50 megas. Ces liens seront en effet configurés comme disposant d'une capacité réservable de 60 megas. Tous les autres liens du réseau opérateur seront configurés comme permettant seulement 30 megas. Les commandes nécessaires à cette configuration sont disponibles en annexe.

Nous construisons notre tunnel :

```
interface Tunnel1
ip unnumbered Loopback1
tunnel destination 198.18.0.4
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 1 dynamic
no sh
```

On constate que notre tunnel est fonctionnel :

```
PE2#sh mpls traffic-eng tunnels tunnel 1
Name: PE2_t1                              (Tunnel1) Destination: 198.18.0.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, type dynamic (Basis for Setup, path weight 4)

  Config Parameters:
    Bandwidth: 50000    kbps (Global)  Priority: 1  1   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
```

```
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 50000    bw-based

  InLabel  :  -
  OutLabel : FastEthernet1/0, 26
  RSVP Signalling Info:
       Src 198.18.0.3, Dst 198.18.0.4, Tun_Id 1, Tun_Instance 20
    RSVP Path Info:
      My Address: 198.18.240.9
      Explicit Route: 198.18.240.8 198.18.240.6 198.18.240.7 198.18.240.3
                      198.18.240.2 198.18.240.4 198.18.240.5 198.18.0.4
      Record Route:  NONE
      Tspec: ave rate=50000 kbits, burst=1000 bytes, peak rate=50000 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=50000 kbits, burst=1000 bytes, peak rate=50000 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 1 (TE)
    Explicit Route: 198.18.240.15 198.18.240.14 198.18.0.4

PE2#traceroute 198.18.0.4
  1 198.18.240.8 [MPLS: Label 26 Exp 0] 12 msec 12 msec 16 msec
  2 198.18.240.7 [MPLS: Label 24 Exp 0] 16 msec 12 msec 12 msec
  3 198.18.240.2 [MPLS: Label 26 Exp 0] 16 msec 12 msec 12 msec
  4 198.18.240.5 12 msec *  28 msec

PE2#traceroute vrf vpnb 198.18.39.254
  1 198.18.240.8 [MPLS: Labels 26/27 Exp 0] 16 msec 16 msec 16 msec
  2 198.18.240.7 [MPLS: Labels 24/27 Exp 0] 12 msec 12 msec 12 msec
  3 198.18.240.2 [MPLS: Labels 26/27 Exp 0] 20 msec 12 msec 12 msec
  4 198.18.39.254 16 msec *  36 msec
```

On constate que ce tunnel est toujours unidirectionnel et que le trafic acheminable via ce tunnel est toujours le même :

- trafic à destination de 198.18.0.4 ou de toute autre loopback de PE3 et de tout autre préfixe disponible uniquement en passant par PE3 ;
- ce qui inclu le trafic à destination du site 2 du VPN A et du site 2 du VPN B.

3.3 Question 3

En théorie, il est possible de définir des FEC avec une granularité plus fine, c'est même un des objectifs de MPLS : appliquer un traitement différencié, assurer une qualité de service différente à des flux critiques.

Néanmoins, mes recherches montrent que les commandes IOS nécessaires ne sont pas disponibles sur l'image IOS que nous utilisons.

Remarquons néanmoins que, dans la pratique, ce type de configuration est difficilement (main)tenable sur un réseau d'opérateur. Les opérateurs préfèrent donc utiliser les classes de trafic établies par le groupe de travail DiffServ (voir : `http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsdserv3.html`).

3.4 Question 4

La question est ambiguë : nous n'avons pas de MPLS entre nos CE et nos PE car les CE sont destinées à être installé chez le client et doivent donc avoir une configuration simple.

Il existe plusieurs modes pour choisir les labels MPLS :

- par VRF Egress ;
- par interface de sortie. Ce mode permet d'éviter une itération dans la table de routage en réception ;
- par préfixe. Seul ce mode permet de faire de la QoS.

Sur des routeurs Cisco, la commande « mpls label mode vrf vrf-name | all-vrfs protocol bgp-vpnv4 per-prefix | per-vrf » permet de choisir une attribution par VRF ou par préfixe. Cette commande est disponible sur la série 6500 et/ou à partir de la version 15 de l'IOS. Cette commande n'est donc pas disponible sur l'image IOS que nous utilisons.

Nous remarquons que, sur nos routeurs, le choix des labels se fait par préfixe. Exemple, sur PE2, pour le VPN A :

```
PE2#sh ip vrf detail vpna
VRF vpna; default RD 64501:1; default VPNID <not set>
  Interfaces:
    Fa4/0
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:64501:1
```

```
Import VPN route-target communities
  RT:64501:1
No import route-map
No export route-map
VRF label distribution protocol: not configured
VRF label allocation mode: per-prefix
```

3.5 Question 5

Comme je l'ai déjà écrit pour les questions 1 et 2, si le next-hop dans la VRF est aussi la destination d'un tunnel TE, alors le trafic du VPN passera par aussi le tunnel sans configuration supplémentaire.

Pour pouvoir choisir un tunnel différent par VPN (mes exemples se basent sur PE2 et PE3), il faudrait :

- Que PE2 ait une loopback supplémentaire, loopback 2 ;
- Que PE2 annonce, à PE3, son préfixe VPN A avec, comme next-hop, l'adresse de sa loopback 1 et son préfixe VPN B avec, comme next-hop, l'adresse de sa loopback 2 ;
- Que PE3 construise 2 tunnels, un à destination de la loopback 1 de PE2 et l'autre, à destination de la loopback 2.

Même si l'on arrivait à re-écrire le next-hop (avec une route-map, ou avec la commande bgp next-hop dans notre VRF, par exemple), on ne pourra pas créer les deux tunnels TE. En effet, comme nous l'avons montré dans la première et la deuxième question, le tunnel TE englobe toutes les loopback d'un routeur. Donc PE3 empruntera aussi le premier tunnel TE pour joindre la deuxième loopback de PE2.

Donc il n'est pas possible de choisir des tunnels TE différents en fonction du VPN si les sites distants de chaque VPN sont sur le même PE.

En revanche, il est tout à fait possible de le faire quand que les PE source et destination n'ont pas plusieurs VPN. Ainsi, sur ma maquette, un tunnel TE PE2->PE1 permet de faire passer le trafic à destination du site 3 du VPN A alors qu'un tunnel TE PE2->PE4 permet de faire passer le trafic à destination du site 1 du VPN B. Les deux tunnels ne sont pas incompatibles.

4 Bibliographie

Dans l'ordre d'utilisation.

- Mise en place d'un VPN MPLS - http://ccie.julienberton.fr/2012/01/01/mise-en-place-dun-vpn-m
- MPLS and VPN Architectures - 2
- MPLS VPN Concepts - http://www.cisco.com/en/US/docs/net_mgmt/ip_solution_center/5.0.1/mpls_vpn/user/guide/isc_mpls.html
- État de l'art des VPN BGP / MPLS - http://www.brimbelle.org/mattieu/projects/bgpmpls/etat-art.htm
- MPLS VPN Hub and Spoke Topology - https://www.youtube.com/watch?v=KBHwp4nEwxE
- Using Route Targets to Configure VPN Topologies - https://www.juniper.net/techpubs/software/erx/junose93/swconfig-bgp-mpls/using-route-targets-to-configure-vpn-topologies.html
- MPLS TE explicit-path - http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/TE_1208S.html#wp9131
- Cours de Jean-Jacques Pansiot - http://www-r2.u-strasbg.fr/~pansiot/enseignement/Reseaux-Opera VPN-BGP-MPLS.pdf
- MPLS VPN—Per VRF Label - http://www.cisco.com/en/US/docs/ios/mpls/configuration/guide/mp_vpn_per_vrf_lbl.pdf
- MPLS VPN traffic engineering tunnel selection - http://www.gossamer-threads.com/lists/cisco/nsp/85083

A Annexes

A.1 Fichier de configuration Dynagen pour tout le TP

```
[localhost]
    [[7200]]
        image = ./C7200-AD.BIN
        ram = 256
idlepc = 0x607335f0

    [[ROUTER P]]
        model = 7200
        f1/0 = PE1 f2/0
        f2/0 = PE4 f2/0
f3/0 = PE2 f2/0
f4/0 = PE3 f2/0

    [[ROUTER PE1]]
        model = 7200
        f1/0 = PE4 f1/0
f3/0 = PE3 f1/0

    [[ROUTER PE2]]
        model = 7200
f1/0 = PE4 f3/0
f3/0 = PE3 f3/0
f4/0 = CE1 f1/0
f5/0 = CE3 f1/0

    [[ROUTER PE3]]
        model = 7200
f4/0 = CE2 f1/0

    [[ROUTER PE4]]
        model = 7200

    [[ROUTER CE1]]
```

```
    model = 7200

[[ROUTER CE2]]
    model = 7200

[[ROUTER CE3]]
    model = 7200
```

A.2 Commandes IOS pour effectuer la configuration "de base" des routeurs pour tout le TP

A.2.1 P

```
enable
conf t
hostname P
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
exit


int fastEthernet 1/0
ip address 198.18.240.3 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 2/0
ip address 198.18.240.7 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 3/0
ip address 198.18.240.10 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 4/0
ip address 198.18.240.12 255.255.255.254
mpls ip
no shutdown
exit
int loopback 1
ip address 198.18.0.1 255.255.255.255
no shutdown
```

```
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp

exit
copy r s
```

A.2.2 PE1

```
enable
conf t
hostname PE1
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
exit

ip vrf vpna
rd 64501:1
route-target both 64501:1
exit

int fastEthernet 1/0
ip address 198.18.240.0 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 2/0
ip address 198.18.240.2 255.255.255.254
mpls ip
no shutdown
```

```
exit
int fastEthernet 3/0
ip address 198.18.240.4 255.255.255.254
mpls ip
no shutdown
exit
int loopback 1
ip address 198.18.0.2 255.255.255.255
no shutdown
exit
int loopback 2
ip vrf forwarding vpna
ip address 198.18.27.254 255.255.252.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp

router rip
version 2
address-family ipv4 vrf vpna
network 198.18.24.0
network 198.18.25.0
network 198.18.26.0
network 198.18.27.0
no auto-summary
redistribute bgp 64501 metric 1
exit
exit

router bgp 64501
```

```
neighbor 198.18.0.3 remote-as 64501
neighbor 198.18.0.3 update-source loopback1
neighbor 198.18.0.4 remote-as 64501
neighbor 198.18.0.4 update-source loopback1

address-family vpnv4
neighbor 198.18.0.3 activate
neighbor 198.18.0.3 send-community extended
neighbor 198.18.0.4 activate
neighbor 198.18.0.4 send-community extended
exit

address-family ipv4 vrf vpna
redistribute rip ! ou network 198.18.24.0 mask 255.255.252.0 car RIP se justifie pas vraiment
exit
exit
exit
copy r s
```

A.2.3 PE2

```
enable
conf t
hostname PE2
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
exit

ip vrf vpna
rd 64501:1
route-target both 64501:1
exit

ip vrf vpnb
rd 64501:2
```

```
route-target both 64501:2
exit

int fastEthernet 1/0
ip address 198.18.240.9 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 2/0
ip address 198.18.240.11 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 3/0
ip address 198.18.240.15 255.255.255.254
mpls ip
no shutdown
exit
interface F4/0
ip vrf forwarding vpna
ip address 198.18.28.0 255.255.255.254
no shutdown
exit
int F5/0
ip vrf forwarding vpnb
ip address 198.18.44.0 255.255.255.254
no sh
exit
int loopback 1
ip address 198.18.0.3 255.255.255.255
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit
```

```
ip cef
mpls label protocol ldp


router rip
version 2
address-family ipv4 vrf vpna
network 198.18.28.0
no auto-summary
redistribute bgp 64501 metric 1
exit
exit

router ospf 2 vrf vpnb
network 198.18.44.0 0.0.3.255 area 0
redistribute bgp 64501 subnets
exit

router bgp 64501
neighbor 198.18.0.2 remote-as 64501
neighbor 198.18.0.2 update-source loopback1
neighbor 198.18.0.4 remote-as 64501
neighbor 198.18.0.4 update-source loopback1
neighbor 198.18.0.5 remote-as 64501
neighbor 198.18.0.5 update-source loopback1

address-family vpnv4
neighbor 198.18.0.2 activate
neighbor 198.18.0.2 send-community extended
neighbor 198.18.0.4 activate
neighbor 198.18.0.4 send-community extended
neighbor 198.18.0.5 activate
neighbor 198.18.0.5 send-community extended
exit
```

```
address-family ipv4 vrf vpna
redistribute rip
exit

address-family ipv4 vrf vpnb
redistribute ospf 2 vrf vpnb
exit
exit
exit
copy r s
```

A.2.4 PE3

```
enable
conf t
hostname PE3
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
exit

ip vrf vpna
rd 64501:1
route-target both 64501:1
exit

ip vrf vpnb
rd 64501:2
route-target both 64501:2
exit

int fastEthernet 1/0
ip address 198.18.240.5 255.255.255.254
mpls ip
no shutdown
exit
```

```
int fastEthernet 2/0
ip address 198.18.240.13 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 3/0
ip address 198.18.240.14 255.255.255.254
mpls ip
no shutdown
exit
interface F4/0
ip vrf forwarding vpna
ip address 198.18.28.2 255.255.255.254
no shutdown
exit
int loopback 1
ip address 198.18.0.4 255.255.255.255
no shutdown
exit
int loopback 2
ip vrf forwarding vpnb
ip address 198.18.39.254 255.255.252.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp

router rip
version 2
address-family ipv4 vrf vpna
network 198.18.28.0
```

```
no auto-summary
redistribute bgp 64501 metric 1
exit
exit

router ospf 2 vrf vpnb
network 198.18.36.0 0.0.3.255  area 0
redistribute bgp 64501 subnets
exit

router bgp 64501
neighbor 198.18.0.2 remote-as 64501
neighbor 198.18.0.2 update-source loopback1
neighbor 198.18.0.3 remote-as 64501
neighbor 198.18.0.3 update-source loopback1
neighbor 198.18.0.5 remote-as 64501
neighbor 198.18.0.5 update-source loopback1

address-family vpnv4
neighbor 198.18.0.2 activate
neighbor 198.18.0.2 send-community extended
neighbor 198.18.0.3 activate
neighbor 198.18.0.3 send-community extended
neighbor 198.18.0.5 activate
neighbor 198.18.0.5 send-community extended
exit

address-family ipv4 vrf vpna
redistribute rip
exit

address-family ipv4 vrf vpnb
redistribute ospf 2 vrf vpnb
exit
exit
exit
```

```
copy r s
```

A.2.5 PE4

```
enable
conf t
hostname PE4
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
exit

ip vrf vpnb
rd 64501:2
route-target both 64501:2
exit

int fastEthernet 1/0
ip address 198.18.240.1 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 2/0
ip address 198.18.240.6 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 3/0
ip address 198.18.240.8 255.255.255.254
mpls ip
no shutdown
exit
int loopback 1
ip address 198.18.0.5 255.255.255.255
no shutdown
exit
```

```
int loopback 2
ip vrf forwarding vpnb
ip address 198.18.35.254 255.255.252.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp

router ospf 2 vrf vpnb
network 198.18.32.0 0.0.3.255 area 0
redistribute bgp 64501 subnets
exit

router bgp 64501
neighbor 198.18.0.3 remote-as 64501
neighbor 198.18.0.3 update-source loopback1
neighbor 198.18.0.4 remote-as 64501
neighbor 198.18.0.4 update-source loopback1

address-family vpnv4
neighbor 198.18.0.3 activate
neighbor 198.18.0.3 send-community extended
neighbor 198.18.0.4 activate
neighbor 198.18.0.4 send-community extended
exit

address-family ipv4 vrf vpnb
redistribute ospf 2 vrf vpnb
exit
exit
exit
```

```
copy r s
```

A.2.6 CE1

```
enable
conf t
hostname CE1
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
exit

int loopback 1
ip address 198.18.19.254 255.255.252.0
no shutdown
int fastEthernet 1/0
ip address 198.18.28.1 255.255.255.254
no shutdown
exit

router rip
network 198.18.16.0
network 198.18.17.0
network 198.18.18.0
network 198.18.19.0
network 198.18.28.0
no auto-summary
version 2
exit
exit
copy r s
```

A.2.7 CE2

```
enable
conf t
```

```
hostname CE2
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
exit

int fastEthernet 1/0
ip address 198.18.28.3 255.255.255.254
no shutdown
exit
int loopback 1
ip address 198.18.23.254 255.255.252.0
no shutdown
exit

router rip
network 198.18.20.0
network 198.18.21.0
network 198.18.22.0
network 198.18.23.0
network 198.18.28.0
no auto-summary
version 2
exit
exit
copy r s
```

A.2.8 CE3

```
enable
conf t
hostname CE3
no ip domain lookup
line console 0
exec-timeout 0 0
logging synchronous
```

```
exit



int F1/0
ip address 198.18.44.1 255.255.255.254
no sh
exit
int loopback 1
ip address 198.18.43.254 255.255.252.0
no sh
exit

router ospf 1
network 198.18.40.0 0.0.3.255 area 0
network 198.18.44.0 0.0.3.255 area 0
exit
exit
copy r s
```

A.3 Commandes IOS pour effectuer la configuration supplémentaire pour la première question de la seconde partie

A.3.1 P

```
conf t
mpls traffic-eng tunnels
router ospf 1
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0
exit

int F1/0
mpls traffic-eng tunnels
int F2/0
mpls traffic-eng tunnels
int F3/0
mpls traffic-eng tunnels
int F4/0
mpls traffic-eng tunnels
exit
exit
```

A.3.2 PE1

```
conf t
mpls traffic-eng tunnels
router ospf 1
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0
exit

int F1/0
mpls traffic-eng tunnels
int F2/0
mpls traffic-eng tunnels
int F3/0
mpls traffic-eng tunnels
```

```
exit
exit
```

A.3.3 PE2

```
conf t
mpls traffic-eng tunnels
router ospf 1
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0
exit

int F1/0
mpls traffic-eng tunnels
int F2/0
mpls traffic-eng tunnels
int F3/0
mpls traffic-eng tunnels
exit
exit
```

A.3.4 PE3

```
conf t
mpls traffic-eng tunnels
router ospf 1
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0
exit

int F1/0
mpls traffic-eng tunnels
int F2/0
mpls traffic-eng tunnels
int F3/0
mpls traffic-eng tunnels
exit
exit
```

A.3.5 PE4

```
conf t
mpls traffic-eng tunnels
router ospf 1
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0
exit

int F1/0
mpls traffic-eng tunnels
int F2/0
mpls traffic-eng tunnels
int F3/0
mpls traffic-eng tunnels
exit
exit
```

A.4 Commandes IOS pour effectuer la configuration supplémentaire pour la deuxième question de la seconde partie

A.4.1 P

```
conf t
int F1/0
ip rsvp bandwidth 60000 60000

int F2/0
ip rsvp bandwidth 60000 60000

int F3/0
ip rsvp bandwidth 30000 30000

int F4/0
ip rsvp bandwidth 30000 30000
exit
exit
```

A.4.2 PE1

```
conf t
int F1/0
ip rsvp bandwidth 30000 30000

int F2/0
ip rsvp bandwidth 60000 60000

int F3/0
ip rsvp bandwidth 60000 60000
exit
exit
```

A.4.3 PE2

```
conf t
int F1/0
ip rsvp bandwidth 60000 60000
```

```
int F2/0
ip rsvp bandwidth 30000 30000

int F3/0
ip rsvp bandwidth 30000 30000
exit
exit
```

A.4.4 PE3

```
conf t
int F1/0
ip rsvp bandwidth 60000 60000

int F2/0
ip rsvp bandwidth 30000 30000

int F3/0
ip rsvp bandwidth 30000 30000
exit
exit
```

A.4.5 PE4

```
conf t
int F1/0
ip rsvp bandwidth 30000 30000

int F2/0
ip rsvp bandwidth 60000 60000

int F3/0
ip rsvp bandwidth 30000 30000
exit
exit
```

Table des figures

Table des matières