

## Réseaux d'opérateur - TP2 - MPLS

---

Guillaume LUCAS

Hamza HMAMA

# Sommaire

<b>Sommaire</b>	<b>2</b>
<b>1 Avant de commencer</b>	<b>3</b>
1.1 Adressage . . . . .	3
1.2 Schéma . . . . .	3
<b>2 Question 1</b>	<b>5</b>
<b>3 Question 2</b>	<b>6</b>
<b>4 Question 3</b>	<b>9</b>
<b>5 Question 4</b>	<b>10</b>
<b>6 Question 5</b>	<b>10</b>
<b>7 Question 6</b>	<b>11</b>
<b>8 Question 7</b>	<b>14</b>
<b>9 Question 8</b>	<b>17</b>
<b>10 Question 9</b>	<b>19</b>
<b>11 Question 10</b>	<b>21</b>
<b>12 Bibliographie</b>	<b>32</b>
<b>A Annexes</b>	<b>33</b>
A.1 Fichier de configuration Dynagen pour tout le TP . . . . .	33
A.2 Commandes IOS pour tout le TP . . . . .	34
A.3 Commandes IOS supplémentaires spécifiques à la question 10 . . . . .	40
<b>Table des figures</b>	<b>46</b>
<b>Table des matières</b>	<b>47</b>

# 1 Avant de commencer

## 1.1 Adressage

### 1.1.1 IPv4

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

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

1. Les premiers /20 sont dédiés aux "services" : chacune des loopback demandées dans l'énoncé se trouvera dans un /20 différent.
2. Le milieu est réservé pour un usage futur (expansion des "services", expansion des interconnexions, ...).
3. Le dernier /20 est dédié aux interconnexions des routeurs internes. Il sera découpé en /31, chaque /31 étant dédié à une interconnexion.

Nous ne prévoyons pas d'avoir plus de 2048 interconnexions internes mais si c'était le cas, on prendrait 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, ...).

Les préfixes de nos loopback seront :

- R1 - P1 : 198.18.15.254/20
- R2 - P2 : 198.18.31.254/20
- R3 - P3 : 198.18.47.254/20
- R4 - P4 : 198.18.63.254/20
- R5 - P5 : 198.18.79.254/20
- R6 - P6 : 198.18.95.254/20

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 notre plan d'adressage.

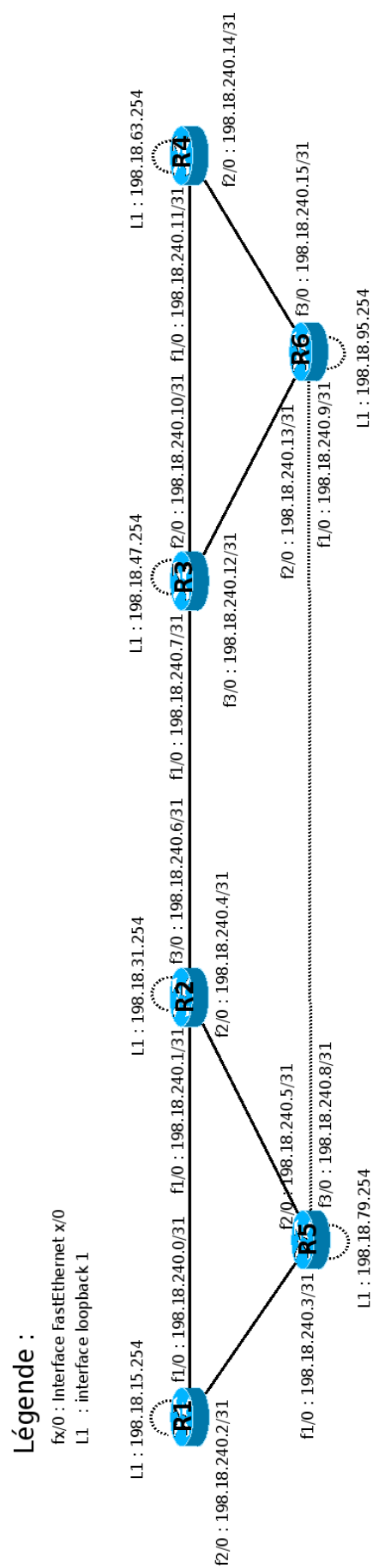


FIGURE 1 – Schéma de notre maquette. D'après l'énoncé.

## 2 Question 1

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

La configuration des routeurs se fait facilement : il n'y a pas de commandes que nous n'avons pas vues lors du premier TP ou qui n'ont pas été données lors de la séance de TP. Les configurations intégrales de nos routeurs se trouvent en annexe 2.

Pour illustration, voici un extrait de la LIB (Label Information Base) et la LFIB (Label Forwarding Information Base) de notre routeur R1 :

```
R1#show mpls ldp bindings
```

```
[...]
```

```
    tib entry: 198.18.31.254/32, rev 8
```

```
local binding:  tag: 16
```

```
remote binding:  tsr: 198.18.79.254:0, tag: 26
```

```
    tib entry: 198.18.47.254/32, rev 24
```

```
local binding:  tag: 23
```

```
remote binding:  tsr: 198.18.79.254:0, tag: 19
```

```
remote binding:  tsr: 198.18.31.254:0, tag: 23
```

```
    tib entry: 198.18.63.254/32, rev 26
```

```
local binding:  tag: 24
```

```
remote binding:  tsr: 198.18.79.254:0, tag: 27
```

```
remote binding:  tsr: 198.18.31.254:0, tag: 24
```

```
[...]
```

```
    tib entry: 198.18.240.10/31, rev 18
```

```
        local binding:  tag: 20
```

```
        remote binding:  tsr: 198.18.79.254:0, tag: 22
```

```
        remote binding:  tsr: 198.18.31.254:0, tag: 19
```

```
    tib entry: 198.18.240.12/31, rev 16
```

```
        local binding:  tag: 19
```

```
        remote binding:  tsr: 198.18.79.254:0, tag: 18
```

```
        remote binding:  tsr: 198.18.31.254:0, tag: 20
```

```
    tib entry: 198.18.240.14/31, rev 20
```

```
        local binding:  tag: 21
```

```
        remote binding:  tsr: 198.18.79.254:0, tag: 23
```

```
        remote binding:  tsr: 198.18.31.254:0, tag: 21
```

```
R1#show mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	198.18.31.254/32	0	Fa1/0	198.18.240.1
17	Pop tag	198.18.240.4/31	0	Fa2/0	198.18.240.3
	Pop tag	198.18.240.4/31	0	Fa1/0	198.18.240.1
18	Pop tag	198.18.240.6/31	0	Fa1/0	198.18.240.1
19	20	198.18.240.12/31	0	Fa1/0	198.18.240.1
20	19	198.18.240.10/31	0	Fa1/0	198.18.240.1
21	21	198.18.240.14/31	0	Fa1/0	198.18.240.1
22	Pop tag	198.18.240.8/31	0	Fa2/0	198.18.240.3
23	23	198.18.47.254/32	0	Fa1/0	198.18.240.1
24	24	198.18.63.254/32	0	Fa1/0	198.18.240.1
25	Untagged	198.18.79.254/32	0	Fa2/0	198.18.240.3
26	26	198.18.95.254/32	0	Fa1/0	198.18.240.1
27	27	198.18.111.254/32	0	Fa1/0	198.18.240.1

On observe aussi que chaque routeur crée automatiquement un LSP vers la loopback de chaque autre routeur, ce qui est le comportement attendu.

### 3 Question 2

Pour répondre à cette question, nous nous sommes mis dans la situation où nous pingons P4 en mettant P1 en adresse source. Il est toujours mieux de pinger d'une loopback à l'autre. Nous ferons ainsi pour le reste de ce TP.

On interroge récursivement les LFIB de nos routeurs. D'abord pour l'aller :

```
R1#show mpls forwarding-table 198.18.63.254
```

24	24	198.18.63.254/32	0	Fa1/0	198.18.240.1 ! -> R2
----	----	------------------	---	-------	----------------------

```
R2#show mpls forwarding-table 198.18.63.254
```

24	24	198.18.63.254/32	0	Fa3/0	198.18.240.7 ! -> R3
----	----	------------------	---	-------	----------------------

```
R3#show mpls forwarding-table 198.18.63.254
```

24	Untagged	198.18.63.254/32	0	Fa2/0	198.18.240.11 ! -> R4
----	----------	------------------	---	-------	-----------------------

Puis pour le retour :

```
R4#show mpls forwarding-table 198.18.15.254
```

```
22      22      198.18.15.254/32  0      Fa1/0      198.18.240.10 ! -> R3
```

```
R3#show mpls forwarding-table 198.18.15.254
```

```
22      16      198.18.15.254/32  0      Fa1/0      198.18.240.6 ! -> R2
```

```
R2#show mpls forwarding-table 198.18.15.254
```

```
16      Untagged  198.18.15.254/32  0      Fa1/0      198.18.240.0 ! -> R1
```

À l'aller, on aura donc le chemin et les labels MPLS suivants : R1 -> [24] -> R2 -> [24] -> R3 -> [] -> R4

Au retour, on aura le chemin et les labels MPLS suivants : R4 -> [22] -> R3 -> [16] -> R2 -> [] -> R1

Vérifications avec des traceroute :

```
R1#traceroute 198.18.63.254 source 198.18.15.254
```

```
1 198.18.240.1 [MPLS: Label 24 Exp 0] 8 msec 20 msec 12 msec
2 198.18.240.7 [MPLS: Label 24 Exp 0] 8 msec 8 msec 8 msec
3 198.18.240.11 44 msec * 16 msec
```

```
R4#traceroute 198.18.15.254 source 198.18.63.254
```

```
1 198.18.240.10 [MPLS: Label 22 Exp 0] 12 msec 44 msec 44 msec
2 198.18.240.6 [MPLS: Label 16 Exp 0] 40 msec 8 msec 12 msec
3 198.18.240.0 8 msec * 16 msec
```

Vérification avec nos captures réseau. À l'aller :

### R1 F1/0

```
▶ Frame 77: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▶ Ethernet II, Src: ca:00:59:7d:00:1c (ca:00:59:7d:00:1c), Dst: ca:01:59:7d:00:1c (ca:01:59:7d:00:1c)
▶ MultiProtocol Label Switching Header, Label: 24, Exp: 0, S: 1, TTL: 255
▶ Internet Protocol Version 4, Src: 198.18.15.254 (198.18.15.254), Dst: 198.18.63.254 (198.18.63.254)
▶ Internet Control Message Protocol
```

### R2 F3/0

```
▶ Frame 77: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▶ Ethernet II, Src: ca:01:59:7d:00:54 (ca:01:59:7d:00:54), Dst: ca:02:59:7d:00:1c (ca:02:59:7d:00:1c)
▶ MultiProtocol Label Switching Header, Label: 24, Exp: 0, S: 1, TTL: 254
▶ Internet Protocol Version 4, Src: 198.18.15.254 (198.18.15.254), Dst: 198.18.63.254 (198.18.63.254)
▶ Internet Control Message Protocol
```

### R3 F2/0

```
▶ Frame 56: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
▶ Ethernet II, Src: ca:02:59:7d:00:38 (ca:02:59:7d:00:38), Dst: ca:03:59:7d:00:1c (ca:03:59:7d:00:1c)
▶ Internet Protocol Version 4, Src: 198.18.15.254 (198.18.15.254), Dst: 198.18.63.254 (198.18.63.254)
▶ Internet Control Message Protocol
```

### R4 F1/0

```
▶ Frame 38: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
▶ Ethernet II, Src: ca:02:59:7d:00:38 (ca:02:59:7d:00:38), Dst: ca:03:59:7d:00:1c (ca:03:59:7d:00:1c)
▶ Internet Protocol Version 4, Src: 198.18.15.254 (198.18.15.254), Dst: 198.18.63.254 (198.18.63.254)
▶ Internet Control Message Protocol
```

FIGURE 2 – Capture réseau. Ping P4 (198.18.63.254) depuis P1 (198.18.15.254). Aller.

Au retour :

### R4 F1/0

```
▶ Frame 39: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▶ Ethernet II, Src: ca:03:59:7d:00:1c (ca:03:59:7d:00:1c), Dst: ca:02:59:7d:00:38 (ca:02:59:7d:00:38)
▶ MultiProtocol Label Switching Header, Label: 22, Exp: 0, S: 1, TTL: 255
▶ Internet Protocol Version 4, Src: 198.18.63.254 (198.18.63.254), Dst: 198.18.15.254 (198.18.15.254)
▶ Internet Control Message Protocol
```

### R3 F2/0

```
▶ Frame 57: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▶ Ethernet II, Src: ca:03:59:7d:00:1c (ca:03:59:7d:00:1c), Dst: ca:02:59:7d:00:38 (ca:02:59:7d:00:38)
▶ MultiProtocol Label Switching Header, Label: 22, Exp: 0, S: 1, TTL: 255
▶ Internet Protocol Version 4, Src: 198.18.63.254 (198.18.63.254), Dst: 198.18.15.254 (198.18.15.254)
▶ Internet Control Message Protocol
```

### R2 F3/0

```
▶ Frame 78: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▶ Ethernet II, Src: ca:02:59:7d:00:1c (ca:02:59:7d:00:1c), Dst: ca:01:59:7d:00:54 (ca:01:59:7d:00:54)
▶ MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 1, TTL: 254
▶ Internet Protocol Version 4, Src: 198.18.63.254 (198.18.63.254), Dst: 198.18.15.254 (198.18.15.254)
▶ Internet Control Message Protocol
```

### R1 F1/0

```
▶ Frame 78: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
▶ Ethernet II, Src: ca:01:59:7d:00:1c (ca:01:59:7d:00:1c), Dst: ca:00:59:7d:00:1c (ca:00:59:7d:00:1c)
▶ Internet Protocol Version 4, Src: 198.18.63.254 (198.18.63.254), Dst: 198.18.15.254 (198.18.15.254)
▶ Internet Control Message Protocol
```

FIGURE 3 – Capture réseau. Ping P4 (198.18.63.254) depuis P1 (198.18.15.254). Retour.



## 4 Question 3

Avant de répondre, regardons les LFIB de R1, R5, R2 et R3 :

```
R1#show mpls forwarding-table 198.18.63.254
```

24	24	198.18.63.254/32	0	Fa1/0	198.18.240.1
----	----	------------------	---	-------	--------------

```
R5#show mpls forwarding-table 198.18.63.254
```

27	24	198.18.63.254/32	0	Fa2/0	198.18.240.4
----	----	------------------	---	-------	--------------

```
R2#show mpls forwarding-table 198.18.63.254
```

24	24	198.18.63.254/32	0	Fa3/0	198.18.240.7
----	----	------------------	---	-------	--------------

```
R3#show mpls forwarding-table 198.18.63.254
```

24	Untagged	198.18.63.254/32	2474	Fa2/0	198.18.240.11
----	----------	------------------	------	-------	---------------

On constate bien que, quelle que soit l'interface d'entrée dans R2, un paquet à destination de P4 sera toujours tagué avec le label 24. Pour commuter un paquet destiné à P4, R3 attend qu'il soit tagué avec le label 24, peu importe quel routeur, "caché" derrière R2, a envoyé ce paquet. Cela est tout à fait logique : il y a un label par destination par routeur.

Un même label ne peut pas être utilisé dans deux sens sur un même lien car il y a un label par destination et il faut éviter la formation de boucles. Supposons que, pour le retour de la question 2, R4 aurait envoyé un paquet tagué avec le label 24 à R3. Pour R3, 24 est associé à 198.18.63.254 via R4. R3 aurait donc transféré ce paquet à R4, non à R2 (pour que lui-même transmette à R1).

Un même label peut être utilisé sur plusieurs liens d'un même routeur mais pour une même destination uniquement. Illustration :

```
R5#traceroute 198.18.111.254 source 198.18.79.254
```

```
1 198.18.240.9 [MPLS: Label 26 Exp 0] 12 msec 12 msec 12 msec
2 198.18.240.12 [MPLS: Label 26 Exp 0] 12 msec 12 msec 12 msec
3 198.18.240.11 8 msec * 28 msec
```

Ici, on a coupé les liens R2<->R3 et R6<->R4. On constate qu'entre R5 et R6, le label utilisé est 26 qui est le même entre R6 et R3. On a donc deux liens, partant du même routeur (R6), qui portent le même label pour une même destination.

## 5 Question 4

La fonctionnalité penultimate hop popping permet de faire supprimer le label non pas par le routeur de sortie, l'Egress, mais par l'avant-dernier. Cela permet d'éviter au routeur Egress une double itération : l'une dans sa LFIB et l'autre dans sa table de routage IP. L'opération de recherche sur le label dans sa LFIB est inutile, car dans tous les cas le routeur devra effectuer une recherche dans sa table de routage IP. En utilisant la fonctionnalité PHP, l'Egress doit simplement faire une recherche dans sa table de routage IP.

Par défaut, les routeurs Cisco font du penultimate popping. Source : <http://www.cisco.com/en/US/docs/routers/asr9000/software/mpls/configuration/guide/gcasr9kldp.html>

Pour constater ce phénomène, il suffit de revenir à la question 2 : quand R1 veut joindre la loopback de R4, c'est R3, qui est directement connecté à R4, qui supprime le label. On a donc du penultimate hop popping.

Si l'on faisait un traceroute entre deux adresses IP de lien (pas des loopback), comme 198.18.240.0- >198.18.240.11, on observerait que l'on perd un label MPLS supplémentaire (R2 supprimerait le label) ce qui correspond bien au comportement du PHP.

## 6 Question 5

Cette commande permet d'activer ou de désactiver la propagation du TTL depuis le paquet IP original vers le header MPLS. La désactivation de cette propagation est utilisée par les opérateurs qui veulent masquer les tunnels MPLS de leurs réseaux.

Par défaut, les routeurs Cisco propagent le TTL IP vers MPLS. Source : [http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp\\_m1.html#wp1013846](http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp_m1.html#wp1013846)

Quand la fonctionnalité est activée, un traceroute met en évidence tous les routeurs d'un chemin comme nous l'avons vu lors des questions précédentes.

Il suffit de désactiver la fonctionnalité sur l'Ingress pour qu'elle soit prise en compte. En effet, quelle que soit la configuration des autres routeurs, seule celle de l'Ingress compte : si ce dernier met un TTL fixe (255 sur Cisco, mais cela change selon les équipementiers) car la propagation est désactivée, il est ensuite trop tard pour revenir sur ce choix.

Illustration : nous désactivons le propagate-ttl sur R1 et nous faisons un traceroute vers R4 :

```
traceroute 198.18.63.254 source 198.18.15.254
 1 198.18.240.7 [MPLS: Label 23 Exp 0] 12 msec 8 msec 8 msec
 2 198.18.240.11 48 msec * 8 msec
```

On constate bien que l'absence de la propagation du TTL du paquet IP original vers le header MPLS. Cela fait que les routeurs intermédiaires du tunnel MPLS (ici : R2) disparaissent du traceroute.

## 7 Question 6

Cette commande permet de spécifier comment une réponse ICMP à un paquet ayant un TTL expiré est renvoyée : on utilise MPLS ou la table de routage IP ? Plus précisément, elle permet de définir un seuil. Si le nombre de labels dans le paquet reçu est supérieur à ce seuil, alors le paquet sera commuté avec MPLS. Sinon, on utilisera le routage IP classique.

La commande `mpls ip ttl-expiration pop 1` est la commande la plus utilisée par les opérateurs pour activer le transfert (forward) basé sur plusieurs labels.

Le détail de la fonctionnalité est donné en : [http://www.cisco.com/en/US/docs/ios/mps/command/reference/mp\\_m1.html#wp1013945](http://www.cisco.com/en/US/docs/ios/mps/command/reference/mp_m1.html#wp1013945)

Vérification :

Dans un premier temps, on utilise la configuration par défaut, c'est-à-dire sans la fonctionnalité `ttl-expiration pop`.

On fait alors un `traceroute` de 100 paquets avec un TTL fixé à 1 entre la loopback de R1 et celle de R4. R2 va recevoir les paquets. Il va les supprimer et générer un message ICMP TTL expired pour chacun des 100 paquets. Ces paquets vont être transférés jusqu'à R4 qui va les renvoyer à R1 via R3 et R2.

Pour illustrer cela, nous regardons les statistiques de l'interface `f1/0` (qui est connectée à R3) de R4 : on remarque que R4 a reçu les 100 paquets ICMP et les a retransmis par la même interface.

Avant (ligne « Route cache ») :

```
R4#show int F1/0 stats
```

```
FastEthernet1/0
```

Switching path	Pkts In	Chars In	Pkts Out	Chars Out
Processor	35903	3130038	35580	3065492
<b>Route cache</b>	<b>1012</b>	<b>184184</b>	<b>1030</b>	<b>191580</b>
Total	36915	3314222	36610	3257072

Commande utilisée :

```
R1#traceroute
```

```
Protocol [ip]:
```

```
Target IP address: 198.18.63.254
```

```
Source address: 198.18.15.254
```

```
Numeric display [n]:
```

```
Timeout in seconds [3]: 1
```

```
Probe count [3]: 100
```

```

Minimum Time to Live [1]: 1
Maximum Time to Live [30]: 1
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 198.18.63.254

```

```

 1 198.18.240.1 [MPLS: Label 24 Exp 0] 68 msec 44 msec 12 msec 12 msec 12 msec
 8 msec 12 msec 8 msec 52 msec 8 msec 12 msec 44 msec 40 msec 8 msec 12 msec
12 msec 8 msec 44 msec 44 msec 12 msec 12 msec 12 msec 8 msec 12 msec 12 msec
[...]

```

Après (ligne « Route cache ») :

```
R4#show int F1/0 stats
```

```
FastEthernet1/0
```

Switching path	Pkts In	Chars In	Pkts Out	Chars Out
Processor	35912	3130722	35588	3066070
<b>Route cache</b>	<b>1112</b>	<b>202384</b>	<b>1130</b>	<b>210180</b>
Total	37024	3333106	36718	3276250

On voit bien qu'il y a eu précisément 100 paquets reçus et réémis.

Illustration :

872	883.001385	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
873	883.003485	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
874	883.014095	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
875	883.016121	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
876	883.026672	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
877	883.028701	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
878	883.039235	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
879	883.041262	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
880	883.053824	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
881	883.055877	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
882	883.066387	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
883	883.068413	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
884	883.078918	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
885	883.080975	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
886	883.123212	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
887	883.127298	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
888	883.142093	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
889	883.144119	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
890	883.154602	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
891	883.156695	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
892	883.202934	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
893	883.205024	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
894	883.246997	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
895	883.251200	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
896	883.261651	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
897	883.263699	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
898	883.274133	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
899	883.276184	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)
900	883.286646	198.18.240.1	198.18.15.254	ICMP	182 Time-to-live exceeded (Time to live exceeded in transit)
901	883.288674	198.18.240.1	198.18.15.254	ICMP	186 Time-to-live exceeded (Time to live exceeded in transit)

FIGURE 4 – On observe que R4 reçoit bien les paquets ICMP expired

Dans un deuxième temps, on utilise la fonctionnalité ttl-expiration pop 1 et on reproduit l'expérience donnée ci-dessus. R2 va donc recevoir les 100 paquets. Puisqu'on a un seul label, R2 va popper le label et émettre des messages ICMP TTL expired en utilisant la table de routage IP. Cela fait que, dans ce cas, les messages ICMP vont être retransmis directement à R1 sans passer par R3 et R4.

Avant :

```
R4#show int F1/0 stats
```

```
FastEthernet1/0
```

Switching path	Pkts In	Chars In	Pkts Out	Chars Out
Processor	35966	3135652	35640	3070754
<b>Route cache</b>	<b>1112</b>	<b>202384</b>	<b>1130</b>	<b>210180</b>
Total	37078	3338036	36770	3280934

Commande utilisée :

```
R1#traceroute
```

```
Protocol [ip]:
```

```
Target IP address: 198.18.63.254
```

```
Source address: 198.18.15.254
```

```
Numeric display [n]:
```

```
Timeout in seconds [3]: 1
```

```
Probe count [3]: 100
```

```
Minimum Time to Live [1]: 1
```

```
Maximum Time to Live [30]: 1
```

```
Port Number [33434]:
```

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

```
Type escape sequence to abort.
```

```
Tracing the route to 198.18.63.254
```

```

  1 198.18.240.1 [MPLS: Label 24 Exp 0] 32 msec 4 msec 4 msec 4 msec 4 msec
    8 msec 32 msec 4 msec 4 msec 4 msec 8 msec 8 msec 32 msec 4 msec 4 msec
    8 msec 4 msec 4 msec 40 msec 4 msec 4 msec 4 msec 4 msec 4 msec 40 msec
    4 msec
[...]
```

On remarque un RTT moyen inférieur à celui obtenu lors de la première étape ce qui tend à confirmer, dans le contexte de notre maquette (la réalité demanderait plus de prudence), que le chemin de retour est raccourci.

Après :

```
R4#show int F1/0 stats
```

```
FastEthernet1/0
```

Switching path	Pkts In	Chars In	Pkts Out	Chars Out
Processor	35968	3135822	35644	3071060
<b>Route cache</b>	<b>1112</b>	<b>202384</b>	<b>1130</b>	<b>210180</b>
Total	37080	3338206	36774	3281240

## 8 Question 7

Il y a quatre types de tunnels :

1. Explicit : support du RFC 4950 et activation du propagate-ttl. On voit donc chaque routeur et l'on sait qu'il y a un tunnel MPLS.
2. Implicit : pas de support du RFC 4950 et activation du propagate-ttl. On voit donc chaque routeur mais l'on ne sait pas qu'il y a un tunnel MPLS.
3. Opaque : support du RFC 4950 et désactivation du propagate-ttl. On voit donc uniquement le dernier routeur du tunnel MPLS ainsi que la destination et l'on sait qu'il y a un tunnel MPLS sans avoir les détails du tunnel (routeurs intermédiaires).
4. Invisible : pas de support du RFC 4950 et désactivation du propagate-ttl. On ne voit pas les routeurs intermédiaires du tunnel et l'on ne sait pas qu'il y a un tunnel MPLS sur le chemin.

RFC4950 \ ttl-propagate	Enabled	Disabled
	Explicit	Opaque
Enabled	Explicit	Opaque
Disabled	Implicit	Invisible

FIGURE 5 – Les différents types de tunnels MPLS en fonction des fonctions supportées/activées.  
Source : <http://www.slideshare.net/edge7/slides-mpls-tunnel>.

Quand on utilise propagate-ttl sur l'Ingress, on obtient un tunnel explicite. Exemple : de la loopback de R1 à celle de R4.

```
R1#traceroute 198.18.63.254 source 198.18.15.254
```

```
1 198.18.240.1 [MPLS: Label 24 Exp 0] 8 msec 8 msec 8 msec
2 198.18.240.7 [MPLS: Label 23 Exp 0] 8 msec 8 msec 8 msec
3 198.18.240.11 8 msec * 16 msec
```

Quand on désactive `propagate-ttl` sur l'Ingress, on obtient un tunnel opaque. Exemple : de la loopback de R1 à celle de R4.

```
R1#traceroute 198.18.63.254 source 198.18.15.254
 1 198.18.240.7 [MPLS: Label 23 Exp 0] 8 msec 8 msec 44 msec
 2 198.18.240.11 8 msec * 8 msec
```

On ne peut pas obtenir les deux autres types de tunnels car cela nécessite de désactiver les extensions du RFC 4950 et il n'y a pas de commandes sur l'IOS pour faire ça. La seule solution est d'utiliser une ancienne image de l'IOS sortie avant la publication du RFC 4950.

Nous avons essayé quelques images (localisées avec un moteur de recherche et « intitle : "index of" IOS [7200] ») sans succès : même la version 12.0 de l'IOS de la série 7200 supporte les extensions du RFC 4950 ! Comme ce point n'est pas le plus fondamental de ce TP, nous n'avons pas insisté.

Par défaut, c'est-à-dire avec la propagation du TTL IP vers MPLS et la fonctionnalité `ttl-expiration` pop désactivée, les réponses `echo-reply` seront directement émises et transmises à la source (la machine qui a lancé l'`echo request`) alors que les messages ICMP `TTL-expired` seront transmis à la source en passant d'abord par le dernier routeur du tunnel. C'est ce que résume l'illustration suivante :

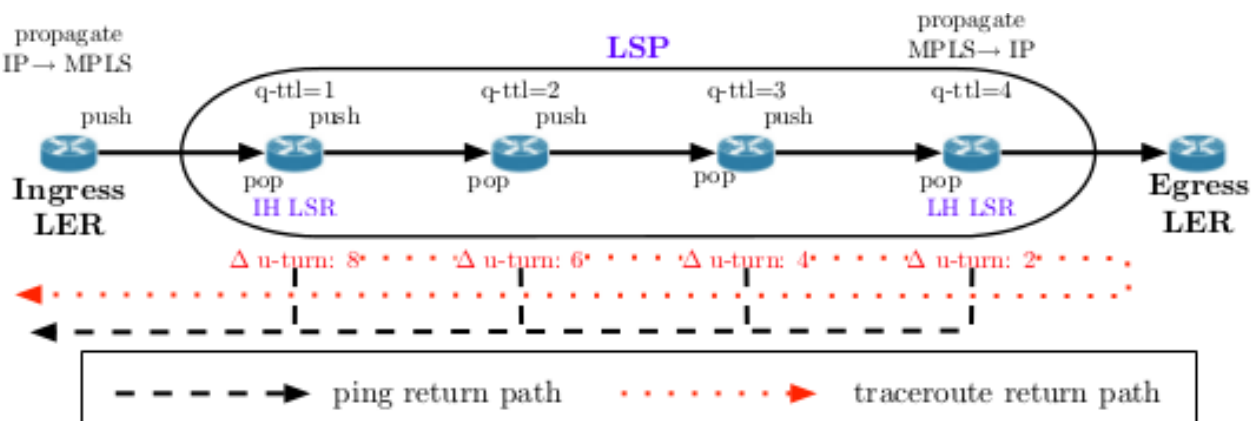


FIGURE 6 – Différence de chemin A/R entre un `echo request/reply` et un `ttl-expired`. Source : [http://www.caida.org/publications/papers/2012/revealing\\_mpls\\_tunnels/revealing\\_mpls\\_tunnels.pdf](http://www.caida.org/publications/papers/2012/revealing_mpls_tunnels/revealing_mpls_tunnels.pdf).

Illustration entre la loopback de R1 et celle de R4 :

```
R1#traceroute
Protocol [ip]:
Target IP address: 198.18.63.254
Source address: 198.18.15.254
Numeric display [n]:
Timeout in seconds [3]: 1
Probe count [3]: 1
```

Minimum Time to Live [1]:  
Maximum Time to Live [30]:  
Port Number [33434]:  
Loose, Strict, Record, Timestamp, Verbose[none]:  
Type escape sequence to abort.  
Tracing the route to 198.18.63.254

```

1 198.18.240.1 [MPLS: Label 23 Exp 0] 32 msec
2 198.18.240.7 [MPLS: Label 23 Exp 0] 48 msec
3 198.18.240.11 8 msec

```

Avec une capture réseau, nous voyons que le message ICMP TTL-expired de R2 (198.18.240.1) revient avec un TTL de 251 :

10474	10386.334577	198.18.15.254	198.18.63.254	UDP	60	Source port: 49265	Destination port: traceroute[Malformed Packet]
10475	10386.349500	198.18.240.1	198.18.15.254	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)	
10476	10386.351593	198.18.15.254	198.18.63.254	UDP	60	Source port: 49266	Destination port: 33435[Malformed Packet]
10477	10386.393215	198.18.240.7	198.18.15.254	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)	
10478	10386.395327	198.18.15.254	198.18.63.254	UDP	60	Source port: 49267	Destination port: 33436[Malformed Packet]
10479	10386.410065	198.18.240.11	198.18.15.254	ICMP	70	Destination unreachable (Port unreachable)	

Frame 10475: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits)							
Ethernet II, Src: ca:01:3c:66:00:1c (ca:01:3c:66:00:1c), Dst: ca:00:3c:66:00:1c (ca:00:3c:66:00:1c)							
Internet Protocol Version 4, Src: 198.18.240.1 (198.18.240.1), Dst: 198.18.15.254 (198.18.15.254)							
Version: 4							
Header length: 20 bytes							
Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))							
Total Length: 168							
Identification: 0x2ebf (11967)							
Flags: 0x00							
Fragment offset: 0							
Time to live: 251							
Protocol: ICMP (1)							
Header checksum: 0x03b1 [correct]							
Source: 198.18.240.1 (198.18.240.1)							
Destination: 198.18.15.254 (198.18.15.254)							
[Source GeoIP: Unknown]							
[Destination GeoIP: Unknown]							
Internet Control Message Protocol							

FIGURE 7 – Quand R1 fait un traceroute vers R4, R2 envoie un ICMP TTL-expired avec un TTL de 251.

Or, un ping direct sur 198.18.240.1 nous indique qu'il n'y a pas de routeur intermédiaire entre R1 et R2 :

13784	37884.41134100	198.18.15.254	198.18.240.1	ICMP	114	Echo (ping) request id=0x0007, seq=0/0, ttl=	
13785	37884.41355400	198.18.240.1	198.18.15.254	ICMP	114	Echo (ping) reply id=0x0007, seq=0/0, ttl=	

Frame 13785: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0							
Ethernet II, Src: ca:01:3c:66:00:1c (ca:01:3c:66:00:1c), Dst: ca:00:3c:66:00:1c (ca:00:3c:66:00:1c)							
Internet Protocol Version 4, Src: 198.18.240.1 (198.18.240.1), Dst: 198.18.15.254 (198.18.15.254)							
Version: 4							
Header length: 20 bytes							
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))							
Total Length: 100							
Identification: 0x001f (31)							
Flags: 0x00							
Fragment offset: 0							
Time to live: 255							
Protocol: ICMP (1)							

FIGURE 8 – Quand R1 fait ping vers 198.18.240.1, on observe un TTL de 255.

Donc le message ICMP ttl-expired a fait le tour de notre tunnel MPLS alors que l'écho reply est revenu directement.



Concernant le traceroute, R2 a reçu un paquet IP avec un TTL de 1. Il effectue la décrémentation avant transfert. Il se rend compte que le TTL est de 0. Il détruit le paquet et émet un message ICMP TTL-expired à destination de R1. Ce nouveau paquet IP a donc un TTL de 255. Le paquet arrive en R3, qui va décrémentation le TTL (254) puis transmettre le paquet à R4 qui va décrémentation le TTL (253) puis transmettre le paquet à R3 qui va décrémentation à nouveau le TTL (252) puis transmettre le paquet à R2 qui va décrémentation le TTL (251) puis transmettre à R1.

En excluant le dernier saut R2-> R1 du message ICMP, on se rend compte qu'il y a 3 routeurs intermédiaires et que 198.18.240.11 est l'Egress d'un tunnel MPLS et que R2 et R3 sont aussi traversés par ce tunnel.

On pourra confirmer cela en analysant les TTL des autres messages ICMP TTL-expired (ceux de R3 et R4).

C'est sur cette différence entre les TTL que se base la détection des tunnels MPLS implicite.

## 9 Question 8

Le nouveau préfixe P5 sera 198.18.111.254.

Théoriquement, R4 devra annoncer la nouvelle loopback via OSPF puis on devrait avoir des échanges LDP.

Vérification : on obtient bien le comportement décrit ci-dessus.

1 0.000000000	198.18.240.11	224.0.0.5	OSPF	134 LS Update
2 2.508559000	198.18.240.10	224.0.0.5	OSPF	78 LS Acknowledge
3 5.252599000	198.18.63.254	198.18.47.254	LDP	109 Address Message Label Mapping Message
4 5.254738000	198.18.47.254	198.18.63.254	TCP	60 ldp > 18503 [ACK] Seq=1 Ack=56 Win=3677 Len=0
5 8.187550000	198.18.47.254	198.18.63.254	LDP	92 Label Mapping Message
6 8.189612000	198.18.63.254	198.18.47.254	TCP	60 18503 > ldp [ACK] Seq=56 Ack=39 Win=3676 Len=0

```

> Frame 1: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
> Ethernet II, Src: ca:03:28:fe:00:1c (ca:03:28:fe:00:1c), Dst: IPv4mcast_00:00:05 (01:00:5e:00:00:05)
> Internet Protocol Version 4, Src: 198.18.240.11 (198.18.240.11), Dst: 224.0.0.5 (224.0.0.5)
> Open Shortest Path First
  > OSPF Header
  > LS Update Packet
    > Number of LSAs: 1
    > LS Type: Router-LSA
      > LS Age: 1 seconds
      > Do Not Age: False
      > Options: 0x22 (DC, E)
      > Link-State Advertisement Type: Router-LSA (1)
      > Link State ID: 198.18.63.254
      > Advertising Router: 198.18.63.254 (198.18.63.254)
      > LS Sequence Number: 0x80000004
      > LS Checksum: 0x19fd
      > Length: 72
      > Flags: 0x00
      > Number of Links: 4
        > Type: Stub ID: 198.18.111.254 Data: 255.255.255.255 Metric: 1
        > Type: Stub ID: 198.18.63.254 Data: 255.255.255.255 Metric: 1
        > Type: Transit ID: 198.18.240.15 Data: 198.18.240.14 Metric: 1
        > Type: Transit ID: 198.18.240.11 Data: 198.18.240.11 Metric: 1

```

FIGURE 9 – R4 annonce sa nouvelle loopback en OSPF.

1	0.000000000	198.18.240.11	224.0.0.5	OSPF	134 LS Update
2	2.508559000	198.18.240.10	224.0.0.5	OSPF	78 LS Acknowledge
3	5.252599000	198.18.63.254	198.18.47.254	LDP	109 Address Message Label Mapping Message
4	5.254738000	198.18.47.254	198.18.63.254	TCP	60 ldp > 18503 [ACK] Seq=1 Ack=56 Win=3677 Len=0
5	8.187550000	198.18.47.254	198.18.63.254	LDP	92 Label Mapping Message
6	8.189612000	198.18.63.254	198.18.47.254	TCP	60 18503 > ldp [ACK] Seq=56 Ack=39 Win=3676 Len=0

```

Frame 3: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface 0
Ethernet II, Src: ca:03:28:fe:00:1c (ca:03:28:fe:00:1c), Dst: ca:02:28:fe:00:38 (ca:02:28:fe:00:38)
Internet Protocol Version 4, Src: 198.18.63.254 (198.18.63.254), Dst: 198.18.47.254 (198.18.47.254)
Transmission Control Protocol, Src Port: 18503 (18503), Dst Port: ldp (646), Seq: 1, Ack: 1, Len: 55
Label Distribution Protocol
  Version: 1
  PDU Length: 51
  LSR ID: 198.18.63.254 (198.18.63.254)
  Label Space ID: 0
  Address Message
    0... .... = U bit: Unknown bit not set
    Message Type: Address Message (0x300)
    Message Length: 14
    Message ID: 0x00000059
  Address List TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Address List TLV (0x101)
    TLV Length: 6
    Address Family: IPv4 (1)
  Addresses
    Address 1: 198.18.111.254
  Label Mapping Message
    0... .... = U bit: Unknown bit not set
    Message Type: Label Mapping Message (0x400)
    Message Length: 23
    Message ID: 0x0000005b
  Forwarding Equivalence Classes TLV
  Generic Label TLV

```

FIGURE 10 – R4 annonce sa nouvelle loopback via LDP.

1	0.000000000	198.18.240.11	224.0.0.5	OSPF	134 LS Update
2	2.508559000	198.18.240.10	224.0.0.5	OSPF	78 LS Acknowledge
3	5.252599000	198.18.63.254	198.18.47.254	LDP	109 Address Message Label Mapping Message
4	5.254738000	198.18.47.254	198.18.63.254	TCP	60 ldp > 18503 [ACK] Seq=1 Ack=56 Win=3677 Len=0
5	8.187550000	198.18.47.254	198.18.63.254	LDP	92 Label Mapping Message
6	8.189612000	198.18.63.254	198.18.47.254	TCP	60 18503 > ldp [ACK] Seq=56 Ack=39 Win=3676 Len=0

```

Frame 5: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
Ethernet II, Src: ca:02:28:fe:00:38 (ca:02:28:fe:00:38), Dst: ca:03:28:fe:00:1c (ca:03:28:fe:00:1c)
Internet Protocol Version 4, Src: 198.18.47.254 (198.18.47.254), Dst: 198.18.63.254 (198.18.63.254)
Transmission Control Protocol, Src Port: ldp (646), Dst Port: 18503 (18503), Seq: 1, Ack: 56, Len: 38
Label Distribution Protocol
  Version: 1
  PDU Length: 34
  LSR ID: 198.18.47.254 (198.18.47.254)
  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: 0x0000008a
  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: 198.18.111.254
  Generic Label TLV
    00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x00)
    TLV Type: Generic Label TLV (0x200)
    TLV Length: 4
    Generic Label: 26

```

FIGURE 11 – Attribution des labels pour la nouvelle loopback : pour transmettre des paquets à destination de la nouvelle loopback, R3 attend qu'ils soient tagués avec le label 26.

Non, l'attribution des labels n'est pas downstream on demand car, dans ce mode-là, un PDU LDP Request descend depuis l'Ingress et une réponse LDP Mapping remonte jusqu'à l'Ingress.

Or, dans notre cas, R4 ne répond pas à une requête mais informe spontanément ses voisins. On est donc dans un mode unsolicited downstream où c'est l'Egress qui initie le mapping.

## 10 Question 9

Théoriquement, des messages OSPF de type LS UPDATE seront émis et l'on constatera la disparition du lien R1<->R2. Puis, des échanges LDP devront aussi indiquer la disparition du voisin.

Vérification : on "éteint" l'interface f1/0 de R1.

86	81.087834000	198.18.240.6	224.0.0.5	OSPF	94 LS Update
87	81.129670000	198.18.240.6	224.0.0.5	OSPF	122 LS Update
<p>&gt; Frame 87: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0</p> <p>&gt; Ethernet II, Src: ca:01:28:fe:00:54 (ca:01:28:fe:00:54), Dst: IPv4mcast_00:00:05 (01:00:5e:00:00:05)</p> <p>&gt; Internet Protocol Version 4, Src: 198.18.240.6 (198.18.240.6), Dst: 224.0.0.5 (224.0.0.5)</p> <p>Open Shortest Path First</p> <p>  OSPF Header</p> <p>    LS Update Packet</p> <p>      Number of LSAs: 1</p> <p>      LS Type: Router-LSA</p> <p>        LS Age: 1 seconds</p> <p>        Do Not Age: False</p> <p>        Options: 0x22 (DC, E)</p> <p>        Link-State Advertisement Type: Router-LSA (1)</p> <p>        Link State ID: 198.18.31.254</p> <p>        Advertising Router: 198.18.31.254 (198.18.31.254)</p> <p>        LS Sequence Number: 0x80000005</p> <p>        LS Checksum: 0xd615</p> <p>        Length: 60</p> <p>        Flags: 0x00</p> <p>        Number of Links: 3</p> <p>        Type: Stub ID: 198.18.31.254 Data: 255.255.255.255 Metric: 1</p> <p>        Type: Transit ID: 198.18.240.7 Data: 198.18.240.6 Metric: 1</p> <p>        Type: Transit ID: 198.18.240.5 Data: 198.18.240.4 Metric: 1</p>					

FIGURE 12 – R2 émet un LS UPDATE dans lequel il n'annonce plus avoir un lien avec R1 (absence de 198.18.240.0 et de 198.18.15.254).

134	121.104868000	198.18.31.254	198.18.47.254	LDP	92 Label Withdrawal Message
135	121.140420000	198.18.47.254	198.18.31.254	TCP	60 17854 > ldp [ACK] Seq=37 Ack=103 Win=3810 Len=0
136	121.142504000	198.18.47.254	198.18.31.254	LDP	92 Label Release Message
137	121.146626000	198.18.31.254	198.18.47.254	TCP	60 ldp > 17854 [ACK] Seq=103 Ack=75 Win=3856 Len=0
138	121.181669000	198.18.31.254	198.18.47.254	LDP	92 Label Mapping Message
139	121.183805000	198.18.47.254	198.18.31.254	TCP	60 17854 > ldp [ACK] Seq=75 Ack=141 Win=3772 Len=0
<p>Frame 134: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0</p> <p>Ethernet II, Src: ca:01:28:fe:00:54 (ca:01:28:fe:00:54), Dst: ca:02:28:fe:00:1c (ca:02:28:fe:00:1c)</p> <p>Internet Protocol Version 4, Src: 198.18.31.254 (198.18.31.254), Dst: 198.18.47.254 (198.18.47.254)</p> <p>Transmission Control Protocol, Src Port: ldp (646), Dst Port: 17854 (17854), Seq: 65, Ack: 37, Len: 38</p> <p>Label Distribution Protocol</p> <p>  Version: 1</p> <p>  PDU Length: 34</p> <p>  LSR ID: 198.18.31.254 (198.18.31.254)</p> <p>  Label Space ID: 0</p> <p>  Label Withdrawal Message</p> <p>    0... .. = U bit: Unknown bit not set</p> <p>    Message Type: Label Withdrawal Message (0x402)</p> <p>    Message Length: 24</p> <p>    Message ID: 0x000000cf</p> <p>  Forwarding Equivalence Classes TLV</p> <p>    00... .. = TLV Unknown bits: Known TLV, do not Forward (0x00)</p> <p>    TLV Type: Forwarding Equivalence Classes TLV (0x100)</p> <p>    TLV Length: 8</p> <p>  FEC Elements</p> <p>    FEC Element 1</p> <p>      FEC Element Type: Prefix FEC (2)</p> <p>      FEC Element Address Type: IPv4 (1)</p> <p>      FEC Element Length: 31</p> <p>      Prefix: 198.18.240.0</p> <p>  Generic Label TLV</p>					

FIGURE 13 – R2 annonce à R3 la disparition de R1 (198.18.240.0) avec un message LDP Withdrawal. R3 annonce la disparition d'un routeur en amont (R1) avec un message LDP Release

Dans le même temps, R1 annoncera aussi, à R5, en OSPF et via LDP, sa perte de connectivité avec R2.

On observe le même comportement (LS UPDATE puis LDP Withdrawal puis LDP Release) sur R5. Sur les autres routeurs, on aura LS UPDATE et LDP Release.

## 11 Question 10

Voici le schéma de notre maquette avec la capacité réservable pour chaque lien.

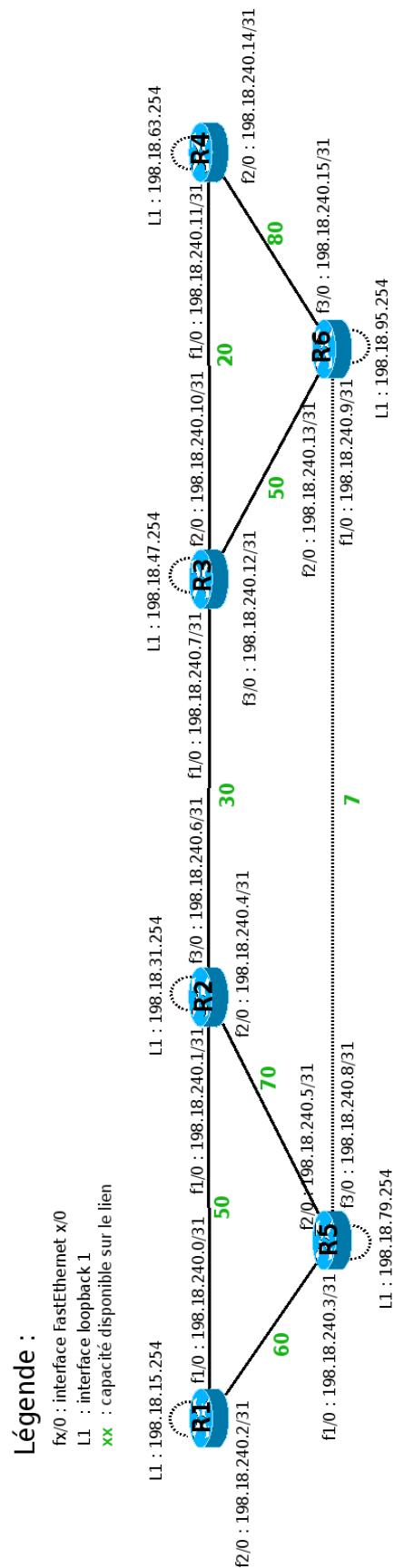


FIGURE 14 – Schéma de notre maquette avec la capacité réservable sur chaque lien. D’après l’énoncé.

Notes :

- Sur chaque lien, nous n'avons pas l'égalité « capacité réservable = capacité réelle » car nous avons laissé de la place pour le trafic best-effort comme cela est recommandé.
- Dans un premier temps, nous avons un seul niveau de priorité.

Pour mettre en œuvre l'ingénierie de trafic sur notre maquette, il faut activer les tunnels TE et OSPF-TE puis indiquer, pour chaque lien, la capacité réservable. Sur les routeurs Cisco, cela se fait avec les commandes suivantes (ici, sur R1) :

```
!Tunnels TE
mpls traffic-eng

!OSPF-TE
router ospf 1
mpls traffic-eng router-id Loopback1
mpls traffic-eng area 0

!Pour chaque lien, la capacité réservable
interface FastEthernet1/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 50000 50000
```

Les configurations intégrales de nos routeurs se trouvent en annexe 3.

On remarque que, dès l'activation d'OSPF-TE sur R1, celui-ci annonce de nouveaux LSA, des LSA-TE, qui se propagent dans toute notre maquette. Ici, sur R4 :

1	0.000000	198.18.240.15	224.0.0.5	OSPF	194 LS Update
2	466.567509	198.18.240.15	224.0.0.5	OSPF	194 LS Update
▸ Frame 1: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits)					
▸ Ethernet II, Src: ca:05:28:fe:00:54 (ca:05:28:fe:00:54), Dst: IPv4mcast_00:00:05 (01:00:5e:00:00:05)					
▸ Internet Protocol Version 4, Src: 198.18.240.15 (198.18.240.15), Dst: 224.0.0.5 (224.0.0.5)					
▾ Open Shortest Path First					
▸ OSPF Header					
▾ LS Update Packet					
Number of LSAs: 1					
▾ LS Type: Opaque LSA, Area-local scope					
LS Age: 3 seconds					
Do Not Age: False					
▸ Options: 0x20 (DC)					
Link-State Advertisement Type: Opaque LSA, Area-local scope (10)					
Link State ID Opaque Type: Traffic Engineering LSA (1)					
Link State ID TE-LSA Reserved: 0					
Link State ID TE-LSA Instance: 0					
Advertising Router: 198.18.15.254 (198.18.15.254)					
LS Sequence Number: 0x80000001					
LS Checksum: 0x7836					
Length: 132					
▾ MPLS Traffic Engineering LSA					
▸ Router Address: 198.18.15.254					
▾ Link Information					
TLV Type: 2 - Link Information					
TLV Length: 100					
▸ Link Type: 2 - Multi-access					
▸ Link ID: 198.18.240.0					
▸ Local Interface IP Address					
▸ Traffic Engineering Metric: 1					
▸ Maximum Bandwidth: 12500000 bytes/s (100000000 bits/s)					
▸ Maximum Reservable Bandwidth: 0 bytes/s (0 bits/s)					
▸ Unreserved Bandwidth					
▸ Resource Class/Color: 0x00000000					
▸ Unknown Link sub-TLV: 32770					

FIGURE 15 – R1 annonce la capacité du lien R1<->R2 mais indique une capacité réservable nulle.

Puis, quand on configure la capacité réservable sur le lien R1<->R2, R1 l'annonce dans ses LSA-TE. Toujours sur R4 :

1	0.000000	198.18.240.15	224.0.0.5	OSPF	194 LS Update
2	466.567509	198.18.240.15	224.0.0.5	OSPF	194 LS Update
▸ Frame 2: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits)					
▸ Ethernet II, Src: ca:05:28:fe:00:54 (ca:05:28:fe:00:54), Dst: IPv4mcast_00:00:05 (01:00:5e:00:00:05)					
▸ Internet Protocol Version 4, Src: 198.18.240.15 (198.18.240.15), Dst: 224.0.0.5 (224.0.0.5)					
▾ Open Shortest Path First					
▸ OSPF Header					
▾ LS Update Packet					
Number of LSAs: 1					
▾ LS Type: Opaque LSA, Area-local scope					
LS Age: 3 seconds					
Do Not Age: False					
▸ Options: 0x20 (DC)					
Link-State Advertisement Type: Opaque LSA, Area-local scope (10)					
Link State ID Opaque Type: Traffic Engineering LSA (1)					
Link State ID TE-LSA Reserved: 0					
Link State ID TE-LSA Instance: 0					
Advertising Router: 198.18.15.254 (198.18.15.254)					
LS Sequence Number: 0x80000003					
LS Checksum: 0xbdbb					
Length: 132					
▾ MPLS Traffic Engineering LSA					
▸ Router Address: 198.18.15.254					
▾ Link Information					
TLV Type: 2 - Link Information					
TLV Length: 100					
▸ Link Type: 2 - Multi-access					
▸ Link ID: 198.18.240.0					
▸ Local Interface IP Address					
▸ Traffic Engineering Metric: 1					
▸ Maximum Bandwidth: 12500000 bytes/s (100000000 bits/s)					
▸ Maximum Reservable Bandwidth: 6250000 bytes/s (50000000 bits/s)					
▸ Unreserved Bandwidth					
▸ Resource Class/Color: 0x00000000					
▸ Unknown Link sub-TLV: 32770					

FIGURE 16 – Maintenant, R1 annonce aussi la capacité réservable sur le lien R1<->R2.

Nous pouvons maintenant construire nos tunnels. Nous avons choisi de construire 3 tunnels :

1. T0, de R1 jusqu'à P4 : 30. Ce tunnel nous permettra de constater que ce n'est pas toujours le chemin le plus court qui est sélectionné mais celui qui répond d'abord à la contrainte de capacité.
2. T1, de R1 jusqu'à P3 : 3. Ce tunnel nous permettra de constater ce qui arrive quand le lien R2<->R3 n'a plus de capacité réservable.
3. T2, de R1 jusqu'à P6 : 15. Ce tunnel nous permettra de constater ce qui arrive quand une contrainte ne peut pas être satisfaite.

Créons T0 :

```
interface Tunnel0
ip unnumbered Loopback1
tunnel destination 198.18.63.254
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng bandwidth 30000
tunnel mpls traffic-eng path-option 1 dynamic
no routing dynamic
```

Nous constatons qu'un PDU RSVP-TE de type PATH part de R1 jusqu'à R4 en passant par R2, R3 et R6. Illustrations sur R1 et R2 :

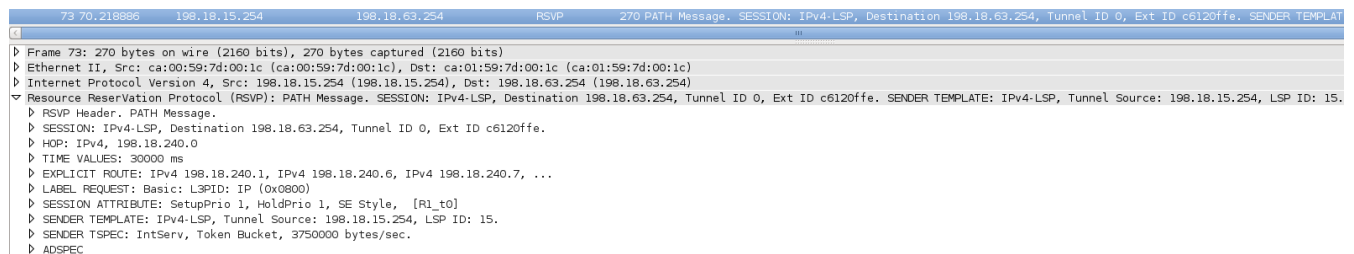


FIGURE 17 – R1 fait sa demande de tunnel avec un PDU RSVP-TE de type PATH.

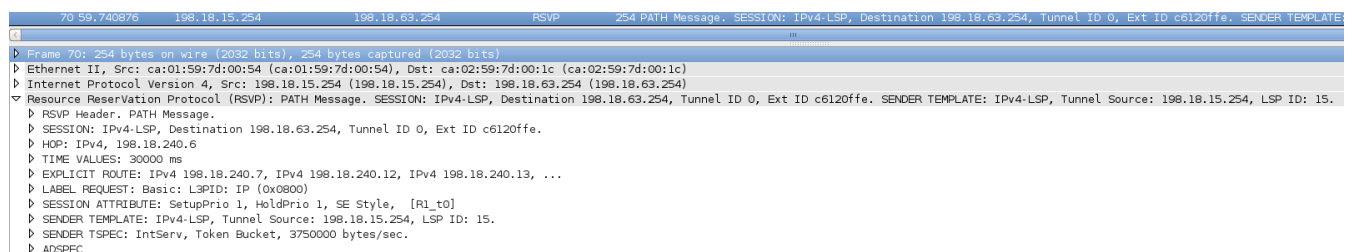


FIGURE 18 – R2 transmet simplement la demande de tunnel de R1.

R4 est en mesure de satisfaire la demande et il répond donc à R1 avec un PDU RSVP-TE de type RESV. Cette fois-ci, le message est adressé au prochain nœud du chemin : R6.



44 37.474521	198.18.240.14	198.18.240.15	RSVP	142 RESV Message. SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe. FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.
<pre> Frame 44: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0 Ethernet II, Src: ca:03:59:7d:00:38 (ca:03:59:7d:00:38), Dst: ca:05:59:7d:00:54 (ca:05:59:7d:00:54) Internet Protocol Version 4, Src: 198.18.240.14 (198.18.240.14), Dst: 198.18.240.15 (198.18.240.15) Resource Reservation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe. FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.   RSVP Header. RESV Message.     SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe.     HOP: IPv4, 198.18.240.14     TIME VALUES: 30000 ms     STYLE: Shared-Explicit (18)     FLOWSPEC: Controlled Load: Token Bucket, 3750000 bytes/sec.     FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.     LABEL: 0 </pre>				

FIGURE 19 – R4 émet un PDU RSVP-TE de type RESV à destination de R6.

R6 peut satisfaire la demande et émet donc un nouveau PDU RSVP-TE RESV à R3.

63 54.109188	198.18.240.13	198.18.240.12	RSVP	142 RESV Message. SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe. FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.
<pre> Frame 63: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0 Ethernet II, Src: ca:03:59:7d:00:38 (ca:03:59:7d:00:38), Dst: ca:02:59:7d:00:54 (ca:02:59:7d:00:54) Internet Protocol Version 4, Src: 198.18.240.13 (198.18.240.13), Dst: 198.18.240.12 (198.18.240.12) Resource Reservation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe. FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.   RSVP Header. RESV Message.     SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe.     HOP: IPv4, 198.18.240.13     TIME VALUES: 30000 ms     STYLE: Shared-Explicit (18)     FLOWSPEC: Controlled Load: Token Bucket, 3750000 bytes/sec.     FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.     LABEL: 16 </pre>				

FIGURE 20 – R6 émet un PDU RSVP-TE de type RESV à destination de R3.

Cela continue jusqu'à R1 :

75 70.306942	198.18.240.1	198.18.240.0	RSVP	142 RESV Message. SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe. FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.
<pre> Frame 75: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0 Ethernet II, Src: ca:01:59:7d:00:1c (ca:01:59:7d:00:1c), Dst: ca:00:59:7d:00:1c (ca:00:59:7d:00:1c) Internet Protocol Version 4, Src: 198.18.240.1 (198.18.240.1), Dst: 198.18.240.0 (198.18.240.0) Resource Reservation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe. FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.   RSVP Header. RESV Message.     SESSION: IPv4-LSP, Destination 198.18.63.254, Tunnel ID 0, Ext ID c6120ffe.     HOP: IPv4, 198.18.240.1     TIME VALUES: 30000 ms     STYLE: Shared-Explicit (18)     FLOWSPEC: Controlled Load: Token Bucket, 3750000 bytes/sec.     FILTERSPEC: IPv4-LSP, Tunnel Source: 198.18.15.254, LSP ID: 15.     LABEL: 18 </pre>				

FIGURE 21 – R1 reçoit le PDU RSVP-TE RESV de R2 et sait donc que son tunnel TE est prêt.

En parallèle, OSPF propage les nouvelles métriques TE pour tenir compte des changements concernant la capacité réservable sur les liens traversés. Exemple : R2 annonce qu'on ne peut plus réserver de capacité sur le lien R2<->R3 sauf si l'on est plus prioritaire que le précédent demandeur (comme R1 était en priorité 1, il faut être en priorité 0) :

77 70.323665	198.18.240.1	224.0.0.5	OSPF	310 LS Update
<pre> Link ID: 198.18.240.7 Local Interface IP Address Traffic Engineering Metric: 1 Maximum Bandwidth: 12500000 bytes/s (100000000 bits/s) Maximum Reservable Bandwidth: 3750000 bytes/s (30000000 bits/s) Unreserved Bandwidth   TLV Type: 8: Unreserved Bandwidth   TLV Length: 32   Pri (or TE-Class) 0: 3750000 bytes/s (30000000 bits/s)   Pri (or TE-Class) 1: 0 bytes/s (0 bits/s)   Pri (or TE-Class) 2: 0 bytes/s (0 bits/s)   Pri (or TE-Class) 3: 0 bytes/s (0 bits/s)   Pri (or TE-Class) 4: 0 bytes/s (0 bits/s)   Pri (or TE-Class) 5: 0 bytes/s (0 bits/s)   Pri (or TE-Class) 6: 0 bytes/s (0 bits/s)   Pri (or TE-Class) 7: 0 bytes/s (0 bits/s) </pre>				

FIGURE 22 – R2 annonce, en OSPF, que plus aucune capacité n'est réservable sur le lien R2<->R3.

On vérifie la bonne création du tunnel sur R1 :

```
R1#show mpls traffic-eng tunnels Tunnel 0
Name: R1_t0                               (Tunnel0) Destination: 198.18.63.254
Status:
  Admin: up          Oper: up          Path: valid          Signalling: connected

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

InLabel   : -
OutLabel  : FastEthernet1/0, 18
RSVP Signalling Info:
  Src 198.18.15.254, Dst 198.18.63.254, Tun_Id 0, Tun_Instance 86
RSVP Path Info:
  My Address: 198.18.240.0
  Explicit Route: 198.18.240.1 198.18.240.6 198.18.240.7 198.18.240.12
                  198.18.240.13 198.18.240.15 198.18.240.14 198.18.63.254
  Record Route:  NONE
  Tspec: ave rate=30000 kbits, burst=1000 bytes, peak rate=30000 kbits
RSVP Resv Info:
  Record Route:  NONE
  Fspec: ave rate=30000 kbits, burst=1000 bytes, peak rate=30000 kbits
Shortest Unconstrained Path Info:
  Path Weight: 3 (TE)
  Explicit Route: 198.18.240.0 198.18.240.1 198.18.240.6 198.18.240.7
                  198.18.240.10 198.18.240.11 198.18.63.254
```

```
R1#traceroute 198.18.63.254 source 198.18.15.254
 1 198.18.240.1 [MPLS: Label 16 Exp 0] 20 msec 40 msec 40 msec
 2 198.18.240.7 [MPLS: Label 27 Exp 0] 80 msec 12 msec 12 msec
 3 198.18.240.13 [MPLS: Label 27 Exp 0] 80 msec 44 msec 40 msec
 4 198.18.240.14 44 msec * 56 msec
```

On constate que le tunnel est en cours de fonctionnement et que, comme prévu, le chemin suivi n'est pas le plus court mais celui qui respecte la contrainte de capacité : R1<->R2<->R3<->R6<->R4.

Créons notre deuxième tunnel (T1) :

```
interface Tunnel1
ip unnumbered Loopback1
tunnel destination 198.18.47.254
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng bandwidth 4000
tunnel mpls traffic-eng path-option 1 dynamic
no routing dynamic
```

Nous passons le processus et nous regardons directement le résultat sur R1 :

```
R1#show mpls traffic-eng tunnels Tunnel 1
```

```
Name: R1_t1                               (Tunnel1) Destination: 198.18.47.254
```

```
Status:
```

```
Admin: up          Oper: up          Path: valid          Signalling: connected
```

```
path option 1, type dynamic (Basis for Setup, path weight 12)
```

```
InLabel   : -
```

```
OutLabel  : FastEthernet2/0, 16
```

```
RSVP Signalling Info:
```

```
Src 198.18.15.254, Dst 198.18.47.254, Tun_Id 1, Tun_Instance 12
```

```
RSVP Path Info:
```

```
My Address: 198.18.240.2
```

```
Explicit Route: 198.18.240.3 198.18.240.8 198.18.240.9 198.18.240.13
                198.18.240.12 198.18.47.254
```

```
Record Route: NONE
```

```
Tspec: ave rate=4000 kbits, burst=1000 bytes, peak rate=4000 kbits
```

```
RSVP Resv Info:
```

```
Record Route: NONE
```

```
Fspec: ave rate=4000 kbits, burst=1000 bytes, peak rate=4000 kbits
```

```
Shortest Unconstrained Path Info:
```

```
Path Weight: 2 (TE)
```

```
Explicit Route: 198.18.240.0 198.18.240.1 198.18.240.6 198.18.240.7
                198.18.47.254
```

```
R1#traceroute 198.18.47.254 source 198.18.15.254
```

```
 1 198.18.240.3 [MPLS: Label 16 Exp 0] 44 msec 8 msec 8 msec
 2 198.18.240.9 [MPLS: Label 28 Exp 0] 8 msec 12 msec 12 msec
 3 198.18.240.12 8 msec * 8 msec
```

On constate que le tunnel est fonctionnel. Comme il n'y a plus de capacité disponible sur le lien R2<->3 et que l'on n'est pas plus prioritaire que le créateur du tunnel précédent, on passe par le lien R5<->R6 qui, là encore, n'est pas le plus court chemin mais celui qui satisfait notre contrainte de capacité.

Créons notre dernier tunnel :

```
interface Tunnel2
ip unnumbered Loopback1
tunnel destination 198.18.95.254
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng bandwidth 15000
tunnel mpls traffic-eng path-option 1 dynamic
no routing dynamic
```

Regardons le résultat sur R1 :

```
R1#show mpls traffic-eng tunnels Tunnel 2
```

```
Name: R1_t2 (Tunnel2) Destination: 198.18.95.254
```

Status:

```
Admin: up      Oper: down  Path: not valid  Signalling: Down
path option 1, type dynamic
```

Config Parameters:

```
Bandwidth: 15000 kbps (Global) Priority: 1 1 Affinity: 0x0/0xFFFF
Metric Type: TE (default)
AutoRoute: enabled LockDown: disabled Loadshare: 15000 bw-based
auto-bw: disabled
```

Shortest Unconstrained Path Info:

```
Path Weight: 3 (TE)
Explicit Route: 198.18.240.0 198.18.240.1 198.18.240.6 198.18.240.7
                198.18.240.12 198.18.240.13 198.18.95.254
Last Error: PCALC:: No path to destination, 198.18.95.254
```

Le tunnel n'est pas fonctionnel car la contrainte TE n'a pas pu être satisfaite : le lien R2<->R3 est "plein" et nous ne sommes pas plus prioritaires. De plus, la capacité que nous demandons ne peut pas être réservée sur le lien R5<->R6.

Notons qu'au niveau réseau, aucun paquet RSVP-TE n'a été émis. C'est tout à fait logique : en effectuant l'algorithme CSPF, R1 s'est rendu compte qu'aucun chemin respectant la contrainte n'était disponible.

Comme dernier exercice, augmentons la priorité de notre dernier tunnel :

```
interface Tunnel2
tunnel mpls traffic-eng priority 0 0
shutdown
no shutdown
```

Cette fois-ci, notre demande a pu être satisfaite :

```
R1#show mpls traffic-eng tunnels Tunnel 2
Name: R1_t2                                (Tunnel2) Destination: 198.18.95.254
Status:
  Admin: up          Oper: up          Path: valid          Signalling: connected

  path option 1, type dynamic (Basis for Setup, path weight 3)

Config Parameters:
  Bandwidth: 15000    kbps (Global) Priority: 0 0  Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled  LockDown: disabled Loadshare: 15000    bw-based
  auto-bw: disabled

InLabel  : -
OutLabel : FastEthernet1/0, 16
RSVP Signalling Info:
  Src 198.18.15.254, Dst 198.18.95.254, Tun_Id 2, Tun_Instance 40
RSVP Path Info:
  My Address: 198.18.240.0
  Explicit Route: 198.18.240.1 198.18.240.6 198.18.240.7 198.18.240.12
                  198.18.240.13 198.18.95.254
  Record Route: NONE
```

Tspec: ave rate=15000 kbits, burst=1000 bytes, peak rate=15000 kbits

RSVP Resv Info:

Record Route: NONE

Fspec: ave rate=15000 kbits, burst=1000 bytes, peak rate=15000 kbits

Shortest Unconstrained Path Info:

Path Weight: 3 (TE)

Explicit Route: 198.18.240.0 198.18.240.1 198.18.240.6 198.18.240.7  
198.18.240.12 198.18.240.13 198.18.95.254

Quelques instants plus tard, notre premier tunnel tombe : réception d'un PDU RSVP-TE PATH ERROR : notre demande n'étant plus la plus prioritaire et en l'absence de liens alternatifs disposant d'une capacité réservable suffisante, il n'est plus possible de réserver 30 mégas sur le lien R2<->R3.

Illustration :

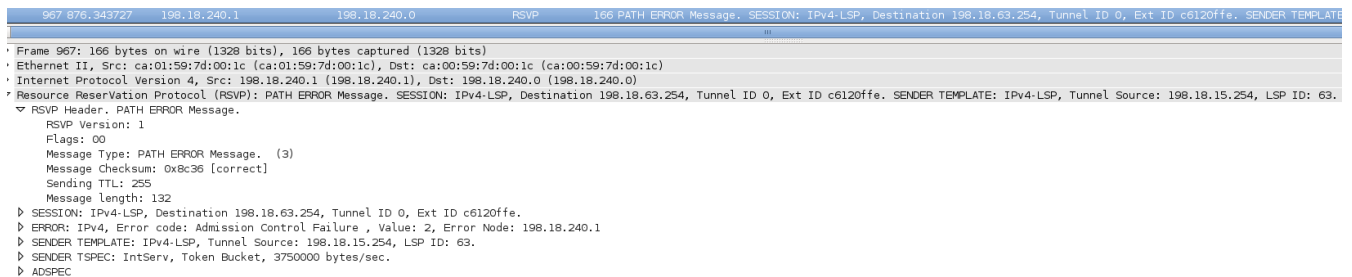


FIGURE 23 – R1 reçoit un PDU RSVP-TE PATH ERROR pour notre premier tunnel dont la contrainte ne peut plus être satisfaite.

Vérifions sur R1 :

R1#show mpls traffic-eng tunnels Tunnel 0

Name: R1\_t0 (Tunnel0) Destination: 198.18.63.254

Status:

Admin: up Oper: down Path: not valid Signalling: Down

path option 1, type dynamic

Config Parameters:

Bandwidth: 30000 kbps (Global) Priority: 1 1 Affinity: 0x0/0xFFFF

Shortest Unconstrained Path Info:

Path Weight: 3 (TE)

Explicit Route: 198.18.240.0 198.18.240.1 198.18.240.6 198.18.240.7  
198.18.240.10 198.18.240.11 198.18.63.254

Prior LSP:

ID: path option 1 [63]

Removal Trigger: path error

Dès que l'on démonte/"éteint" ce dernier tunnel, le premier tunnel est remonté automatiquement.

Pour terminer, nous voulions faire des demandes de réservation de tunnels depuis plusieurs routeurs, pas seulement depuis R1. Après avoir supprimé le dernier tunnel de R1, nous avons tenté de le faire depuis R5 et avons obtenu le même résultat : sans priorité, on ne passe pas. Avec une priorité plus forte, on passe et le premier tunnel sur R1 tombe.

## 12 Bibliographie

Dans l'ordre d'utilisation.

- How to configure OSPF cost - <https://supportforums.cisco.com/docs/DOC-5349>
- Multiprotocol Label Switching on Cisco Routers - <http://www.wolf-lab.com/webAD/webedit/uploadfile/200793165456611.PDF>
- Revealing MPLS Tunnels Obscured from Traceroute - [http://www.caida.org/publications/papers/2012/revealing\\_mpls\\_tunnels/revealing\\_mpls\\_tunnels.pdf](http://www.caida.org/publications/papers/2012/revealing_mpls_tunnels/revealing_mpls_tunnels.pdf)
- Course project for AANSW Revealing MPLS Tunnels Obscured from Traceroute - <http://www.slideshare.net/edge7/slides-mpls-tunnel>
- MPLS Traffic Engineering - [http://www.cisco.com/en/US/docs/ios/12\\_0s/feature/guide/TE\\_1208S.html#wp36136](http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/TE_1208S.html#wp36136)



## A Annexes

### A.1 Fichier de configuration Dynagen pour tout le TP

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

  [[ROUTER R1]]
    model = 7200
    f1/0 = R2 f1/0
    f2/0 = R5 f1/0

  [[ROUTER R2]]
    model = 7200
    f2/0 = R5 f2/0
    f3/0 = R3 f1/0

  [[ROUTER R3]]
    model = 7200
    f2/0 = R4 f1/0
    f3/0 = R6 f2/0

  [[ROUTER R4]]
    model = 7200
    f2/0 = R6 f3/0

  [[ROUTER R5]]
    model = 7200
    f3/0 = R6 f1/0

  [[ROUTER R6]]
    model = 7200
```

## A.2 Commandes IOS pour tout le TP

### A.2.1 R1

```
enable
conf t
hostname R1
no ip domain lookup

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 loopback 1
ip address 198.18.15.254 255.255.240.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp
```

## A.2.2 R2

```
enable

conf t
hostname R2
no ip domain lookup

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 198.18.240.4 255.255.255.254
mpls ip
no shutdown
exit

int fastEthernet 3/0
ip address 198.18.240.6 255.255.255.254
mpls ip
no shutdown
exit

int loopback 1
ip address 198.18.31.254 255.255.240.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp
```

### A.2.3 R3

```
enable
conf t
hostname R3
no ip domain lookup

int fastEthernet 1/0
ip address 198.18.240.7 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 2/0
ip address 198.18.240.10 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 3/0
ip address 198.18.240.12 255.255.255.254
mpls ip
no shutdown
exit
int loopback 1
ip address 198.18.47.254 255.255.240.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp
```

#### A.2.4 R4

```
enable
conf t
hostname R4
no ip domain lookup

int fastEthernet 1/0
ip address 198.18.240.11 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 2/0
ip address 198.18.240.14 255.255.255.254
mpls ip
no shutdown
exit
int loopback 1
ip address 198.18.63.254 255.255.240.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp
```

### A.2.5 R5

```
enable
conf t
hostname R5
no ip domain lookup

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 198.18.240.5 255.255.255.254
mpls ip
no shutdown
exit
int fastEthernet 3/0
bandwidth 10000 !restreindre le lien R5-R6 à 10 mégas
ip address 198.18.240.8 255.255.255.254
mpls ip
no shutdown
exit
int loopback 1
ip address 198.18.79.254 255.255.240.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp
```

## A.2.6 R6

```
enable
conf t
hostname R6
no ip domain lookup

int fastEthernet 1/0
ip address 198.18.240.9 255.255.255.254
bandwidth 10000 !restreindre le lien R5-R6 à 10 mégas
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 198.18.240.15 255.255.255.254
mpls ip
no shutdown
exit

int loopback 1
ip address 198.18.95.254 255.255.240.0
no shutdown
exit

router ospf 1
network 198.18.0.0 0.0.255.255 area 0
exit

ip cef
mpls label protocol ldp
```

## A.3 Commandes IOS supplémentaires spécifiques à la question 10

### A.3.1 R1

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

interface FastEthernet1/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 50000 50000

interface FastEthernet2/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 60000 60000
exit
exit
```



### A.3.2 R2

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

interface FastEthernet1/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 50000 50000

interface FastEthernet2/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 70000 70000

interface FastEthernet3/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 30000 30000
exit
exit
```

### A.3.3 R3

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

interface FastEthernet1/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 30000 30000

interface FastEthernet2/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 20000 20000

interface FastEthernet3/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 50000 50000
exit
exit
```

#### A.3.4 R4

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

interface FastEthernet1/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 20000 20000

interface FastEthernet2/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 80000 80000
exit
exit
```

### A.3.5 R5

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

interface FastEthernet1/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 60000 60000

interface FastEthernet2/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 70000 70000

interface FastEthernet3/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 7000 7000
exit
exit
```

### A.3.6 R6

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

interface FastEthernet1/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 7000 7000

interface FastEthernet2/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 50000 50000

interface FastEthernet3/0
mpls ip
mpls traffic-eng tunnels
ip rsvp bandwidth 80000 80000
exit
exit
```

## Table des figures

1	Schéma de notre maquette. D'après l'énoncé. . . . .	4
2	Capture réseau. Ping P4 (198.18.63.254) depuis P1 (198.18.15.254). Aller. . . . .	8
3	Capture réseau. Ping P4 (198.18.63.254) depuis P1 (198.18.15.254). Retour. . . . .	8
4	On observe que R4 reçoit bien les paquets ICMP expired . . . . .	12
5	Les différents types de tunnels MPLS en fonction des fonctions supportées/activées. Source : <a href="http://www.slideshare.net/edge7/slides-mpls-tunnel">http://www.slideshare.net/edge7/slides-mpls-tunnel</a> . . . . .	14
6	Différence de chemin A/R entre un echo request/reply et un ttl-expired. Source : <a href="http://www.caida.org/publications/papers/2012/revealing_mpls_tunnels/revealing_mpls_tunnels.pdf">http://www.caida.org/publications/papers/2012/revealing_mpls_tunnels/revealing_mpls_tunnels.pdf</a> . . . . .	15
7	Quand R1 fait un traceroute vers R4, R2 envoie un ICMP TTL-expired avec un TTL de 251. . . . .	16
8	Quand R1 fait ping vers 198.18.240.1, on observe un TTL de 255. . . . .	16
9	R4 annonce sa nouvelle loopback en OSPF. . . . .	17
10	R4 annonce sa nouvelle loopback via LDP. . . . .	18
11	Attribution des labels pour la nouvelle loopback : pour transmettre des paquets à destination de la nouvelle loopback, R3 attend qu'ils soient tagués avec le label 26. . . . .	18
12	R2 émet un LS UPDATE dans lequel il n'annonce plus avoir un lien avec R1 (absence de 198.18.240.0 et de 198.18.15.254). . . . .	19
13	R2 annonce à R3 la disparition de R1 (198.18.240.0) avec un message LDP Withdrawal. R3 annonce la disparition d'un routeur en amont (R1) avec un message LDP Release .	19
14	Schéma de notre maquette avec la capacité réservable sur chaque lien. D'après l'énoncé.	21
15	R1 annonce la capacité du lien R1<->R2 mais indique une capacité réservable nulle. .	23
16	Maintenant, R1 annonce aussi la capacité réservable sur le lien R1<->R2. . . . .	23
17	R1 fait sa demande de tunnel avec un PDU RSVP-TE de type PATH. . . . .	24
18	R2 transmet simplement la demande de tunnel de R1. . . . .	24
19	R4 émet un PDU RSVP-TE de type RESV à destination de R6. . . . .	25
20	R6 émet un PDU RSVP-TE de type RESV à destination de R3. . . . .	25
21	R1 reçoit le PDU RSVP-TE RESV de R2 et sait donc que son tunnel TE est prêt. . .	25
22	R2 annonce, en OSPF, que plus aucune capacité n'est réservable sur le lien R2<->R3.	25
23	R1 reçoit un PDU RSVP-TE PATH ERROR pour notre premier tunnel dont la contrainte ne peut plus être satisfaite. . . . .	30

# Table des matières

<b>Sommaire</b>	<b>2</b>
<b>1 Avant de commencer</b>	<b>3</b>
1.1 Adressage . . . . .	3
1.1.1 IPv4 . . . . .	3
1.2 Schéma . . . . .	3
<b>2 Question 1</b>	<b>5</b>
<b>3 Question 2</b>	<b>6</b>
<b>4 Question 3</b>	<b>9</b>
<b>5 Question 4</b>	<b>10</b>
<b>6 Question 5</b>	<b>10</b>
<b>7 Question 6</b>	<b>11</b>
<b>8 Question 7</b>	<b>14</b>
<b>9 Question 8</b>	<b>17</b>
<b>10 Question 9</b>	<b>19</b>
<b>11 Question 10</b>	<b>21</b>
<b>12 Bibliographie</b>	<b>32</b>
<b>A Annexes</b>	<b>33</b>
A.1 Fichier de configuration Dynagen pour tout le TP . . . . .	33
A.2 Commandes IOS pour tout le TP . . . . .	34
A.2.1 R1 . . . . .	34
A.2.2 R2 . . . . .	35
A.2.3 R3 . . . . .	36
A.2.4 R4 . . . . .	37
A.2.5 R5 . . . . .	38
A.2.6 R6 . . . . .	39
A.3 Commandes IOS supplémentaires spécifiques à la question 10 . . . . .	40
A.3.1 R1 . . . . .	40

A.3.2	R2 . . . . .	41
A.3.3	R3 . . . . .	42
A.3.4	R4 . . . . .	43
A.3.5	R5 . . . . .	44
A.3.6	R6 . . . . .	45
<b>Table des figures</b>		<b>46</b>
<b>Table des matières</b>		<b>47</b>