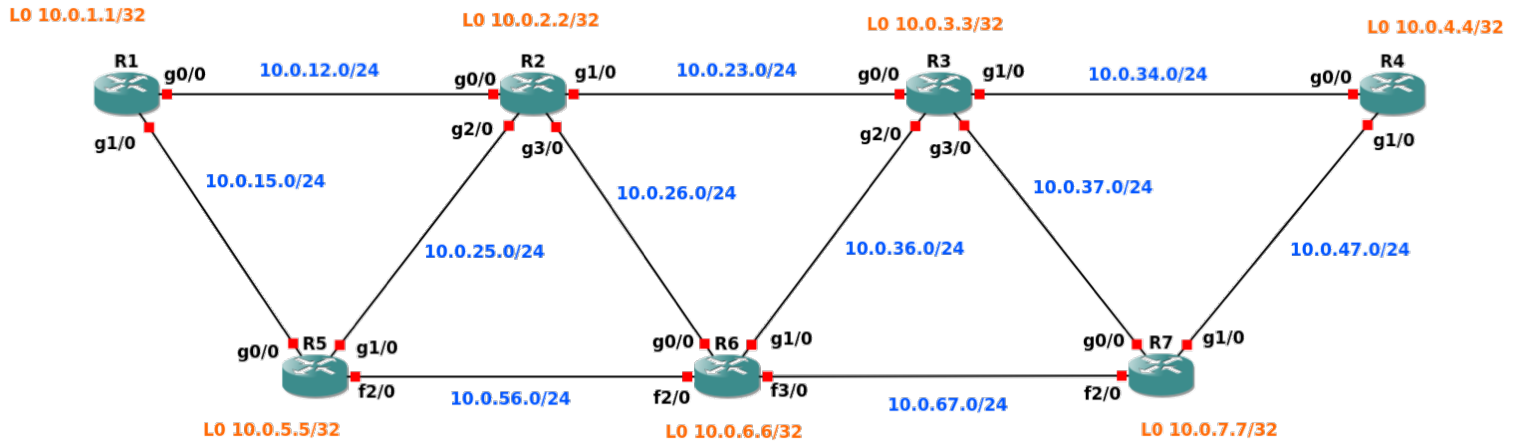


## TP2 – MPLS-TE routing

### ➤ Topologie



Topologie de la configuration de base

Le fichier `tp2_mpls.gns3` décrit cette topologie.

Les routeurs ont pu être redémarrés entre les exercices, la valeur des labels peut donc être différente.

### ➤ Configuration de base

**Q1.** Les fichiers de configuration des routeurs `Rx.cfg` sont joints à l'archive.

L'adressage IP est tel que le sous-réseau entre  $R_x$  et  $R_y$  (avec  $x < y$ ) est  $10.0.xy.0/24$ . L'adresse IP associée à l'interface de  $R_x$  est  $10.0.xy.x$ , de même que celle associée à l'interface de  $R_y$  est  $10.0.xy.y$ . Chaque routeur possède une adresse de loopback sur l'interface `lo0` de type  $10.0.x.x/32$  pour  $R_x$  par exemple.

Les liens  $R_5-R_6$  et  $R_6-R_7$  sont en FastEthernet (100 Mbps) tandis que les autres liens sont en GigabitEthernet (1 000 Mbps).

Par exemple, pour  $R_5$ , la configuration est comme suit :

```
R5(config)#int g0/0
R5(config-if)#ip addr 10.0.15.5 255.255.255.0
R5(config-if)#no shutdown

R5(config)#int g1/0
R5(config-if)#ip addr 10.0.25.5 255.255.255.0
R5(config-if)#no shutdown

R5(config)#int f2/0
R5(config-if)#ip addr 10.0.56.5 255.255.255.0
R5(config-if)#no shutdown

R5(config)#int loopback 0
R5(config-if)#ip addr 10.0.5.5 255.255.255.255
R5(config-if)#no shutdown

R5(config)#router ospf 1
R5(config-router)#network 10.0.15.0 0.0.0.255 area 0
R5(config-router)#network 10.0.25.0 0.0.0.255 area 0
R5(config-router)#network 10.0.56.0 0.0.0.255 area 0
R5(config-router)#network 10.0.5.0 0.0.0.255 area 0
```

Toutes les adresses IP allouées sont joignables : un routeur peut ping n'importe quel autre routeur sur n'importe quelle interface.

**Q2.** Pour activer MPLS, on utilise la commande `mpls ip` sur chaque interface (sauf l'interface de loopback) de chaque routeur.

La table des labels contient l'ensemble des préfixes du réseau et même certains préfixes en plusieurs fois. En effet, ceux-ci sont joignables par plusieurs chemins grâce à OSPF.

Par exemple, pour atteindre le préfixe `10.0.12.0/24` depuis R5 on peut passer par `10.0.25.2` ou `10.0.15.1`. On peut le vérifier avec la table de routage et la table des labels de R5 :

```
R5#sh mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	10.0.1.1/32	0	Gi0/0	10.0.15.1
17	Pop tag	10.0.12.0/24	0	Gi1/0	10.0.25.2
	Pop tag	10.0.12.0/24	0	Gi0/0	10.0.15.1

Table des labels partielle de R5

```
R5#sh ip route
```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, \* - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 19 subnets, 2 masks

C	10.0.15.0/24	is directly connected, GigabitEthernet0/0
O	10.0.12.0/24	[110/2] via 10.0.25.2, 00:22:04, GigabitEthernet1/0
		[110/2] via 10.0.15.1, 00:22:04, GigabitEthernet0/0

Table de routage partielle de R5

**Q3.** R3, R4 et R6 sont les LSR upstreams du préfixe loopback de R7. Les tables des labels affichent "Pop tag" pour le préfixe `10.0.7.7`. Par exemple, sur R3 :

```
R3#sh mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
28	Pop tag	10.0.7.7/32	0	Gi3/0	10.0.37.7

Table des labels partielle de R3

On peut déduire l'ensemble des tunnels et labels utilisés entre R1 et R7 à l'aide des tables des labels. Par exemple, pour aller de R7 à R1, on regarde la table des labels de R7 :

```
R7#sh mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
18	17	10.0.1.1/32	0	Fa2/0	10.0.67.6
	22	10.0.1.1/32	0	Gi0/0	10.0.37.3

Table des labels partielle de R7

On choisit un des deux chemins disponibles, par exemple le deuxième affiché.

La table des labels de R7 indique que pour atteindre le préfixe `10.0.1.1/24`, on peut passer par `10.0.37.3` ("Next Hop") et mettre un label de 22 ("Outgoing tag"). 18 ("Local tag") correspond au label "Outgoing tag" précédent ou l'origine, ici R7.

Ensuite, on cherche la ligne ayant pour "Local tag" 22 et préfixe de destination `10.0.1.1/24` sur le routeur "Next Hop", ici R3.

De la même manière sur R3, la table des labels de R3 indique que pour atteindre le préfixe `10.0.1.1/24`, il faut passer par `10.0.23.2` ("Next Hop") et mettre un label de 20 ("Outgoing tag"). 22 ("Local tag") correspond au label "Outgoing tag" précédent.

```
R3#sh mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
22	20	10.0.1.1/32	0	Gi0/0	10.0.23.2

Table des labels partielle de R3

```
R2#sh mpls forwarding-table
Local   Outgoing   Prefix      Bytes tag  Outgoing     Next Hop
tag     tag or VC  or Tunnel Id switched   interface
20      Pop tag    10.0.1.1/32  0          Gi0/0        10.0.12.1
```

```
R7#traceroute 10.0.1.1

Type escape sequence to abort.
Tracing the route to 10.0.1.1

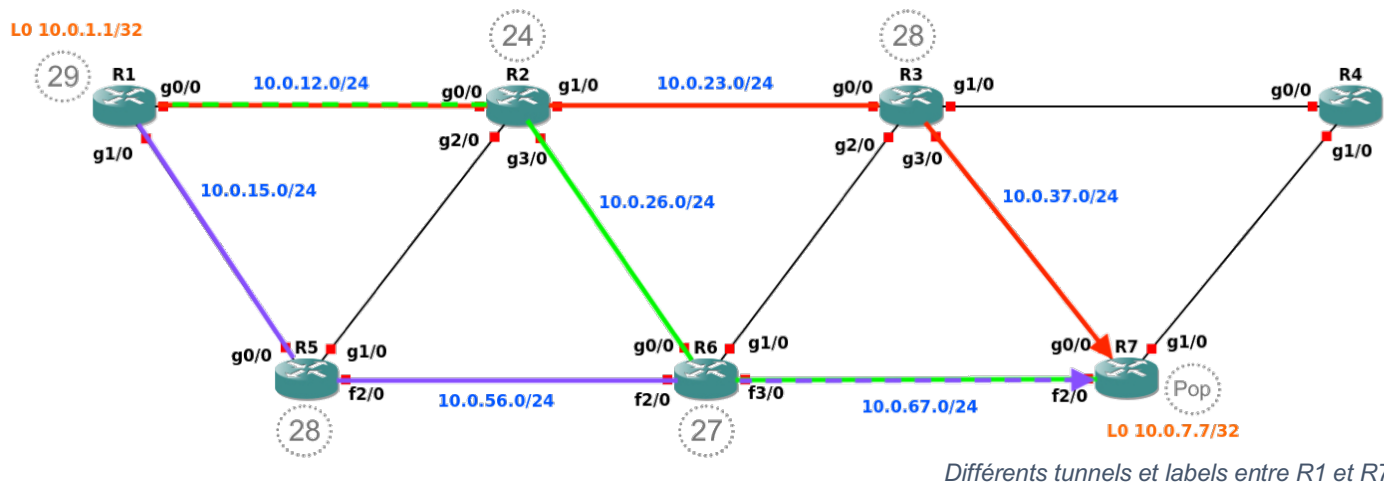
 1  10.0.67.6 [MPLS: Label 17 Exp 0] 448 msec
    10.0.37.3 [MPLS: Label 22 Exp 0] 1044 msec
    10.0.67.6 [MPLS: Label 17 Exp 0] 472 msec
 2  10.0.23.2 [MPLS: Label 20 Exp 0] 92 msec
    10.0.26.2 [MPLS: Label 20 Exp 0] 636 msec
    10.0.23.2 [MPLS: Label 20 Exp 0] 544 msec
 3  10.0.12.1 60 msec 68 msec 44 msec
```

Source	Destination	Protocol	Length	Info
10.0.67.7	10.0.1.1	ICMP	114	Echo (ping) request id=0x0000, seq=0/0, ttl=253 (reply in 130)
10.0.1.1	10.0.67.7	ICMP	118	Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (request in 129)

Source	Destination	Protocol	Length	Info
10.0.1.1	10.0.67.7	ICMP	114	Echo (ping) reply id=0x0000, seq=0/0, ttl=254

Source	Destination	Protocol	Length	Info
10.0.1.1	10.0.67.7	ICMP	114	Echo (ping) reply id=0x0000, seq=0/0, ttl=253

3/10



Différents tunnels et labels entre R1 et R7

**Q4.** On choisit le préfixe 10.0.1.1 comme destination. Un `traceroute` à partir de R6 et de R3 montre qu'il y a convergence en R2 et utilisation du même label, 23, entre R2 et R1.

```
R6#traceroute 10.0.1.1
 1 10.0.56.5 [MPLS: Label 24 Exp 0] 12 msec
   10.0.26.2 [MPLS: Label 23 Exp 0] 84 msec
   10.0.56.5 [MPLS: Label 24 Exp 0] 16 msec
 2 10.0.12.1 32 msec
   10.0.15.1 84 msec
   10.0.12.1 36 msec
```

```
R3#traceroute 10.0.1.1
 1 10.0.23.2 [MPLS: Label 23 Exp 0] 16 msec 44 msec 64 msec
 2 10.0.12.1 32 msec 48 msec 48 msec
```

Traceroute de R3 vers R1

Traceroute de R6 vers R1

**Q5.** Les valeurs des labels utilisés sur R6 dépendent des interfaces entrantes dans le sens où elles doivent correspondre à celles des labels sortants de ses routeurs upstreams (R2, R3, R5 et R7). Une même valeur de label peut être utilisée dans les deux sens sur un même lien puisque les labels sont locaux. Pour l'instant, ils servent à déterminer les différents tunnels.

## ➤ Signalisation

**Q1.** On ajoute le préfixe 10.0.6.66/32 à l'interface lo2 de R6.

```
R6(config)#int loopback 2
R6(config-if)#ip addr 10.0.6.66 255.255.255.255
R6(config-if)#no shutdown
```

On observe, à l'aide d'une capture de paquets sur le lien R5 – R6, le message OSPF "Link State Update" envoyé en multicast par R6 et la réponse OSPF du voisin R5 "Link State Acknowledge". Ces messages sont visibles sur l'ensemble des liens.

On observe aussi des messages LDP de deux types : "Label Mapping Message", visible sur l'ensemble des liens, contenant l'information du nouveau préfixe et le label associé et "Address Message", visible uniquement sur les liens directement connectés à R6, contenant l'adresse du nouveau préfixe et envoyé de R6 à ses proches voisins.

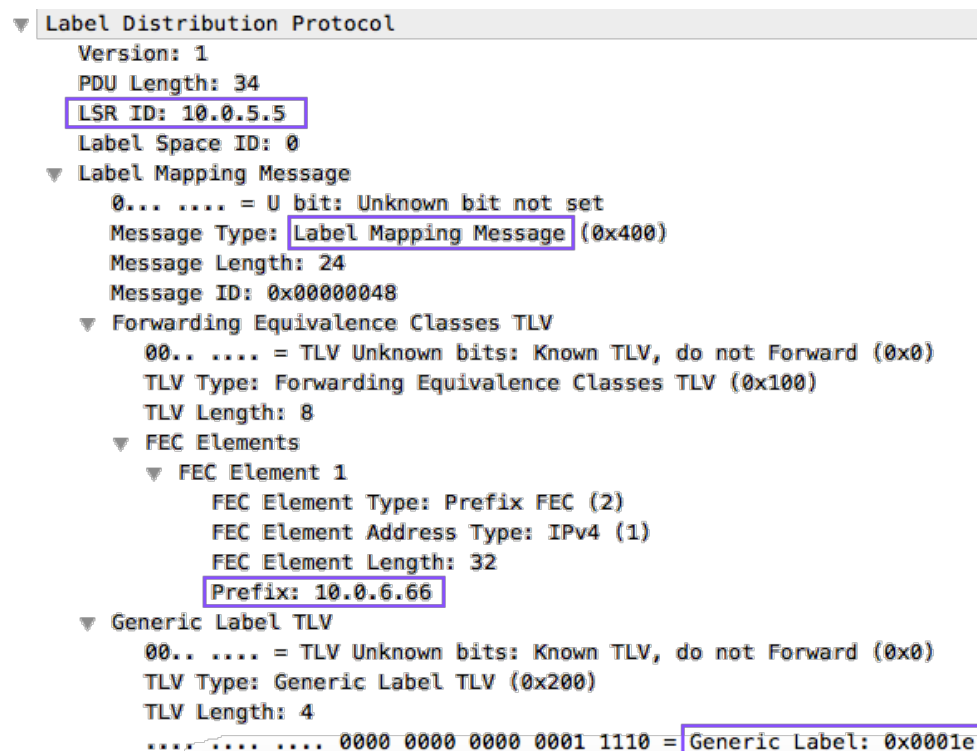
Source	Destination	Protocol	Length	Info
10.0.56.6	224.0.0.5	OSPF	158	LS Update
10.0.56.5	224.0.0.5	OSPF	78	LS Acknowledge
10.0.5.5	10.0.6.6	LDP	92	Label Mapping Message
10.0.6.6	10.0.5.5	LDP	110	Address Message Label Mapping Message

Capture de paquets sur le lien R5 – R6 avec Wireshark

Source	Destination	Protocol	Length	Info
10.0.15.5	224.0.0.5	OSPF	158	LS Update
10.0.15.1	224.0.0.5	OSPF	158	LS Update
10.0.15.1	224.0.0.5	OSPF	78	LS Acknowledge
10.0.5.5	10.0.1.1	LDP	92	Label Mapping Message
10.0.1.1	10.0.5.5	LDP	92	Label Mapping Message

Capture de paquets sur le lien R1 – R5 avec Wireshark

En effet, en regardant de plus près le message "Label Mapping Message" sur le lien R5 – R6, on obtient ces informations :



Capture de paquets sur le lien R5 – R6 avec Wireshark

Pour atteindre le préfixe 10.0.6.66, R5 ajoute le label 30 (= 1e en hexadécimal). On peut vérifier la mise à jour de la table des labels de ce dernier avec l'ajout du label 30 :

```
R5#sh mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
30	Pop tag	10.0.6.66/32	0	Fa2/0	10.0.56.6

Table des labels partielle de R5

**Q2.** ECMP est déjà utiliser par défaut. Son interaction avec MPLS est visible lors d'un traceroute : on remarque que plusieurs chemins sont disponibles.

D'abord, on observe le préfixe dans la table de routage d'un routeur. Par exemple sur R1, celui-ci est accessible en OSPF via les liens R1 – R5 et R1 – R2 :

```
R1#sh ip route
```

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 19 subnets, 2 masks					
0	10.0.6.66/32	[110/3]	via 10.0.15.5,	00:18:19,	GigabitEthernet1/0
		[110/3]	via 10.0.12.2,	00:18:19,	GigabitEthernet0/0

Table de routage partielle de R1

De même, ce préfixe est visible dans la table des labels de R1 via ces deux mêmes liens.

```
R1#sh mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
29	28	10.0.6.66/32	0	Gi1/0	10.0.15.5
	21	10.0.6.66/32	0	Gi0/0	10.0.12.2

Table des labels partielle de R1

Un traceroute affiche aussi la possibilité de passer par ces liens pour rejoindre le préfixe depuis R1 avec les labels vus dans la table précédente.

```
R1#traceroute 10.0.6.66
```

1	10.0.15.5	[MPLS: Label 28 Exp 0]	40 msec
	10.0.12.2	[MPLS: Label 21 Exp 0]	32 msec
	10.0.15.5	[MPLS: Label 28 Exp 0]	28 msec
2	10.0.26.6		24 msec
	10.0.56.6		32 msec
	10.0.26.6		28 msec

Traceroute de R1 vers R6



**Q3.** On désactive l'interface g3/0 sur R2. On observe les mêmes messages que précédemment pour OSPF. Par exemple, sur le lien R4 – R7 :

Source	Destination	Protocol	Length	Info
10.0.47.7	224.0.0.5	OSPF	134	LS Update
10.0.47.4	224.0.0.5	OSPF	78	LS Acknowledge

*Capture de paquets sur le lien R4 – R7 avec Wireshark*

Cependant, les messages LDP sont différents. D'abord "Address Withdrawal Message", sur le même principe que "Address Message" mais pour dire que celle-ci n'est plus disponible, ici 10.0.26.2. Ensuite "Label Withdrawal Message" pour retirer le label associé puis "Label Release Message", envoyé par le routeur voisin, pour signifier qu'il a bien pris en compte les changements.

Source	Destination	Protocol	Length	Info
10.0.2.2	10.0.1.1	LDP	82	Address Withdrawal Message
10.0.2.2	10.0.1.1	LDP	91	Label Withdrawal Message
10.0.1.1	10.0.2.2	LDP	91	Label Release Message
10.0.2.2	10.0.1.1	LDP	91	Label Mapping Message

*Capture de paquets sur le lien R1 – R2 avec Wireshark*

## ➤ Ingénierie de trafic

**Q1.** On considère le "sous-réseau" composé des routeurs R1, R2, R5 et R6. On active les tunnels et OSPF-TE sur l'ensemble des routeurs du "sous-réseau". Par exemple, sur R5 :

```
R5(config)#mpls traffic-eng tunnels
R5(config)#router ospf 1
R5(config-router)#mpls traffic-eng router-id Loopback0      (Loopback2 sur R6)
R5(config-router)#mpls traffic-eng area 0
```

On paramètre la bande-passante disponible sur chaque interface. On choisit des liens de bande-passante 10 000 kbps pour les liens en GigabitEthernet et 1 000 kbps pour ceux en FastEthernet. On a fait ce choix pour simplifier les commandes car par défaut, la bande-passante maximale est égale à 75 % de celle disponible sur l'interface, ie. 75 % de 1 000 Mbps pour un lien en GigabitEthernet. Par exemple, sur R6 :

```
R6(config)#int g0/0
R6(config-if)#mpls traffic-eng tunnels
R6(config-if)#ip rsvp bandwidth 10000 10000

R6(config)#int f2/0
R6(config-if)#mpls traffic-eng tunnels
R6(config-if)#ip rsvp bandwidth 1000 1000
```

On réserve trois tunnels sur R1 à destination du second loopback de R6. Pour chaque tunnel, on effectue ces commandes :

```
R1(config)#int tunnel <id>
R1(config-if)#ip unnumbered Loopback0
R1(config-if)#tunnel destination 10.0.6.66
R1(config-if)#tunnel mode mpls traffic-eng
R1(config-if)#tunnel mpls traffic-eng autoroute announce
R1(config-if)#tunnel mpls traffic-eng path-option 1 dynamic
R1(config-if)#no routing dynamic
```

Le premier tunnel, de priorité élevée et d'identifiant tunnel 0, demande de réserver une bande-passante de 9 900 kbps :

```
R1(config)#int tunnel 0
R1(config-if)#tunnel mpls traffic-eng priority 1 1
R1(config-if)#tunnel mpls traffic-eng bandwidth 9900
```

Source	Destination	Protocol	Length	Info
10.0.1.1	10.0.6.66	RSVP	238	PATH Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel ID 0
ca:02:10:94:00:08	ca:02:10:94:00:08	LOOP	60	Reply
10.0.12.2	10.0.12.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel ID 0
10.0.12.2	224.0.0.5	OSPF	186	LS Update
10.0.12.1	224.0.0.5	OSPF	194	LS Update
10.0.12.1	224.0.0.5	OSPF	78	LS Acknowledge
10.0.12.2	224.0.0.5	OSPF	78	LS Acknowledge

Capture de paquets sur le lien R1 – R2 avec Wireshark

Dans un premier temps, R1 envoie un message RSVP "PATH Message" afin d'effectuer un marquage du chemin. Dans un second temps, on peut observer qu'il y a une réservation des ressources sur le chemin qui s'effectue au travers d'un "RESV Message". Tous ces messages sont visibles sur les liens empruntés par le premier tunnel, ici R1 – R2 et R2 – R6.

On observe ensuite, le message OSPF "Link State Update" envoyé en multicast et la réponse OSPF du voisin "Link State Acknowledge". Ces messages sont visibles sur l'ensemble des liens du sous-réseau.

On observe le tunnel 0 sur R2 et les différentes informations qui le concernent :

```
R2#show mpls traffic-eng tunnels

LSP Tunnel R1_t0 is signalled, connection is up
 InLabel : GigabitEthernet0/0, 18
 OutLabel : GigabitEthernet3/0, implicit-null
 RSVP Signalling Info:
   Src 10.0.1.1, Dst 10.0.6.66, Tun_Id 0, Tun_Instance 38
 RSVP Path Info:
   My Address: 10.0.26.2
   Explicit Route: 10.0.26.6 10.0.6.66
   Record Route: NONE
   Tspec: ave rate=9900 kbits, burst=1000 bytes, peak rate=9900 kbits
 RSVP Resv Info:
   Record Route: NONE
   Fspec: ave rate=9900 kbits, burst=1000 bytes, peak rate=9900 kbits
```

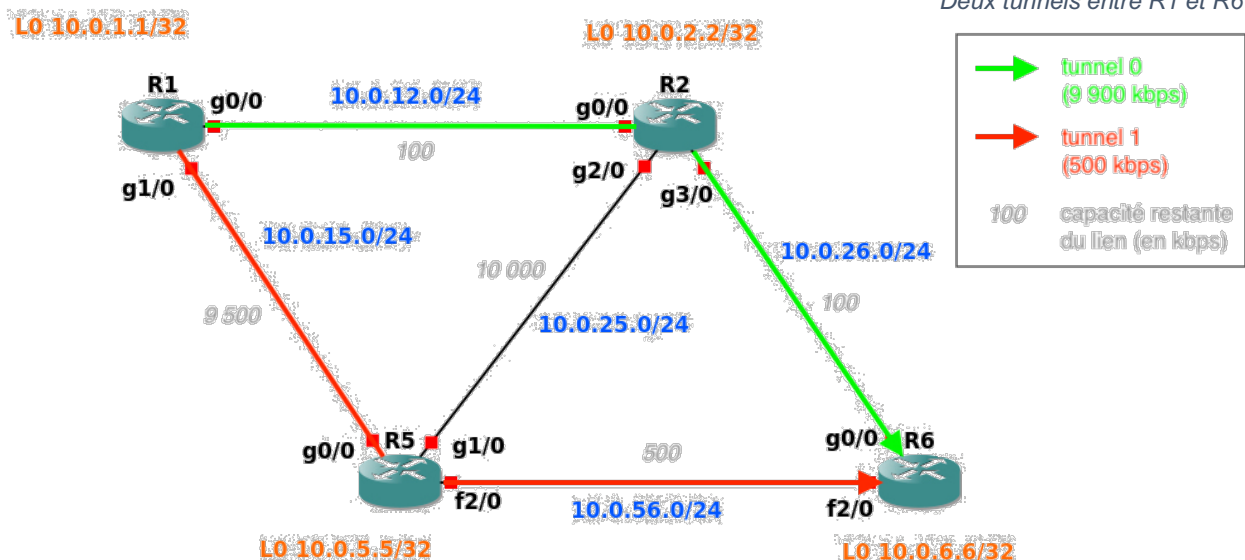
Liste des tunnels sur R2

Le deuxième tunnel, de priorité moins élevée et d'identifiant tunnel 1, demande de réserver une bande-passante de 500 kbps :

```
R1(config)#int tunnel 1
R1(config-if)#tunnel mpls traffic-eng priority 2 2
R1(config-if)#tunnel mpls traffic-eng bandwidth 500
```

Ici nous pourrons observer les mêmes messages que pour le tunnel précédent, la seule différence est le chemin réservé. En effet le tunnel 0 réserve une bande-passante de 9 900 kbps, sachant que le lien a une capacité de 10 000 kbps, il ne reste donc plus que 100 kbps. Or le tunnel 1 demande une bande-passante de 500 kbps. De plus, sa priorité est moins élevée que le tunnel 0 donc le lien R1 – R2 est "éliminé".

Ainsi le chemin réservé est R1 → R5 → R6.



Le troisième tunnel, de priorité encore moins élevée et d'identifiant `tunnel 2`, demande de réserver une bande-passante de 1 000 kbps :

```
R1(config)#int tunnel 2
R1(config-if)#tunnel mpls traffic-eng priority 3 3
R1(config-if)#tunnel mpls traffic-eng bandwidth 1000
```

Pour les mêmes raisons que le `tunnel 1`, les liens `R1 – R2` et `R1 – R5` sont "éliminés". Ainsi aucune réservation de chemin n'est possible. Sur les captures de paquet, nous n'observons rien en relation avec le `tunnel 2`.

Néanmoins, à l'aide de la commande `show mpls traffic-eng tunnels`, on peut observer que ce dernier tunnel est bien présent mais désactivé :

```
Name: R1_t2 (Tunnel2) Destination: 10.0.6.66
Status:
  Admin: up      Oper: down  Path: not valid  Signalling: Down
  path option 1, type dynamic

Config Parameters:
  Bandwidth: 1000 kbps (Global) Priority: 3 1 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 1000 bw-based
  auto-bw: disabled

History:
  Tunnel:
    Time since created: 3 minutes, 38 seconds
    Time since path change: 2 minutes, 47 seconds
  Prior LSP:
    ID: path option 1 [63]
    Removal Trigger: path error
    Last Error: PCALC:: No path to destination, 10.0.6.66
```

Liste partielle des tunnels sur R1

La liste des réservations RSVP sur R6 comprend donc les deux premiers tunnels créés uniquement :

```
R6#sh ip rsvp reservation
To      From      Pro DPort Sport Next Hop      I/F      Fi Serv BPS
10.0.6.66 10.0.1.1    0  0    3    10.0.6.66      SE LOAD 9900K
10.0.6.66 10.0.1.1    0  1    1    10.0.6.66      SE LOAD 500K
```

Réserve RSVP sur R6

**Q2.** On considère le même "sous-réseau" qu'à la question précédente. On "supprime" le dernier tunnel créé et on diminue la priorité du premier tunnel :

```
R1(config)#no int tunnel 2
R1(config)#int tunnel 0
R1(config-if)#no tunnel mpls traffic-eng priority 1 1
R1(config-if)#tunnel mpls traffic-eng priority 2 2
```

On augmente la priorité et la bande-passante demandée du deuxième tunnel (de 500 à 8 000 kbps) :

```
R1(config)#int tunnel 1
R1(config-if)#no tunnel mpls traffic-eng priority 2 2
R1(config-if)#no tunnel mpls traffic-eng bandwidth 500
R1(config-if)#tunnel mpls traffic-eng bandwidth 8000
R1(config-if)#tunnel mpls traffic-eng priority 1 1
```

Source	Destination	Protocol	Length	Info
10.0.1.1	10.0.6.66	RSVP	238	PATH Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel ID 1,
10.0.12.2	10.0.12.1	RSVP	166	PATH ERROR Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel
10.0.12.2	10.0.12.1	RSVP	126	RESV TEAR Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel
10.0.12.2	10.0.12.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel ID 1,
10.0.12.2	224.0.0.5	OSPF	186	LS Update
10.0.1.1	10.0.6.66	RSVP	170	PATH TEAR Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel
10.0.1.1	10.0.6.66	RSVP	170	PATH TEAR Message. SESSION: IPv4-LSP, Destination 10.0.6.66, Short Call ID 0, Tunnel
10.0.12.1	224.0.0.5	OSPF	194	LS Update
10.0.12.1	224.0.0.5	OSPF	186	LS Update
224.0.0.5	OSPF	78	LS Acknowledge	
224.0.0.5	OSPF	98	LS Acknowledge	

Capture de paquets sur le lien `R1 – R2` avec Wireshark



Par rapport à la question précédente, on observe des messages RSVP supplémentaires. D'abord un "PATH ERROR Message" émis par le tunnel 0 dû au fait que le tunnel 1 "écrase" la réservation du chemin faite par le tunnel 0. Puis des "PATH TEAR Message" qui permettent ici de supprimer le tunnel 0 (ils peuvent aussi résulter d'un timeout sur la réservation mais pas ici). On observe le tunnel 1 préempter le tunnel 0. En effet, ce dernier a une priorité moins élevée.

La liste des tunnels sur R6 ne comprend plus qu'un tunnel, le deuxième créé :

```
R6#show mpls traffic-eng tunnels

LSP Tunnel R1_t1 is signalled, connection is up
InLabel  : GigabitEthernet0/0, implicit-null
OutLabel  : -
RSVP Signalling Info:
  Src 10.0.1.1, Dst 10.0.6.66, Tun_Id 1, Tun_Instance 39
RSVP Path Info:
  My Address: 10.0.6.66
  Explicit Route: NONE
  Record Route: NONE
  Tspec: ave rate=8000 kbits, burst=1000 bytes, peak rate=8000 kbits
RSVP Resv Info:
  Record Route: NONE
  Fspec: ave rate=8000 kbits, burst=1000 bytes, peak rate=8000 kbits
```

Liste des tunnels sur R6

### ➤ Plusieurs types de tunnels

**Q1.** Dans l'exercice précédent, nous nous sommes restreints à un "sous-réseau". Ici, on veut passer par l'ensemble des routeurs donc il faut activer les tunnels et OSPF-TE sur les routeurs restants et paramétrer la bande-passante disponible sur chaque interface (cf. *Ingénierie de trafic, Q1*).

Nous n'avons pas trouvé de commandes pour expliciter un chemin au tunnel ou pour forcer le passage par une liste de routeurs. Nous choisissons donc de retirer l'annonce de certains préfixes dans OSPF sur chaque routeur dans le but d'obtenir ce tunnel R1 → R5 → R2 → R6 → R3 → R7 → R4.

Par exemple, on fait en sorte que R6 n'annonce plus les préfixes 10.0.56.0/24 et 10.0.67.0/24, c'est-à-dire qu'il annonce uniquement 10.0.26.0/24, 10.0.36.0/24 et 10.0.6.0/24 pour faire passer le tunnel par R2 → R6 → R3 :

```
R6(config)#router ospf 1
R6(config-router)#no network 10.0.56.0 0.0.0.255 area 0
R6(config-router)#no network 10.0.67.0 0.0.0.255 area 0
```

On peut ensuite créer le tunnel sur R1 à destination de R4 avec une bande-passante de 1 000 kbps pour pouvoir utiliser les liens FastEthernet :

```
R1(config)#int tunnel 0
R1(config-if)#ip unnumbered Loopback0
R1(config-if)#tunnel destination 10.0.4.4
R1(config-if)#tunnel mode mpls traffic-eng
R1(config-if)#tunnel mpls traffic-eng autoroute announce
R1(config-if)#tunnel mpls traffic-eng priority 1 1
R1(config-if)#tunnel mpls traffic-eng bandwidth 1000
R1(config-if)#tunnel mpls traffic-eng path-option 1 dynamic
R1(config-if)#no routing dynamic
```

Le tunnel est bien actif avec comme chemin R1 → R5 → R2 → R6 → R3 → R7 → R4.

```
R1#show mpls traffic-eng tunnels

Name: R1_t0                                (Tunnel0) Destination: 10.0.4.4
Status:
  Admin: up          Oper: up          Path: valid          Signalling: connected

  path option 1, type dynamic (Basis for Setup, path weight 6)

Config Parameters:
  Bandwidth: 1000      kbps (Global) Priority: 1 1  Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled   LockDown: disabled Loadshare: 1000    bw-based
  auto-bw: disabled

InLabel  : -
OutLabel : GigabitEthernet1/0, 27
RSVP Signalling Info:
  Src 10.0.1.1, Dst 10.0.4.4, Tun_Id 0, Tun_Instance 3
RSVP Path Info:
  My Address: 10.0.15.1
  Explicit Route: 10.0.15.5 10.0.25.5 10.0.25.2 10.0.26.2
                  10.0.26.6 10.0.36.6 10.0.36.3 10.0.37.3
                  10.0.37.7 10.0.47.7 10.0.47.4 10.0.4.4
  Record Route: NONE
  Tspec: ave rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
RSVP Resv Info:
  Record Route: NONE
  Fspec: ave rate=1000 kbits, burst=1000 bytes, peak rate=1000 kbits
```

Liste des tunnels sur R1

**Q2.** La commande `mpls ip propagate-ttl` est activée par défaut.

On obtient ces informations lors d'un `traceroute` :

```
R1#traceroute 10.0.4.4

 1 10.0.15.5 [MPLS: Label 27 Exp 0] 120 msec 96 msec 92 msec
 2 10.0.25.2 [MPLS: Label 27 Exp 0] 100 msec 92 msec 100 msec
 3 10.0.26.6 [MPLS: Label 26 Exp 0] 88 msec 84 msec 116 msec
 4 10.0.36.3 [MPLS: Label 27 Exp 0] 96 msec 136 msec 92 msec
 5 10.0.37.7 [MPLS: Label 27 Exp 0] 80 msec 60 msec 72 msec
 6 10.0.47.4 72 msec 108 msec 84 msec
```

```
R1#traceroute 10.0.4.4

 1 10.0.47.4 84 msec 84 msec 152 msec
```

Traceroute de R1 à R4  
(sans propagation TTL)

Traceroute de R1 à R4  
(avec propagation TTL)

Lorsqu'on la désactive avec `no mpls ip propagate-ttl`, le `traceroute` est différent. En effet, lorsque la propagation du TTL est désactivée, le TTL de MPLS est calculé indépendamment du TTL IP et ce dernier reste donc constant le long du tunnel. Les différents sauts ne sont donc pas enregistrés par un `traceroute`.

**Q3.** Nous n'avons pas trouvé de différences en activant ou désactivant la commande `mpls ip ttl-expiration pop <1..6>`.