

# Modes de tunnellation DiffServ pour les réseaux MPLS

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Comportement par défaut](#)

[Pré utilisation des commandes et comportement du logiciel 12.2\(13\)T de Cisco IOS](#)

[Utilisation des commandes et comportement du logiciel 12.2\(13\)T de Cisco IOS de courrier](#)

[Modes de Tunnellation de DiffServ](#)

[Mode uniforme](#)

[Mode de canal](#)

[Mode de Court-canal](#)

[Résumé de tunnel mode](#)

[Informations connexes](#)

## [Introduction](#)

Ce document décrit la mise en œuvre des modes de tunnellation de services différenciés (DiffServ) disponibles pour les environnements de réseau basés sur la Commutation multiprotocole par étiquette (MPLS).

## [Conditions préalables](#)

### [Conditions requises](#)

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- MPLS et MPLS pour le Réseaux privés virtuels (VPN)
- Concepts concernant la Priorité IP, le Type de service (ToS) et le DiffServ
- Fonction Quality of Service (QoS) Packet Marking et classification utilisant l'interface de ligne de commande modulaire CLI (MQC) de QoS

### [Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de

logiciel suivantes :

- Release 12.2(13)T de <sup>®</sup> de logiciel de Cisco IOS pour de nouvelles caractéristiques de QoS. La release 12.1(5)T inclut les caractéristiques d'origine de QoS.
- Tout routeur de Cisco de la gamme 3600 ou de plus tard, comme le Cisco 3660 ou les 7206 qui prennent en charge la fonctionnalité de routeur du noyau de fournisseur MPLS (p)/routeur de Provider Edge (PE).

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

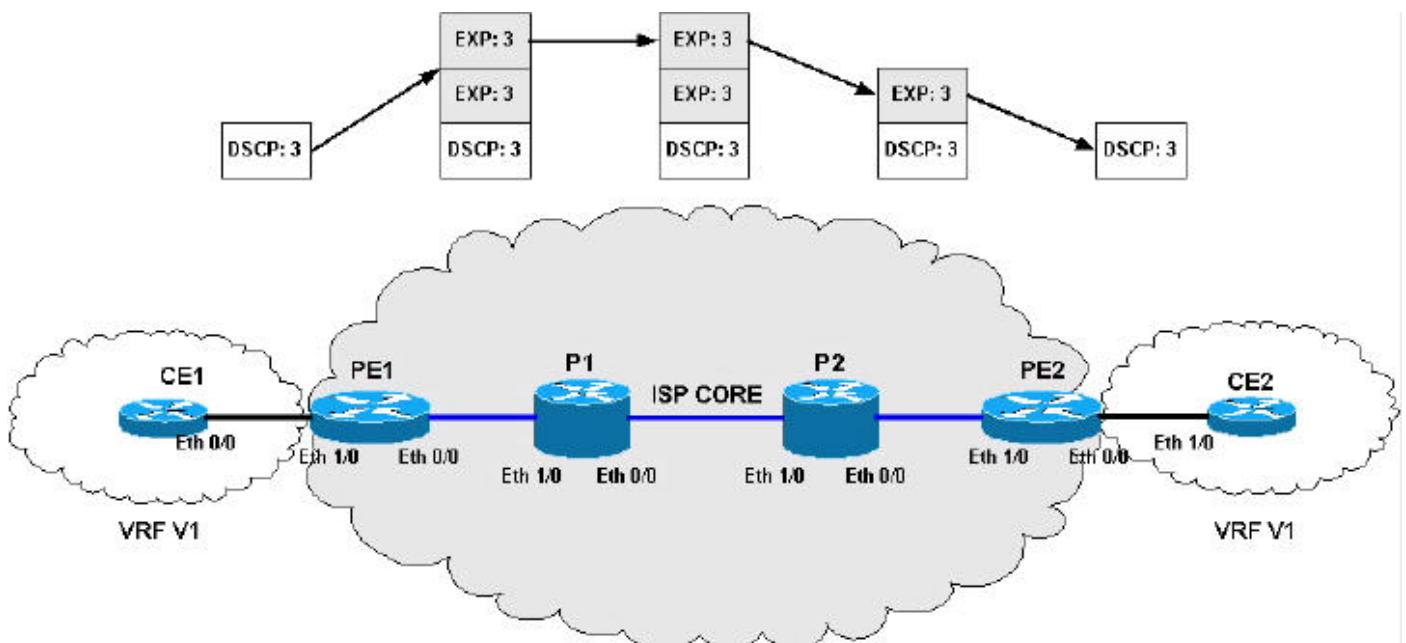
## Conventions

For more information on document conventions, refer to the [Cisco Technical Tips Conventions](#).

## Informations générales

### Comportement par défaut

Ce diagramme décrit le comportement par défaut des bits expérimentaux de point de code de DiffServ (DSCP) /MPLS (EXP) pendant qu'un paquet voyage d'un routeur de Customer Edge (CE) à un autre routeur CE à travers un noyau MPLS :



Ce reste de cette section décrit l'activité dans le diagramme de comportement par défaut.

Imposition de l'étiquette (IP -> étiquette) :

- La Priorité IP du paquet IP entrant est copiée sur les bits d'EXP MPLS de toutes les étiquettes poussées.
- Les trois 1ers bits du bit de DSCP est copiés sur les bits d'EXP MPLS de toutes les étiquettes poussées.

- Cette technique est également connue comme réflexion de tos.

Expédition MPLS (étiquette – > étiquette) :

- L'EXP est copié sur les nouvelles étiquettes qui sont permutées/poussées pendant l'expédition ou l'imposition.
- À l'imposition d'étiquette, les étiquettes étant à la base ne sont pas modifiées avec la valeur de la nouvelle étiquette étant ajoutée à la pile d'étiquette en cours.
- À la disposition d'étiquette, les bits d'EXP ne sont pas copiés vers le bas sur les bits nouvellement exposés d'EXP d'étiquette.

Disposition d'étiquette (étiquette – > IP) :

- À la disposition d'étiquette les bits d'EXP ne sont pas copiés vers le bas sur le champ IP precedence/DSCP du paquet IP nouvellement exposé.

## [Pré utilisation des commandes et comportement du logiciel 12.2\(13\)T de Cisco IOS](#)

Avant que la release 12.2(13)T IOS, la commande de [set mpls experimental](#) ait été la seule méthode disponible pour modifier les bits d'EXP MPLS.

## [Utilisation des commandes et comportement du logiciel 12.2\(13\)T de Cisco IOS de courrier](#)

Avec la version 12.2(13)T et ultérieures IOS, la commande de [set mpls experimental](#) est modifiée de permettre ces options :

- [set mpls experimental topmost {MPLS-exp-valeur | qos-groupe \[table-MAP-nom de table\]}](#)
- [set mpls experimental imposition {MPLS-exp-valeur | qos-groupe \[table-MAP-nom de table\]}](#)

**Remarque:** La nouvelle commande de [set mpls experimental topmost](#) est équivalente à la vieille commande d'imposition de MPLS de positionnement.

Ces deux commandes, en combinaison avec quelques nouveaux commutateurs, permettent un meilleur contrôle de modification de configuration binaire d'EXP MPLS pendant le pousser d'étiquette, l'échange, et les exécutions pops. Ces deux commandes te permettent pour utiliser des modes de Tunnellisation de DiffServ.

## [Modes de Tunnellisation de DiffServ](#)

Les modes de Tunnellisation de DiffServ introduit un nouveau Par-Saut-comportement (PHB), qui permet QoS différencié dans un réseau de fournisseurs. Le mode de Tunnellisation est défini à la périphérie du réseau, normalement dans les Routeurs de commutateur d'étiquette de PE (LSRs) (d'entrée et de sortie). Vous pouvez devoir apporter des modifications dans les Routeurs P ; vous devez également considérer ce qui se produit quand l'étiquette le plus élevé est retirée d'un paquet dû à Pénultième-Saut-sauter (PHP). Il peut être nécessaire de copier la valeur d'EXP MPLS de l'étiquette supérieure qui est sautée à l'étiquette nouvellement exposée ; ceci ne s'applique pas toujours à tous les modes de Tunnellisation.

Dans certains cas (par exemple, un réseau MPLS de brute non-VPN), l'action PHP sur le routeur de la finale P peut exposer un paquet IP ordinaire quand un paquet avec seulement une étiquette est reçu. Quand ce paquet IP est reçu par le de sortie LSR (PE), il n'est pas possible de classier

le paquet basé sur les bits d'EXP MPLS parce qu'il n'y a aucune étiquette maintenant. Dans ces situations, vous devez configurer le routeur PE de sortie pour annoncer une étiquette d'**explicit-null**. Quand l'action PHP est exécutée sur le routeur P, une étiquette avec une valeur de zéro est envoyée, et par cette étiquette spéciale vous pouvez identifier les bits d'EXP en tant que paquets normalement étiquetés, permettant la classification correcte sur le routeur PE de sortie.

Le support de réseau MPLS de la spécification de DiffServ définit ces modes de Tunnellisation :

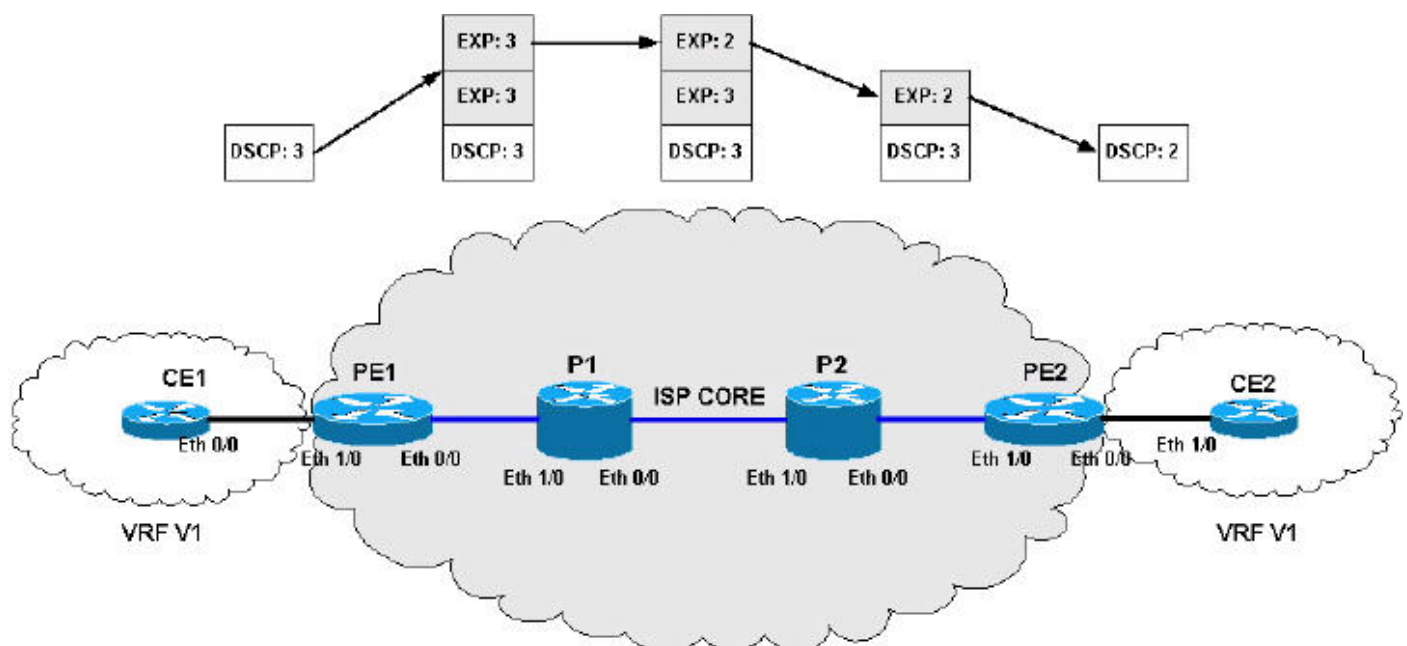
- [Uniforme](#)
- [Canal](#)
- [Court-canal](#)

Les sections suivantes examinent chaque mode de Tunnellisation séparément, et fournissent des exemples pour afficher comment chaque mode de Tunnellisation peut être configuré. Les exemples incluent un plein mappage de Priorité IP aux bits d'EXP MPLS. Il est possible d'avoir un certain nombre de différents paramètres de QoS et modes de Tunnellisation pour chaque client.

**Remarque:** Les exemples de configuration ne sont pas spécifiques pour MPLS VPN, et s'appliquent pour les réseaux MPLS ordinaires et les réseaux de transporteur pris en charge par transporteur (CsC). Il est également possible que votre réseau puisse varier d'un autre réseau — beaucoup de différents paramètres de QoS et modes de Tunnellisation peuvent être utilisés.

## [Mode uniforme](#)

Le DiffServ perçant un tunnel le mode uniforme a seulement une couche de QoS, qui atteint de bout en bout. Le routeur PE d'entrée (PE1) copie le DSCP du paquet IP entrant dans les bits d'EXP MPLS des étiquettes imposées. Pendant que les bits d'EXP voyagent par le noyau, ils peuvent ou ne peuvent être modifiés par les Routeurs intermédiaires P. Dans cet exemple, le routeur P1 P modifie les bits d'EXP de l'étiquette supérieure. Au routeur du de sortie P (P2) nous copions les bits d'EXP sur les bits d'EXP de l'étiquette nouvellement exposée après le PHP (Pénultième-Saut-bruit). Enfin au routeur PE de sortie (PE2) nous copions alors les bits d'EXP sur les bits de DSCP du paquet IP nouvellement exposé.



Configurations de mode uniformes :

## PE1

```
!--- This configuration maps the IP Precedence !--- of
the incoming IP packets to the MPLS EXP bits. class-map
match-all IP-AF11 match ip precedence 0 class-map match-
all IP-AF12 match ip precedence 1 class-map match-all
IP-AF21 match ip precedence 2 class-map match-all IP-
AF22 match ip precedence 3 class-map match-all IP-AF31
match ip precedence 4 class-map match-all IP-AF32 match
ip precedence 5 class-map match-all MPLS-AF11 match mpls
experimental topmost 0 class-map match-all MPLS-AF12
match mpls experimental topmost 1 class-map match-all
MPLS-AF21 match mpls experimental topmost 2 class-map
match-all MPLS-AF22 match mpls experimental topmost 3
class-map match-all MPLS-AF31 match mpls experimental
topmost 4 class-map match-all MPLS-AF32 match mpls
experimental topmost 5 policy-map set-MPLS-PHB
  class IP-AF11
    police 8000 conform-action set-mpls-exp-imposition-
transmit
                                0 exceed-
action drop
  class IP-AF12
    police 10000 conform-action set-mpls-exp-imposition-
transmit
                                1 exceed-action set-mpls-exp-imposition-
transmit 0
  class IP-AF21
    police 12000 conform-action set-mpls-exp-imposition-
transmit
                                2 exceed-action set-mpls-exp-imposition-
transmit 1
  class IP-AF22
    police 12000 conform-action set-mpls-exp-imposition-
transmit
                                3 exceed-action set-mpls-exp-imposition-
transmit 2
  class IP-AF31
    police 12000 conform-action set-mpls-exp-imposition-
transmit
                                4 exceed-action set-mpls-exp-imposition-
transmit 3
  class IP-AF32
    police 12000 conform-action set-mpls-exp-imposition-
transmit
                                5 exceed-action set-mpls-exp-imposition-
transmit 4

policy-map output-qos
  class MPLS-AF11
    bandwidth percent 5
    random-detect
  class MPLS-AF12
    bandwidth percent 10
    random-detect
  class MPLS-AF21
    bandwidth percent 10
    random-detect
  class MPLS-AF22
    bandwidth percent 15
    random-detect
  class MPLS-AF31
```

```

bandwidth percent 20
random-detect
class MPLS-AF32
bandwidth percent 30
random-detect

interface Ethernet0/0
ip address 192.168.0.1 255.255.255.0
max-reserved-bandwidth 90
service-policy output output-qos
tag-switching ip
!
interface Ethernet1/0
ip vrf forwarding v1
ip address 10.0.0.2 255.255.255.0
service-policy input set-MPLS-PHB
!

```

## P1

```

!--- This configuration swaps the top label from 3 to 2,
!--- which does not need to occur to follow the previous
configuration. class-map match-all mpls-in match mpls
experimental topmost 3 ! policy-map mpls-in
class mpls-in
set mpls experimental topmost 2
!
interface Ethernet0/0
ip address 192.168.1.1 255.255.255.0
tag-switching ip
!
interface Ethernet1/0
ip address 192.168.0.2 255.255.255.0
service-policy input mpls-in
tag-switching ip
!

```

## P2

```

!--- Remember to copy down the MPLS EXP value !--- from
the newly exposed label after the PHP. class-map match-
all MPLS-AF11 match mpls experimental topmost 0 class-
map match-all MPLS-AF12 match mpls experimental topmost
1 class-map match-all MPLS-AF21 match mpls experimental
topmost 2 class-map match-all MPLS-AF22 match mpls
experimental topmost 3 class-map match-all MPLS-AF31
match mpls experimental topmost 4 class-map match-all
MPLS-AF32 match mpls experimental topmost 5 ! class-map
match-all qos-group-AF11 match qos-group 0 class-map
match-all qos-group-AF12 match qos-group 1 class-map
match-all qos-group-AF21 match qos-group 2 class-map
match-all qos-group-AF22 match qos-group 3 class-map
match-all qos-group-AF31 match qos-group 4 class-map
match-all qos-group-AF32 match qos-group 5 ! policy-map
qos-group-in
class MPLS-AF11
set qos-group mpls experimental topmost
class MPLS-AF12
set qos-group mpls experimental topmost
class MPLS-AF21
set qos-group mpls experimental topmost

```

```

class MPLS-AF22
  set qos-group mpls experimental topmost
class MPLS-AF31
  set qos-group mpls experimental topmost
class MPLS-AF32
  set qos-group mpls experimental topmost
!
policy-map qos-group-out
class qos-group-AF11
  bandwidth percent 5
  random-detect
  set mpls experimental topmost qos-group
class qos-group-AF12
  bandwidth percent 10
  random-detect
  set mpls experimental topmost qos-group
class qos-group-AF21
  bandwidth percent 10
  random-detect
  set mpls experimental topmost qos-group
class qos-group-AF22
  bandwidth percent 15
  random-detect
  set mpls experimental topmost qos-group
class qos-group-AF31
  bandwidth percent 20
  random-detect
  set mpls experimental topmost qos-group
class qos-group-AF32
  bandwidth percent 30
  random-detect
  set mpls experimental topmost qos-group
!
interface Ethernet0/0
  ip address 192.168.2.1 255.255.255.0
  max-reserved-bandwidth 90
  service-policy output qos-group-out
  tag-switching ip
!
interface Ethernet1/0
  ip address 192.168.1.2 255.255.255.0
  service-policy input qos-group-in
  tag-switching ip
!

```

## PE2

```

!--- Remember to copy down the MPLS EXP bits to the IP
Precedence !--- of the newly exposed IP packet. class-
map match-all MPLS-AF11 match mpls experimental topmost
0 class-map match-all MPLS-AF12 match mpls experimental
topmost 1 class-map match-all MPLS-AF21 match mpls
experimental topmost 2 class-map match-all MPLS-AF22
match mpls experimental topmost 3 class-map match-all
MPLS-AF31 match mpls experimental topmost 4 class-map
match-all MPLS-AF32 match mpls experimental topmost 5 !
class-map match-all qos-group-AF11 match qos-group 0
class-map match-all qos-group-AF12 match qos-group 1
class-map match-all qos-group-AF21 match qos-group 2
class-map match-all qos-group-AF22 match qos-group 3
class-map match-all qos-group-AF31 match qos-group 4
class-map match-all qos-group-AF32 match qos-group 5 !

```

```

policy-map qos-group-in
  class MPLS-AF11
    set qos-group mpls experimental topmost
  class MPLS-AF12
    set qos-group mpls experimental topmost
  class MPLS-AF21
    set qos-group mpls experimental topmost
  class MPLS-AF22
    set qos-group mpls experimental topmost
  class MPLS-AF31
    set qos-group mpls experimental topmost
  class MPLS-AF32
    set qos-group mpls experimental topmost
!
policy-map qos-group-out
  class qos-group-AF11
    bandwidth percent 5
    random-detect
    set precedence qos-group
  class qos-group-AF12
    bandwidth percent 10
    random-detect
    set precedence qos-group
  class qos-group-AF21
    bandwidth percent 10
    random-detect
    set precedence qos-group
  class qos-group-AF22
    bandwidth percent 15
    random-detect
    set precedence qos-group
  class qos-group-AF31
    bandwidth percent 20
    random-detect
    set precedence qos-group
  class qos-group-AF32
    bandwidth percent 30
    random-detect
    set precedence qos-group
!
interface Ethernet0/0
  ip vrf forwarding v1
  ip address 10.0.1.1 255.255.255.0
  max-reserved-bandwidth 90
  service-policy output qos-group-out
!
interface Ethernet1/0
  ip address 192.168.2.2 255.255.255.0
  service-policy input qos-group-in
  tag-switching ip
!

```

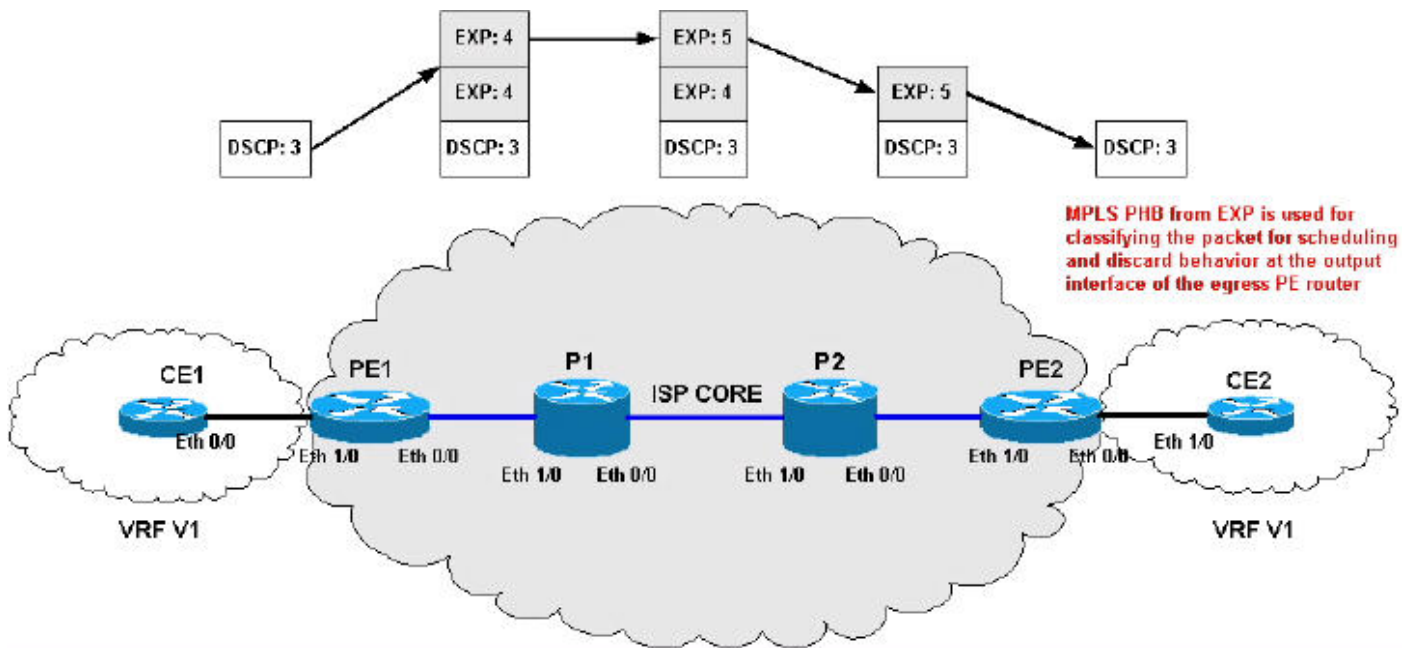
## [Mode de canal](#)

Le mode de canal de Tunnellisation de DiffServ utilise deux couches de QoS :

1. Un QoS sous-jacent pour les données, qui demeurent sans changement en traversant le noyau.
2. Un par-noyau QoS, qui est séparé de celui des paquets IP sous-jacents. Ce par-noyau QoS PHB demeure transparent aux utilisateurs finaux.



Quand un paquet atteint la périphérie du noyau MPLS, le routeur PE de sortie (PE2) classe les paquets IP nouvellement exposés pour la Mise en file d'attente sortante basée sur le MPLS PHB des bits d'EXP de l'étiquette récemment retirée.



Configurations de mode de canal :

```

PE1

!--- On input, the IP Precedence is copied to the MPLS
EXP !--- on label imposition and when leaving the
router, the !--- MPLS EXP bits are used to classify the
traffic into three !--- traffic classes-gold, silver,
and bronze (the topmost). class-map match-all IP-AF22
match ip precedence 0 1 class-map match-all IP-AF31
match ip precedence 2 3 class-map match-all IP-AF32
match ip precedence 4 5 ! class-map match-all bronze
match mpls experimental topmost 2 3 class-map match-all
silver match mpls experimental topmost 4 class-map
match-all gold match mpls experimental topmost 5 !
policy-map set-MPLS-PHB
  class IP-AF22
    police 12000 conform-action set-mpls-exp-imposition-
transmit
                3 exceed-action set-mpls-exp-imposition-
transmit 2
  class IP-AF31
    police 12000 conform-action set-mpls-exp-imposition-
transmit
                4 exceed-action set-mpls-exp-imposition-
transmit 3
  class IP-AF32
    police 12000 conform-action set-mpls-exp-imposition-
transmit
                5 exceed-action set-mpls-exp-imposition-
transmit 4
!
policy-map output-qos
  class gold
    bandwidth 40
    random-detect
  
```

```

class silver
  bandwidth 30
  random-detect
class bronze
  bandwidth 20
  random-detect
!
interface Ethernet0/0
  ip address 192.168.0.1 255.255.255.0
  max-reserved-bandwidth 90
  service-policy output output-qos
  tag-switching ip
!
interface Ethernet1/0
  ip vrf forwarding v1
  ip address 10.0.0.2 255.255.255.0
  service-policy input set-MPLS-PHB
!

```

## P1

```

!--- This configuration swaps the top label from 4 to 5,
!--- which does not need to occur to follow the previous
configuration. class-map match-all mpls-in match mpls
experimental topmost 4 ! policy-map mpls-in
  class mpls-in
    set mpls experimental topmost 5
!
interface Ethernet0/0
  ip address 192.168.1.1 255.255.255.0
  tag-switching ip
!
interface Ethernet1/0
  ip address 192.168.0.2 255.255.255.0
  service-policy input mpls-in
  tag-switching ip
!

```

## P2

```

!--- Remember to copy down the MPLS EXP value from !---
the newly exposed label after the PHP. class-map match-
all MPLS-AF11 match mpls experimental topmost 0 class-
map match-all MPLS-AF12 match mpls experimental topmost
1 class-map match-all MPLS-AF21 match mpls experimental
topmost 2 class-map match-all MPLS-AF22 match mpls
experimental topmost 3 class-map match-all MPLS-AF31
match mpls experimental topmost 4 class-map match-all
MPLS-AF32 match mpls experimental topmost 5 ! class-map
match-all gold match qos-group 5 class-map match-all
silver match qos-group 4 class-map match-all bronze
match qos-group 3 match qos-group 2 ! policy-map qos-
group-in
  class MPLS-AF11
    set qos-group mpls experimental topmost
  class MPLS-AF12
    set qos-group mpls experimental topmost
  class MPLS-AF21
    set qos-group mpls experimental topmost
  class MPLS-AF22
    set qos-group mpls experimental topmost

```

```

class MPLS-AF31
  set qos-group mpls experimental topmost
class MPLS-AF32
  set qos-group mpls experimental topmost
!
policy-map qos-group-out
  class gold
    bandwidth 40
    random-detect
    set mpls experimental topmost qos-group
  class silver
    bandwidth 30
    random-detect
    set mpls experimental topmost qos-group
  class bronze
    bandwidth 20
    random-detect
    set mpls experimental topmost qos-group
!
interface Ethernet0/0
  ip address 192.168.2.1 255.255.255.0
  max-reserved-bandwidth 90
  service-policy output qos-group-out
  tag-switching ip
!
interface Ethernet1/0
  ip address 192.168.1.2 255.255.255.0
  service-policy input qos-group-in
  tag-switching ip
!

```

## PE2

*!--- Remember to queue the newly exposed IP packet based in !--- the MPLS EXP bits of the label we just removed. Use !--- qos-groups to keep track of this value.*

```

class-map match-all MPLS-AF11
  match mpls experimental topmost 0
class-map match-all MPLS-AF12
  match mpls experimental topmost 1
class-map match-all MPLS-AF21
  match mpls experimental topmost 2
class-map match-all MPLS-AF22
  match mpls experimental topmost 3
class-map match-all MPLS-AF31
  match mpls experimental topmost 4
class-map match-all MPLS-AF32
  match mpls experimental topmost 5
!
class-map match-all gold
  match qos-group 5
class-map match-all silver
  match qos-group 4
class-map match-all bronze
  match qos-group 3
  match qos-group 2
!
policy-map qos-group-in
  class MPLS-AF11
    set qos-group mpls experimental topmost
    set discard-class 0

```

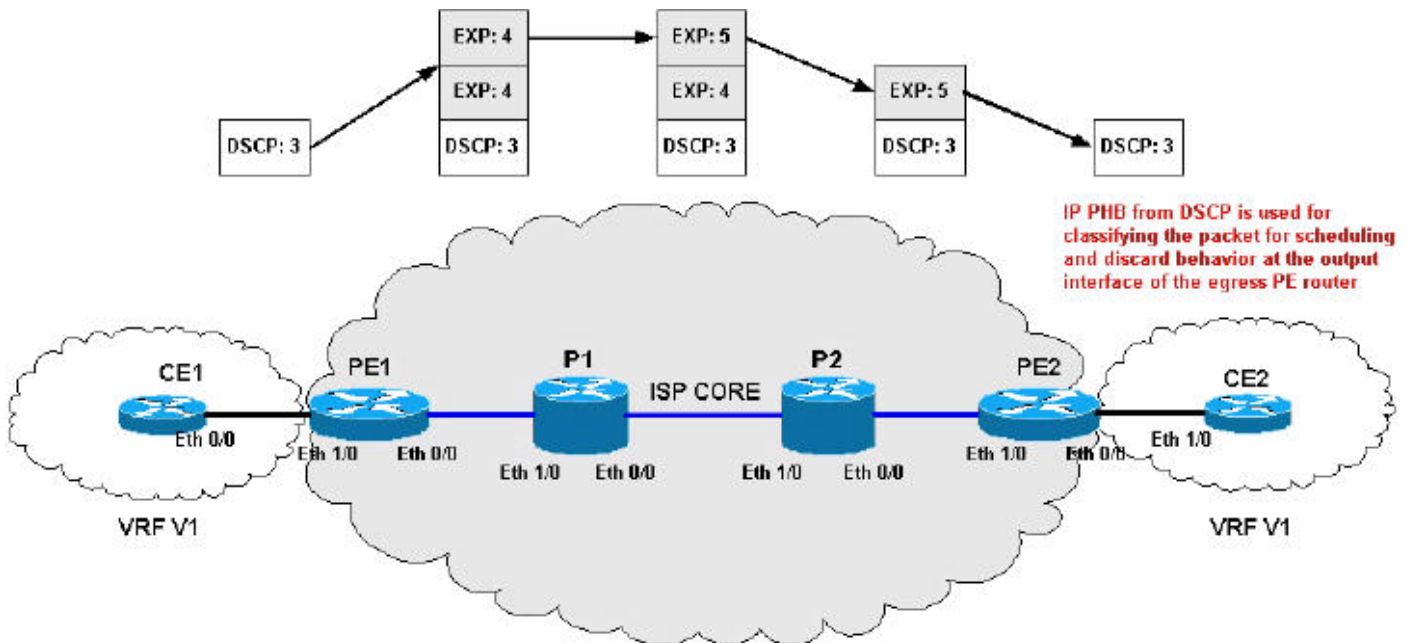
```

class MPLS-AF12
  set qos-group mpls experimental topmost
  set discard-class 1
class MPLS-AF21
  set qos-group mpls experimental topmost
  set discard-class 2
class MPLS-AF22
  set qos-group mpls experimental topmost
  set discard-class 3
class MPLS-AF31
  set qos-group mpls experimental topmost
  set discard-class 4
class MPLS-AF32
  set qos-group mpls experimental topmost
  set discard-class 5
!
policy-map qos-group-out
  class gold
    bandwidth 40
    random-detect discard-class-based
  class silver
    bandwidth 30
    random-detect discard-class-based
  class bronze
    bandwidth 20
    random-detect discard-class-based
!
interface Ethernet0/0
  ip vrf forwarding v1
  ip address 10.0.1.1 255.255.255.0
  max-reserved-bandwidth 90
  service-policy output qos-group-out
!
interface Ethernet1/0
  ip address 192.168.2.2 255.255.255.0
  service-policy input qos-group-in
  tag-switching ip
!

```

## Mode de Court-canal

Le mode de Court-canal de Tunnellisation de DiffServ utilise les mêmes règles et techniques à travers le noyau. La différence est au routeur PE de sortie (PE2) — vous classifiez les paquets IP nouvellement exposés pour la Mise en file d'attente sortante basée sur l'IP PHB de la valeur DSCP de ce paquet IP.



configurations de mode de Court-canal :

```

PE1

!--- On input, the IP Precedent is copied to the MPLS
EXP !--- on label imposition. When leaving the router,
the !--- MPLS EXP bits are used to classify the traffic
into three !--- traffic classes—gold, silver and bronze
(the topmost). class-map match-all IP-AF22 match ip
precedence 0 1 class-map match-all IP-AF31 match ip
precedence 2 3 class-map match-all IP-AF32 match ip
precedence 4 5 ! class-map match-all bronze match mpls
experimental topmost 2 3 class-map match-all silver
match mpls experimental topmost 4 class-map match-all
gold match mpls experimental topmost 5 ! policy-map set-
MPLS-PHB
  class IP-AF22
    police 12000 conform-action set-mpls-exp-imposition-
transmit
      3 exceed-action set-mpls-exp-imposition-
transmit 2
  class IP-AF31
    police 12000 conform-action set-mpls-exp-imposition-
transmit
      4 exceed-action set-mpls-exp-imposition-
transmit 3
  class IP-AF32
    police 12000 conform-action set-mpls-exp-imposition-
transmit
      5 exceed-action set-mpls-exp-imposition-
transmit 4
!
policy-map output-qos
  class gold
    bandwidth 40
    random-detect
  class silver
    bandwidth 30
    random-detect
  class bronze

```

```

bandwidth 20
random-detect
!
interface Ethernet0/0
ip address 192.168.0.1 255.255.255.0
max-reserved-bandwidth 90
service-policy output output-qos
tag-switching ip
!
interface Ethernet1/0
ip vrf forwarding v1
ip address 10.0.0.2 255.255.255.0
service-policy input set-MPLS-PHB
!

```

## P1

```

!--- This configuration swaps the top label from 4 to 5,
!--- which does not need to occur to follow the previous
configuration. class-map match-all mpls-in match mpls
experimental topmost 4 ! policy-map mpls-in
  class mpls-in
    set mpls experimental topmost 5
!
interface Ethernet0/0
ip address 192.168.1.1 255.255.255.0
tag-switching ip
!
interface Ethernet1/0
ip address 192.168.0.2 255.255.255.0
service-policy input mpls-in
tag-switching ip
!

```

## P2

```

!--- Remember to copy down the MPLS EXP value from !---
the newly exposed label after the PHP. class-map match-
all MPLS-AF11 match mpls experimental topmost 0 class-
map match-all MPLS-AF12 match mpls experimental topmost
1 class-map match-all MPLS-AF21 match mpls experimental
topmost 2 class-map match-all MPLS-AF22 match mpls
experimental topmost 3 class-map match-all MPLS-AF31
match mpls experimental topmost 4 class-map match-all
MPLS-AF32 match mpls experimental topmost 5 ! class-map
match-all gold match qos-group 5 class-map match-all
silver match qos-group 4 class-map match-all bronze
match qos-group 3 match qos-group 2 ! policy-map qos-
group-in
  class MPLS-AF11
    set qos-group mpls experimental topmost
  class MPLS-AF12
    set qos-group mpls experimental topmost
  class MPLS-AF21
    set qos-group mpls experimental topmost
  class MPLS-AF22
    set qos-group mpls experimental topmost
  class MPLS-AF31
    set qos-group mpls experimental topmost
  class MPLS-AF32
    set qos-group mpls experimental topmost

```

```

!
policy-map qos-group-out
  class gold
    bandwidth 40
    random-detect
    set mpls experimental topmost qos-group
  class silver
    bandwidth 30
    random-detect
    set mpls experimental topmost qos-group
  class bronze
    bandwidth 20
    random-detect
    set mpls experimental topmost qos-group
!
interface Ethernet0/0
  ip address 192.168.2.1 255.255.255.0
  max-reserved-bandwidth 90
  service-policy output qos-group-out
  tag-switching ip
!
interface Ethernet1/0
  ip address 192.168.1.2 255.255.255.0
  service-policy input qos-group-in
  tag-switching ip
!

```

## PE2

```

!--- Remember to queue the newly exposed IP packet !---
based on the value of the IP Precedent. class-map match-
all gold match precedence 4 5 class-map match-all silver
match precedence 2 3 class-map match-all bronze match
precedence 0 1 ! policy-map qos-group-out
  class gold
    bandwidth 40
    random-detect prec-based
  class silver
    bandwidth 30
    random-detect prec-based
  class bronze
    bandwidth 20
    random-detect prec-based
!
interface Ethernet0/0
  ip vrf forwarding v1
  ip address 10.0.1.1 255.255.255.0
  max-reserved-bandwidth 90
  service-policy output qos-group-out
!
interface Ethernet1/0
  ip address 192.168.2.2 255.255.255.0
  tag-switching ip
!

```

## Résumé de tunnel mode

Cette table récapitule les diverses actions qui se sont appliquées à l'IP ou aux paquets étiquetés à de diverses étapes dans le réseau :

Mode de perçage d'un tunnel	IP -> étiquette	Étiquette -> étiquette	Étiquette -> IP
Uniforme	Copiez IP Prec/DiffServ dans l'EXP MPLS (peut être changé par le fournisseur de services également)	L'EXP MPLS peut être changé par le fournisseur de services	L'EXP MPLS a copié sur IP Prec/DiffServ
Canal	EXP MPLS réglé par la stratégie QoS de fournisseur de services		IP d'original Prec/DiffServ préservé (Mise en file d'attente de sortie basée sur l'EXP MPLS)
Cour-canal		IP d'original Prec/DiffServ préservé (Mise en file d'attente de sortie basée dans IP Prec/DiffServ)	

## Informations connexes

- [Logiciel de Cisco IOS - MPLS](#)
- [&A Q - Qualité de service pour des réseaux de Commutation multiprotocole par étiquette](#)
- [Services de commutation Cisco IOS référence de commandes, version 12.2](#)
- [Solutions guide de configuration de Qualité de service Cisco IOS, version 12.2](#)
- [Solutions référence de commandes de Qualité de service Cisco IOS, version 12.2](#)
- [MPLS Support Page](#)
- [Page d'assistance QoS](#)
- [Support technique - Cisco Systems](#)