

Modes de tunnellation DiffServ pour les réseaux MPLS

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Informations générales](#)

[Comportement par défaut](#)

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

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

[Modes de tunnellation DiffServ](#)

[Mode unifié](#)

[Mode Tuyau](#)

[Mode court-circuit](#)

[Résumé du mode tunnel](#)

[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 réseaux privés virtuels (VPN)
- Concepts relatifs à la priorité IP, au type de service (ToS) et à DiffServ
- Marquage et classification des paquets QoS (Quality of Service) à l'aide de l'interface de ligne de commande (CLI) QoS modulaire

[Components Used](#)

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

logiciel suivantes :

- Logiciel Cisco IOS® Version 12.2(13)T pour les nouvelles fonctionnalités QoS. La version 12.1(5)T inclut les fonctionnalités QoS d'origine.
- Tout routeur Cisco de la gamme 3600 ou ultérieure, tel que le Cisco 3660 ou 7206, qui prend en charge la fonctionnalité de routeur de périphérie (PE) du fournisseur MPLS.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

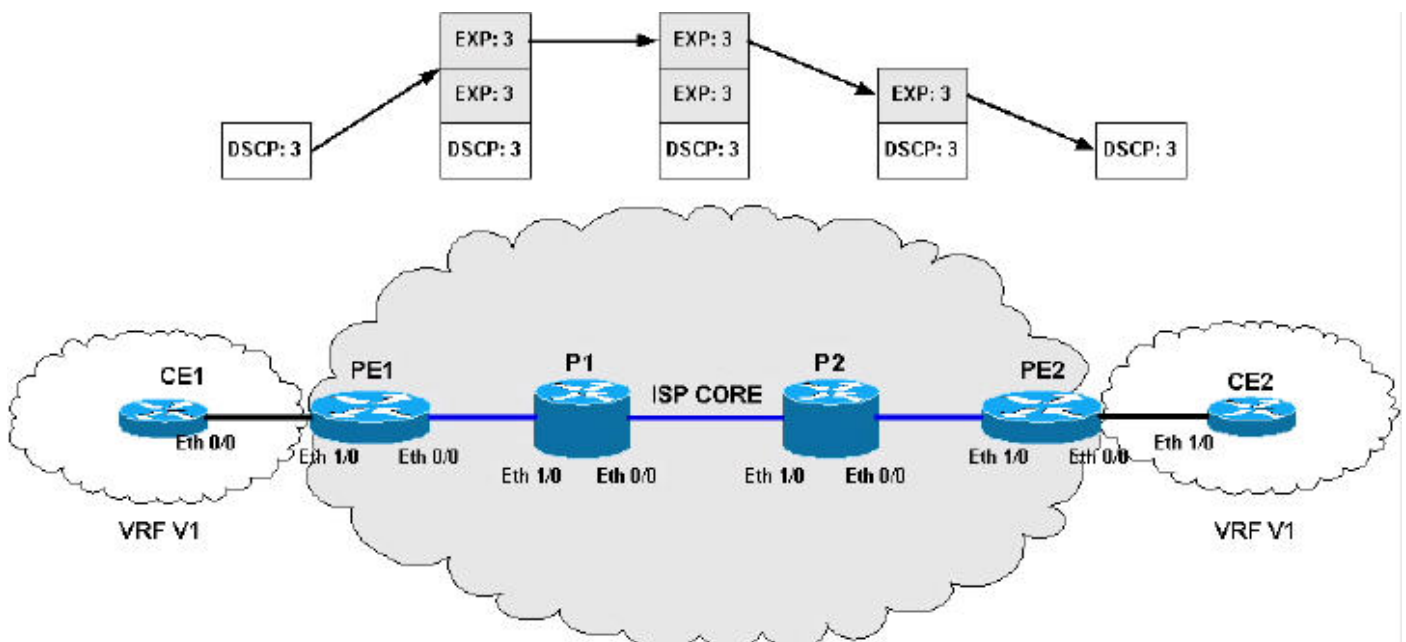
Conventions

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 EXP (DiffServ Code Point)/MPLS expérimentaux (EXP) lorsqu'un paquet circule d'un routeur de périphérie client à un autre routeur CE sur un coeur MPLS :



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

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

- La priorité IP du paquet IP entrant est copiée sur les bits EXP MPLS de toutes les étiquettes poussées.
- Les 1st trois bits du bit DSCP sont copiés sur les bits EXP MPLS de toutes les étiquettes poussées.
- Cette technique est également appelée réflexion ToS.

Transfert MPLS (Étiquette -> Étiquette) :

- L'EXP est copié sur les nouvelles étiquettes qui sont échangées/poussées pendant le transfert ou l'imposition.
- Lors de l'imposition de l'étiquette, les étiquettes sous-jacentes ne sont pas modifiées avec la valeur de la nouvelle étiquette ajoutée à la pile d'étiquettes actuelle.
- Lors de la disposition de l'étiquette, les bits EXP ne sont pas copiés vers les bits EXP de l'étiquette nouvellement exposés.

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

- Lors de la disposition de l'étiquette, les bits EXP ne sont pas copiés dans le champ de priorité IP/DSCP du paquet IP nouvellement exposé.

Utilisation et comportement des commandes du logiciel IOS 12.2(13)T antérieur à Cisco

Avant la version 12.2(13)T de l'IOS, la commande [set mpls expérimental](#) était la seule méthode disponible pour modifier les bits EXP MPLS.

Utilisation et comportement des commandes de la version 12.2(13)T du logiciel Cisco IOS

Avec IOS version 12.2(13)T et ultérieure, la commande **set mpls expérimental** est modifiée pour autoriser ces options :

- [set mpls expérimental top {mpls-exp-value | qos-group \[table table-map-name\]}](#)
- [set mpls application expérimentale {mpls-exp-value | qos-group \[table table-map-name\]}](#)

Note : La nouvelle commande **set mpls expérimental top** est équivalente à l'ancienne commande **set mpls imposition**.

Ces deux commandes, combinées à quelques nouveaux commutateurs de commande, permettent de mieux contrôler la manipulation des bits EXP MPLS lors des opérations push, swap et pop d'étiquette. Ces deux commandes vous permettent d'utiliser des modes de tunnellation DiffServ.

Modes de tunnellation DiffServ

Les modes de tunnellation Diffserv introduisent un nouveau PHB (Per-Hop-Behavior), qui permet une QoS différenciée dans un réseau de fournisseurs. Le mode de tunnellation est défini à la périphérie du réseau, généralement dans les routeurs de commutation d'étiquette de PE (LSR) (en entrée et en sortie). Vous devrez peut-être apporter des modifications aux routeurs P ; vous devez également tenir compte de ce qui se passe lorsque l'étiquette la plus élevée est supprimée d'un paquet en raison de Penult-Hop-Popping (PHP). Il peut être nécessaire de copier la valeur EXP MPLS de l'étiquette supérieure qui est insérée dans l'étiquette nouvellement exposée ; cela ne s'applique pas toujours à tous les modes de transmission tunnel.

Dans certains cas (par exemple, un réseau MPLS non VPN), l'action PHP sur le routeur P final peut exposer un paquet IP ordinaire quand un paquet avec une seule étiquette est reçu. Lorsque ce paquet IP est reçu par le LSR de sortie (PE), il n'est pas possible de classer le paquet en fonction des bits EXP MPLS, car il n'y a pas d'étiquette maintenant. Dans ces situations, vous devez configurer le routeur PE de sortie pour annoncer une étiquette **explicite-nulle**. Lorsque

l'action PHP est exécutée sur le routeur P, une étiquette avec une valeur de zéro est envoyée, et avec cette étiquette spéciale vous pouvez marquer les bits EXP comme paquets normalement étiquetés, permettant ainsi la classification correcte sur le routeur PE de sortie.

La prise en charge du réseau MPLS de la spécification Diffserv définit les modes de transmission tunnel suivants :

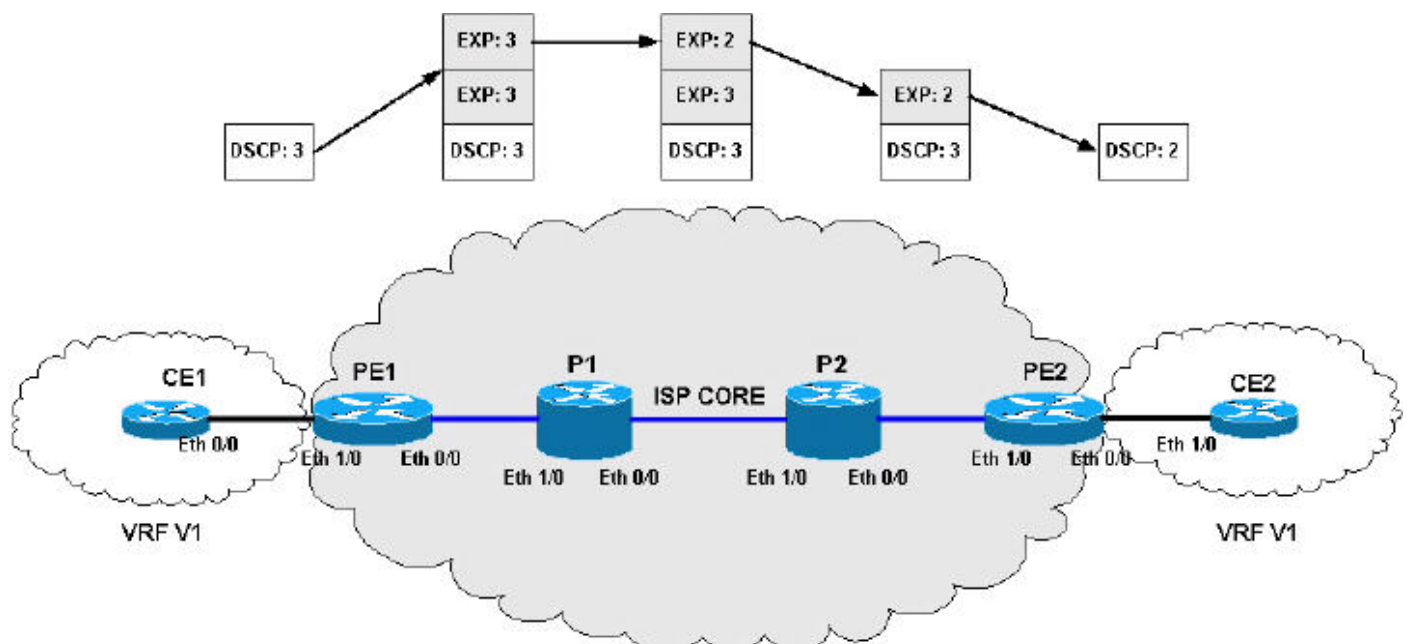
- [Uniforme](#)
- [Tuyau](#)
- [Tronçon court](#)

Les sections suivantes examinent chaque mode de transmission tunnel séparément et fournissent des exemples pour montrer comment chaque mode de transmission tunnel peut être configuré. Les exemples incluent un mappage complet de la priorité IP aux bits EXP MPLS. Il est possible d'avoir plusieurs paramètres QoS et modes de tunnellation différents pour chaque client.

Remarque : Les exemples de configuration ne sont pas spécifiques au VPN MPLS et s'appliquent aux réseaux MPLS simples et aux réseaux CsC (Carrier Support). Il est également possible que votre réseau puisse varier d'un autre réseau : de nombreux paramètres QoS et modes de tunnellation différents peuvent être utilisés.

Mode unifié

Le mode uniforme de tunnellation DiffServ ne comporte qu'une seule 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 EXP MPLS des étiquettes imposées. Lorsque les bits EXP traversent le cœur, ils peuvent ou non être modifiés par des routeurs P intermédiaires. Dans cet exemple, le routeur P1 modifie les bits EXP de l'étiquette supérieure. Au niveau du routeur P de sortie (P2), nous copions les bits EXP dans les bits EXP de l'étiquette nouvellement exposée après le PHP (Penlatter-Hop-Pop). Enfin, au niveau du routeur PE de sortie (PE2), nous copions les bits EXP vers les bits DSCP du paquet IP nouvellement exposé.



Configurations en mode unifié :

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
!

```

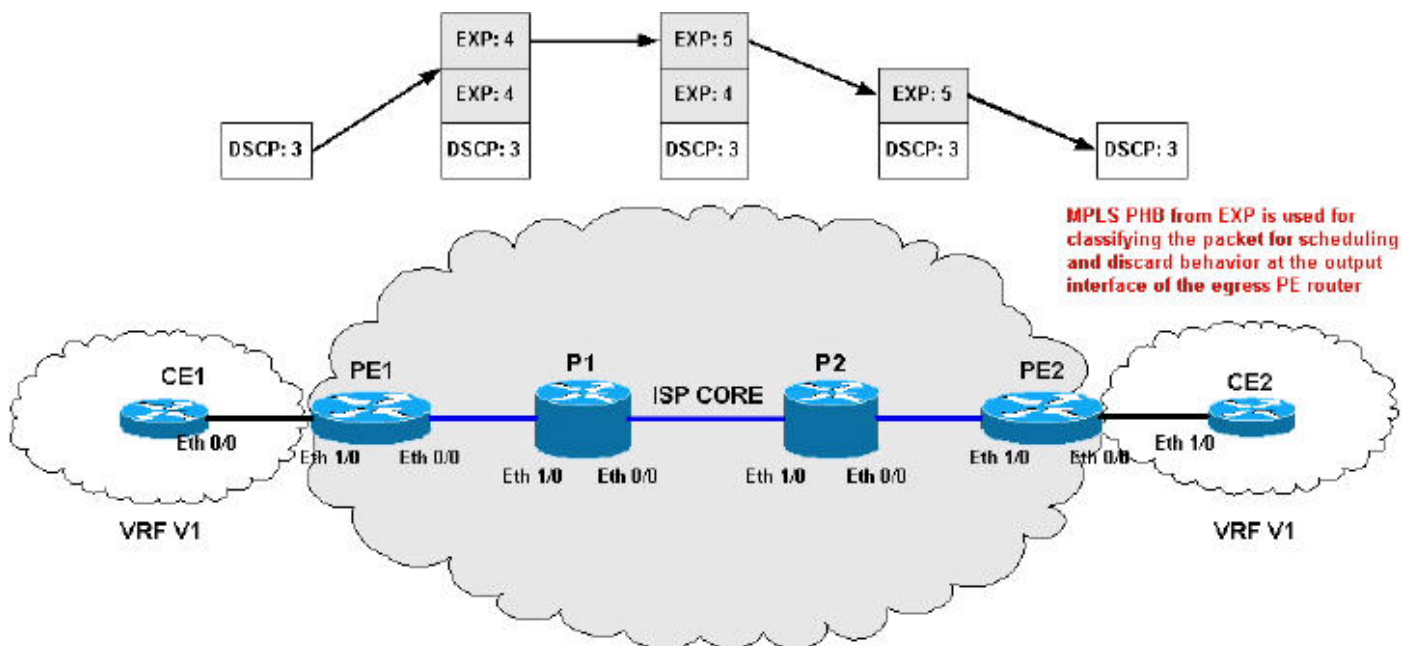
Mode Tuyau

Le mode de tuyauterie de tunnellation Diffserv utilise deux couches de QoS :

1. QoS sous-jacente pour les données, qui reste inchangée lors de la traversée du coeur.
2. Une QoS par coeur, qui est séparée de celle des paquets IP sous-jacents. Ce PHB QoS par coeur reste transparent pour les utilisateurs finaux.

Lorsqu'un paquet atteint la périphérie du coeur MPLS, le routeur PE2 (E/S de sortie) classe les paquets IP nouvellement exposés pour la mise en file d'attente sortante en fonction du PHB MPLS

des bits EXP de l'étiquette récemment supprimée.



Configurations du mode Tuyau :

```

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

```

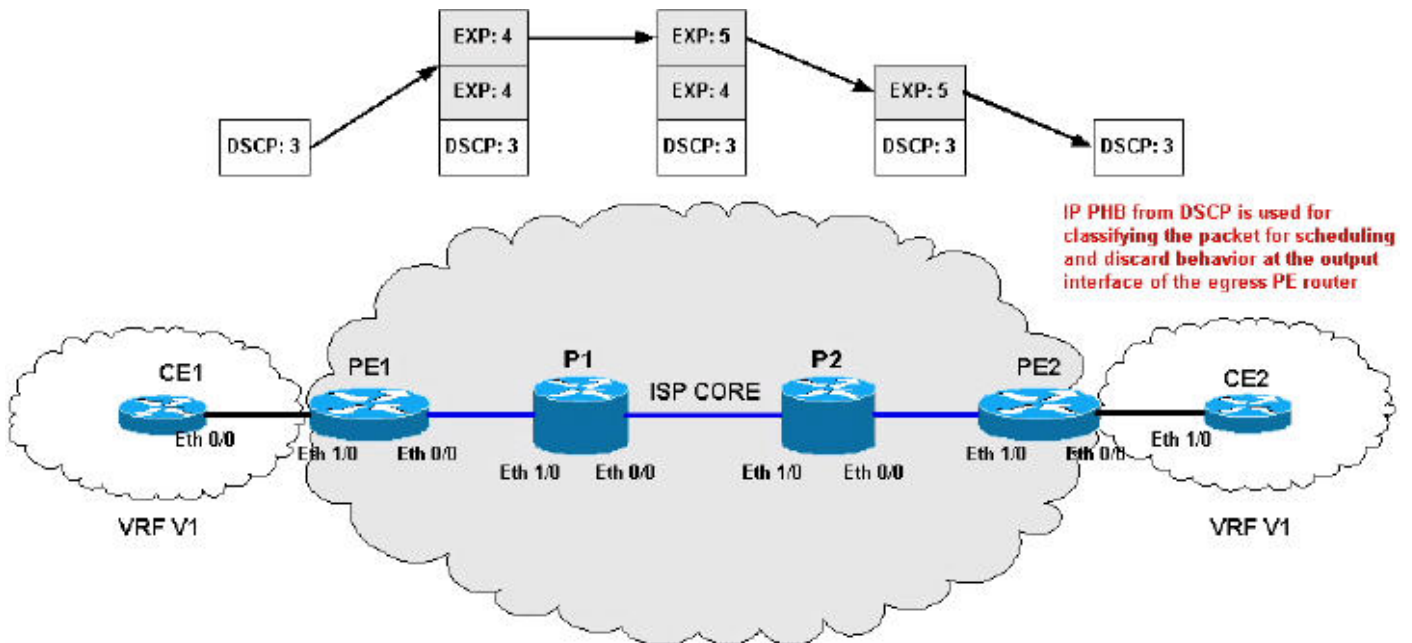
```

    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 court-circuit

Le mode tramway Diffserv abrégé utilise les mêmes règles et techniques sur le coeur de réseau. La différence se situe au niveau du routeur PE de sortie (PE2) : vous classifiez les paquets IP nouvellement exposés pour la mise en file d'attente sortante en fonction de la valeur DSCP de ce paquet IP.



Configurations en mode court-circuit :

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 du mode tunnel

Ce tableau récapitule les différentes actions appliquées aux paquets IP ou étiquetés à différentes étapes du réseau :

Mod e Tun nelin g	Étiquette IP ->	Étiquett e -> Étiquett e	Étiquette -> IP
Unif orm e	Copier IP Δ/DiffServ dans MPLS EXP (peut également être modifié par le SP)	MPLS EXP peut être modifié par SP	EXP MPLS copié vers IP Δ/DiffServ
Tuy au	EXP MPLS défini par la stratégie QoS SP		IP d'origine, protégé/Diffserv (mise en file d'attente de sortie basée sur EXP MPLS)
Tron çon cour t			IP d'origine RP/Diffserv préservée (mise en file d'attente de sortie basée sur IP Δ/DiffServ)

Informations connexes

- [Logiciel Cisco IOS - MPLS](#)
- [Q&R - Qualité de service pour les réseaux de commutation par étiquette multiprotocole](#)
- [Référence des commandes des services de commutation Cisco IOS, version 12.2](#)
- [Guide de configuration des solutions de qualité de service Cisco IOS, version 12.2](#)
- [Référence des commandes des solutions de qualité de service Cisco IOS, version 12.2](#)
- [MPLS Support Page](#)
- [Page d'assistance QoS](#)
- [Support technique - Cisco Systems](#)