

# DiffServ-tunneling voor MPLS-netwerken

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[Standaardgedrag](#)

[Vooraf Cisco IOS-software release 12.2\(13\)T, opdrachtgebruik en -gedrag](#)

[Post Cisco IOS-software release 12.2\(13\)T, opdrachtgebruik en gedrag](#)

[DiffServ-tunneling](#)

[Uniforme modus](#)

[Boormodus](#)

[Snelbuis](#)

[Samenvatting van tunnelmodus](#)

[Gerelateerde informatie](#)

## Inleiding

In dit document wordt de implementatie van Distributed Services (DiffServ) Tunneling-methoden beschreven die beschikbaar zijn voor MPLS-gebaseerde netwerkomgevingen (Multiprotocol Label Switching).

## Voorwaarden

### Vereisten

Lezers van dit document zouden kennis moeten hebben van deze onderwerpen:

- MPLS en MPLS voor Virtual Private Networks (VPN's)
- Concepten met betrekking tot IP-voorrang, type service (ToS) en DiffServ
- QoS-pakketmarkering (Quality of Service) en -classificatie met behulp van modulaire QoS-opdrachtregeel-CLI (MQC)

### Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Cisco IOS-<sup>software</sup>release 12.2(13)T voor nieuwe QoS-functies. release 12.1(5)T bevat

oorspronkelijke QoS-functies.

- Elke Cisco-router van de 3600-serie of later, zoals Cisco 3660 of 7206 die MPLS-router (P)/provider-routerfunctionaliteit (PE) ondersteunt.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

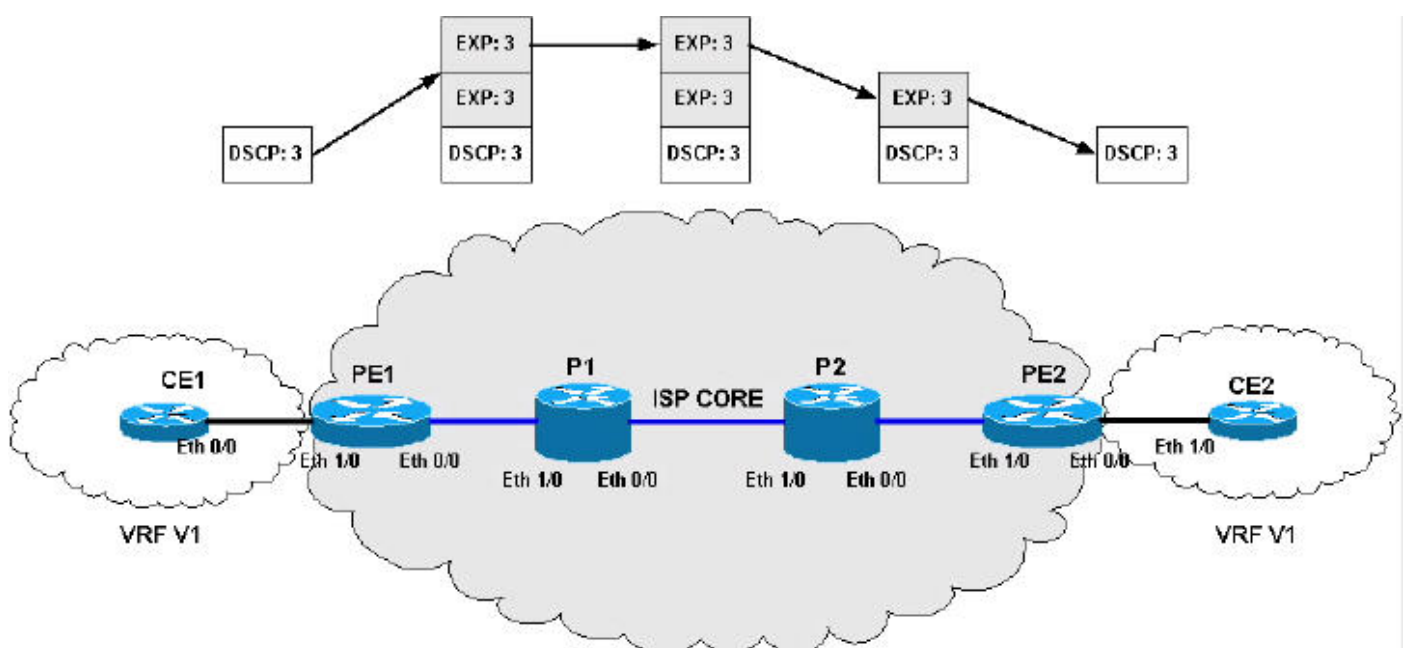
## Conventies

Raadpleeg [Cisco Technical Tips Conventions](#) (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

## Achtergrondinformatie

### Standaardgedrag

In dit diagram wordt het standaardgedrag beschreven van de DiffServ Code Point (DSCP)/MPLS Experimental (EXP)-bits als een pakketreis van één Customer Edge (CE) router naar een andere CE-router via een MPLS-kern:



In dit gedeelte wordt de activiteit beschreven in het standaardgedragsschema.

Opleggen van het label (IP -> Label):

- De IP-voorrang van het inkomende IP-pakket wordt gekopieerd naar de MPLS EXP-bits van alle geduwde label(s).
- De 1<sup>ste</sup> drie bits van het DSCP-bit worden gekopieerd naar de MPLS EXP-bits van alle geduwde label(s).
- Deze techniek staat ook bekend als ToS Reflectie.

MPLS-verzending (Label -> Label):

- EXP wordt gekopieerd naar de nieuwe etiketten die tijdens het verzenden of het opleggen worden geruild/geduwd.
- Bij het instellen van het label worden de onderliggende labels niet aangepast en wordt de waarde van het nieuwe label toegevoegd aan de huidige labelstack.
- Bij labeldispositie worden de EXP bits niet gekopieerd naar de nieuw blootgestelde EXP-bits.

Verwijdering van het label (Label -> IP):

- Bij labelverwerking worden de EXP-bits niet gekopieerd naar het veld IP-voorrang/DSCP van het nieuwe blootgestelde IP-pakket.

## Vooraf Cisco IOS-software release 12.2(13)T, opdrachtgebruik en -gedrag

Vóór IOS release 12.2(13)T was de ingestelde MPLS experimentele opdracht de enige methode die beschikbaar was om de MPLS EXP-bits te wijzigen.

## Post Cisco IOS-software release 12.2(13)T, opdrachtgebruik en gedrag

Met IOS release 12.2(13)T en hoger wordt de **ingestelde** experimentele opdracht MPLS aangepast om deze opties mogelijk te maken:

- [mpls experimenteel bovenste {mpls-exp-waarde instellen | qos-groep \[tabel-kaart-naam\]}](#)
- [Stel mpls experimentele instelling {mpls-exp-waarde in | qos-groep \[tabel-kaart-naam\]}](#)

**Opmerking:** de nieuwe **set mpls experimentele hoogste** opdracht is gelijk aan de oude **set mpls** opdracht.

Deze twee opdrachten, in combinatie met een aantal nieuwe commando switches, maken het mogelijk de MPLS EXP bits manipulatie te controleren tijdens labeldruk, swap en popbewerkingen. Met deze twee opdrachten kunt u DiffServ-tunneling gebruiken.

## DiffServ-tunneling

DiffServ Tunneling Modes introduceert een nieuw Per-Hop-Behavior (PHB), dat gedifferentieerde QoS in een providernetwerk toestaat. De tunneling-modus wordt gedefinieerd aan de rand van het netwerk, normaal gesproken in de LSR-switches (zowel in- als uitloop). U moet mogelijk wijzigingen aanbrengen in de IP-routers; U moet ook overwegen wat er gebeurt wanneer het bovenste label van een pakje wordt verwijderd door Penulent-Hop-Popping (PHP). Het kan nodig zijn de MPLS EXP-waarde te kopiëren van het bovenlabel dat op het nieuwe blootgestelde etiket wordt gedrukt; dit geldt niet altijd voor alle tunnelmodi .

In sommige gevallen (bijvoorbeeld een gewoon niet-VPN MPLS netwerk) kan de PHP actie op de finale IP router een gewoon IP-pakket blootstellen wanneer er een pakket met slechts één label wordt ontvangen. Wanneer dit IP-pakket door de fractionele LSR (PE) wordt ontvangen, is het niet mogelijk om het pakket op basis van de MPLS EXP-bits te classificeren, omdat er nu geen label is. In deze situaties, moet u de router van de stress PE configureren om een **expliciet-nul** etiket aan te geven. Wanneer de PHP actie op de P router wordt uitgevoerd, wordt er een label met een waarde van nul verstuurd, en met dit speciale label kunt u de EXP bits markeren zoals normaal geëtiketteerde pakketten, wat de juiste classificatie op de IP-router toelaat.

De MPLS-netwerkondersteuning van DiffServ-specificatie definieert deze tunnelmodi:

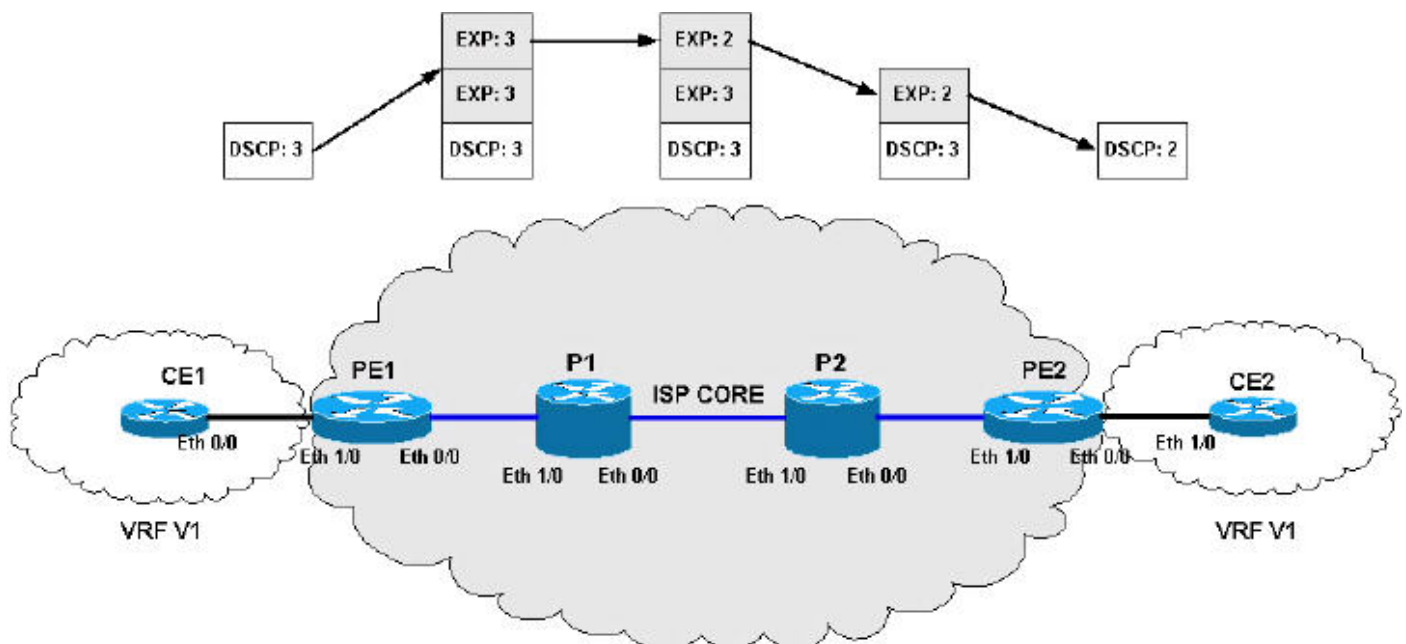
- [uniform](#)
- [buis](#)
- [korte buis](#)

In de volgende secties wordt elke tunneling-modus afzonderlijk onderzocht en worden voorbeelden gegeven om aan te tonen hoe elke tunneling-modus kan worden geconfigureerd. De voorbeelden omvatten een volledige afbeelding van IP-voorrang op MPLS EXP-bits. Het is mogelijk om voor elke klant een aantal verschillende QoS-parameters en tunneling-modi te hebben.

**Opmerking:** de configuratievoorbeelden zijn niet specifiek voor MPLS VPN en zijn van toepassing op gewone MPLS-netwerken en Carrier Support Carrier (CsC)-netwerken. Het is ook mogelijk dat uw netwerk van een ander netwerk kan variëren—veel verschillende QoS parameters en de tunneling modi kunnen worden gebruikt.

## Uniforme modus

DiffServ-tunneling in uniform model heeft slechts één laag van QoS, die end-to-end bereikt. De IP-router (PE1) kopieert de DSCP van het inkomende IP-pakket in de MPLS EXP-bits van de opgelegde labels. Aangezien de EXP bits door de kern reizen, kunnen ze al dan niet worden gewijzigd door intermediaire P routers. In dit voorbeeld, P router P1 wijzigt de EXP bits van het bovenste label. Bij de egress P router (P2) kopiëren we de EXP-bits naar de EXP-bits van het nieuwe blootgestelde label na de PHP PHP (PenFinal-Hop-Pop). Tenslotte kopiëren we op de GRE-router (PE2) de EXP-bits naar de DSCP-bits van het nieuw blootgestelde IP-pakket.



Standaardmodusconfiguraties:

```

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-gos
  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-gos

```

```
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
!

```

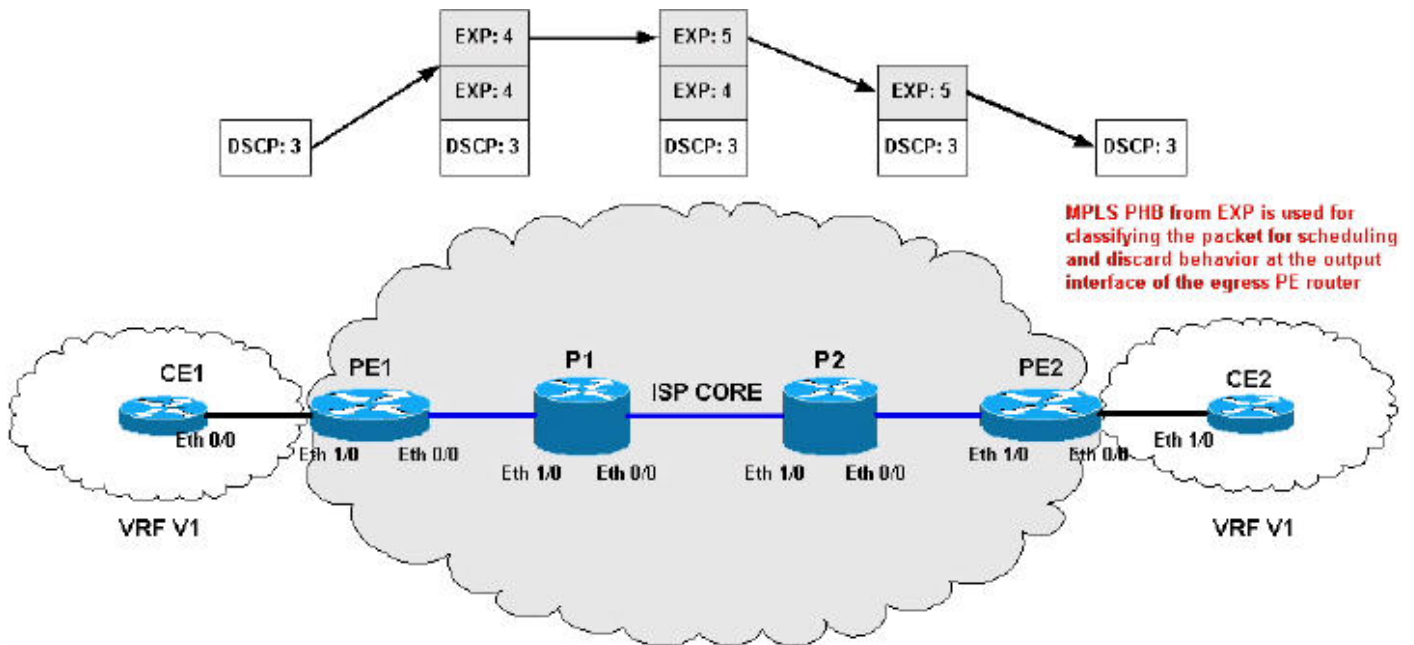
## Boormodus

In de Diffserv Tunneling Pipe-modus worden twee lagen QoS gebruikt:

1. Een onderliggende QoS voor de gegevens, die onveranderd blijven bij het verplaatsen van de kern.
2. Een QoS per-core, die gescheiden is van die van de onderliggende IP-pakketten. Deze per-core QoS PHB blijft transparant voor de eindgebruikers.

Wanneer een pakket de rand van de MPLS-kern bereikt, classificeert de egress PE-router (PE2) de nieuwe blootgestelde IP-pakketten voor uitgaande wachtrijen op basis van MPLS PHB van de EXP-bits van het recent verwijderde label.





Configuratie buis:

```

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 PHB. 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

```

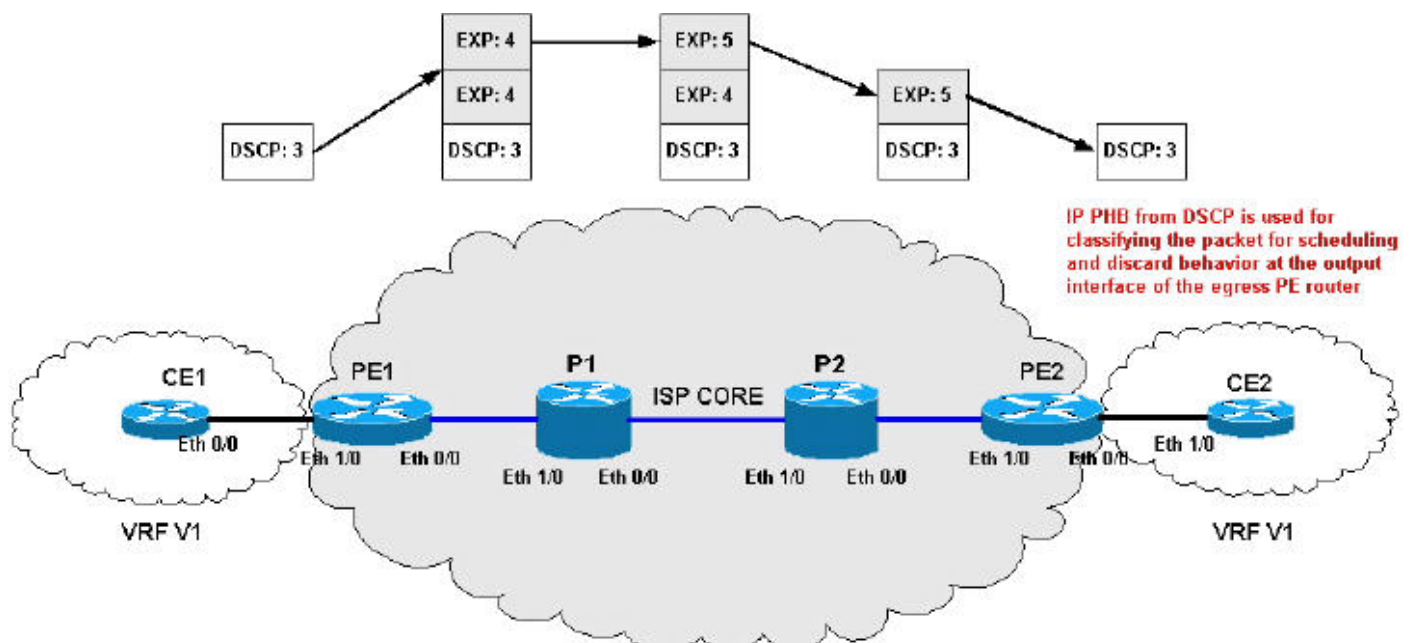
```

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
!

```

## Snelbuis

Diffserv Tunneling Short-pipe Mode gebruikt dezelfde regels en technieken in de kern. Het verschil is bij de GRE-router (PE2) - u classificeert de nieuwe blootgestelde IP-pakketten voor uitgaande wachtrijen op basis van IP PHB van de DSCP-waarde van dit IP-pakket.



## Configuratie van de kortpijpmodus:

### PE1

```
!--- On input, the IP Precedence 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
!
```

## Samenvatting van tunnelmodus

Deze tabel geeft een samenvatting van de verschillende acties die op IP-pakketten of geëtiketteerde pakketten in verschillende fasen in het netwerk zijn toegepast:

Tunneling-modus	IP -> Label	Label -> Label	Label -> IP
uniform	IP Prec/DiffServ kopiëren naar MPLS EXP (dit kan ook door de SP worden gewijzigd)	MPLS EXP kan via SP worden gewijzigd	MPLS EXP gekopieerd naar IP Prec/DiffServ
buis	MPLS EXP ingesteld door het SP QoS-beleid		Originele IP Prec/DiffServ-conserveermiddelen (wachtrij op basis van MPLS EXP)
korte buis			Originele IP Prec/DiffServ-

			conserveermiddelen (wachtrij op basis van IP Prec/DiffServ)
--	--	--	--

## Gerelateerde informatie

- [Cisco IOS-software - MPLS](#)
- [Q&A - Quality-of-Service voor multi-protocol labelswitchingnetwerken](#)
- [Cisco IOS-switching services opdracht, release 12.2](#)
- [Cisco IOS Quality-of-Service oplossingen Configuration, release 12.2](#)
- [Cisco IOS Quality-of-Service oplossingen - Opdrachtreferentie, release 12.2](#)
- [MPLS-ondersteuningspagina](#)
- [QoS-ondersteuningspagina](#)
- [Technische ondersteuning - Cisco-systemen](#)