

Modos de tunelamento DiffServ para Redes MPLS

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Informações de Apoio](#)

[Comportamento padrão](#)

[Comportamento e uso do comando anterior ao Cisco IOS Software 12.2\(13\)T](#)

[Pós-uso e comportamento do comando 12.2\(13\)T do software Cisco IOS](#)

[Modos de tunelamento DiffServ](#)

[Modo Uniforme](#)

[Modo Pipe](#)

[Modo de tubulação curta](#)

[Resumo do modo de túnel](#)

[Informações Relacionadas](#)

[Introduction](#)

Este documento descreve a implantação dos modos de tunelamento de Differentiated Services (Diffserv) disponíveis para ambientes de rede baseados em Multiprotocol Label Switching (MPLS).

[Prerequisites](#)

[Requirements](#)

Os leitores deste documento devem estar cientes destes tópicos:

- MPLS e MPLS para redes privadas virtuais (VPN)
- Conceitos relacionados a precedência de IP, Tipo de Serviço (ToS - Type of Service) e DiffServ
- Classificação e marcação de pacotes de qualidade de serviço (QoS) usando CLI de linha de comando de QoS modular (MQC)

[Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- Software Cisco IOS® versão 12.2(13)T para novos recursos de QoS. A versão 12.1(5)T inclui recursos originais de QoS.
- Qualquer roteador Cisco da série 3600 ou posterior, como o Cisco 3660 ou 7206 que oferece suporte à funcionalidade do roteador de borda de provedor/roteador (PE) 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.

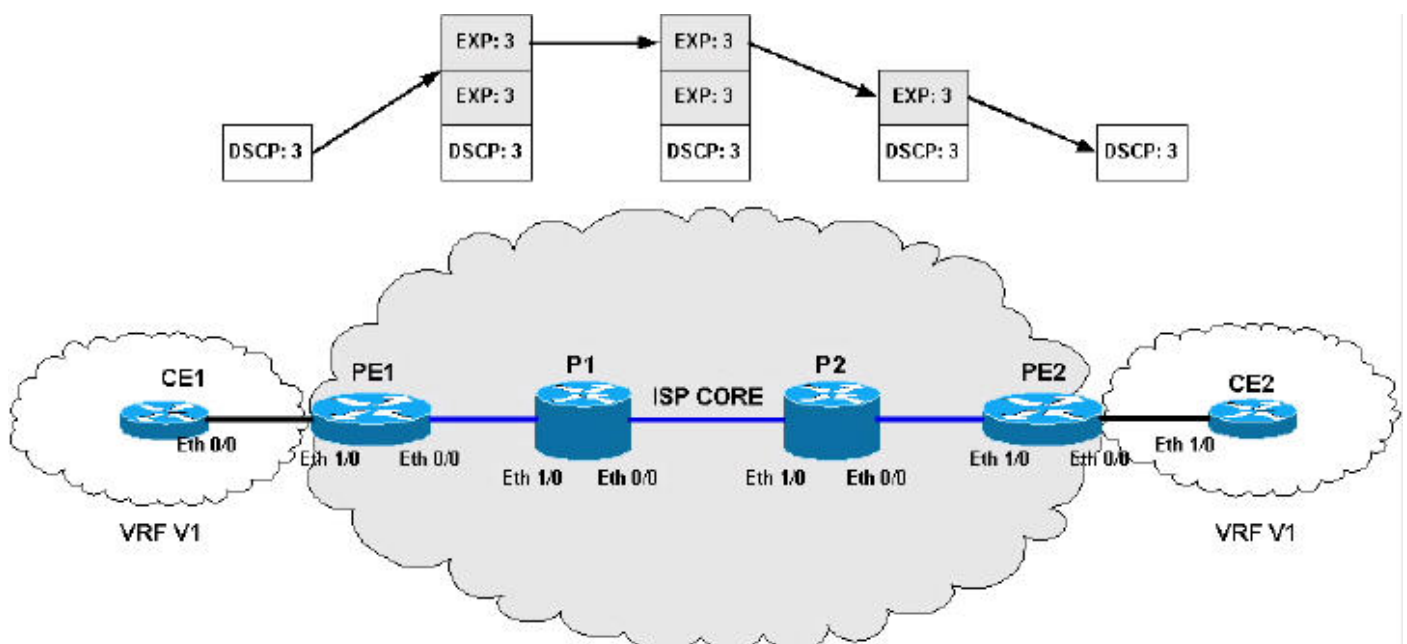
Conventions

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

Informações de Apoio

Comportamento padrão

Este diagrama descreve o comportamento padrão dos bits do Ponto de Código do DiffServ (DSCP)/MPLS Experimental (EXP) à medida que um pacote trafega de um roteador de borda do cliente (CE) para outro roteador CE através de um núcleo MPLS:



O restante desta seção descreve a atividade no diagrama Comportamento padrão.

Imposição do rótulo (IP -> Rótulo):

- A precedência de IP do pacote IP recebido é copiada para os bits EXP MPLS de todos os rótulos enviados.
- Os 1º bit de DSCP são copiados para os bits de MPLS EXP de todos os rótulos enviados.
- Essa técnica também é conhecida como Reflexão de ToS.

Encaminhamento de MPLS (Rótulo -> Rótulo):

- O EXP é copiado para os novos rótulos que são trocados/enviados durante o encaminhamento ou imposição.
- Na imposição do rótulo, os rótulos subjacentes não são modificados com o valor do novo rótulo sendo adicionado à pilha de rótulos atual.
- À disposição do rótulo, os bits EXP não são copiados para os bits EXP do rótulo recentemente exposto.

Disposição do rótulo (Rótulo -> IP):

- À disposição do rótulo, os bits EXP não são copiados para o campo IP precedence/DSCP do pacote IP recém-exposto.

Comportamento e uso do comando anterior ao Cisco IOS Software 12.2(13)T

Antes do IOS versão 12.2(13)T, o comando [set mpls experimental](#) era o único método disponível para modificar os bits EXP MPLS.

Pós-uso e comportamento do comando 12.2(13)T do software Cisco IOS

Com o IOS versão 12.2(13)T e posterior, o comando **set mpls experimental** é modificado para permitir estas opções:

- [set mpls experimental top more {mpls-exp-value | qos-group \[table table-map-name\]}](#)
- [definir imposição experimental de mpls {mpls-exp-value | qos-group \[table table-map-name\]}](#)

Observação: o novo comando **set mpls experimental top** é equivalente ao comando antigo **set mpls import**.

Esses dois comandos, em combinação com alguns novos switches de comando, permitem um melhor controle da manipulação de bits MPLS EXP durante as operações de envio, troca e pop de rótulo. Esses dois comandos permitem usar os Modos de Tunelamento DiffServ.

Modos de tunelamento DiffServ

Os Modos de Encapsulamento Diffserv apresentam um novo Comportamento por Salto (PHB - Per-Hop-Behavior), que permite QoS diferenciada em uma rede de provedores. O modo de tunelamento é definido na borda da rede, normalmente nos roteadores de switch de rótulo PE (LSRs) (tanto de entrada quanto de saída). Talvez seja necessário fazer alterações nos roteadores P; você também deve considerar o que ocorre quando o rótulo mais alto é removido de um pacote devido ao Penúltimos-Hop-Popping (PHP). Pode ser necessário copiar o valor MPLS EXP do rótulo superior que está sendo aberto para o rótulo recém-exposto; isso nem sempre se aplica a todos os modos de tunelamento.

Em alguns casos (por exemplo, uma rede MPLS não VPN simples), a ação do PHP no roteador P final pode expor um pacote IP simples quando um pacote com apenas um rótulo é recebido. Quando esse pacote IP é recebido pelo LSR de saída (PE), não é possível classificar o pacote com base nos bits EXP MPLS porque não há rótulo agora. Nessas situações, você deve configurar o roteador PE de saída para anunciar um rótulo **explícito nulo**. Quando a ação do PHP é executada no roteador P, um rótulo com um valor zero é enviado, e com este rótulo especial você pode marcar os bits EXP como pacotes normalmente rotulados, permitindo a classificação correta no roteador PE de saída.

O suporte de rede MPLS da especificação Diffserv define estes modos de tunelamento:

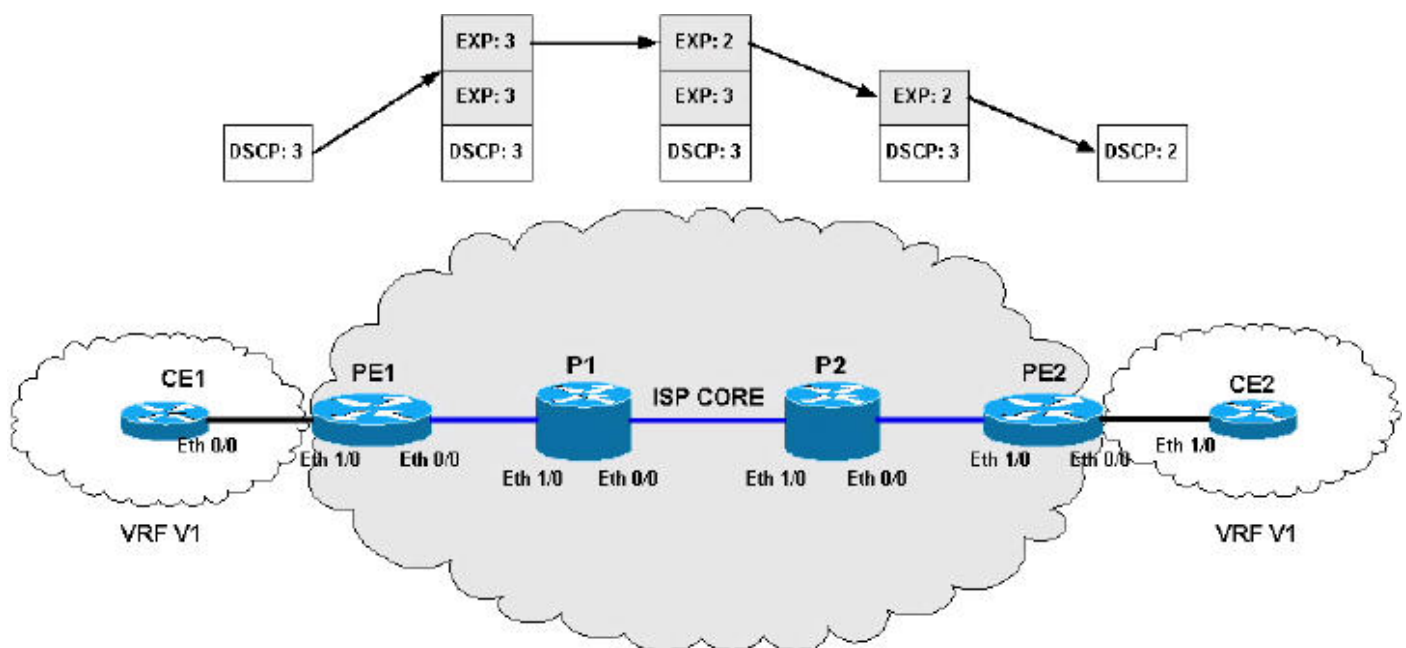
- [Uniforme](#)
- [Cano](#)
- [Cano curto](#)

As próximas seções examinam cada modo de tunelamento separadamente e fornecem exemplos para mostrar como cada modo de tunelamento pode ser configurado. Os exemplos incluem um mapeamento completo da precedência de IP para bits EXP MPLS. É possível ter vários parâmetros de QoS e modos de tunelamento diferentes para cada cliente.

Observação: os exemplos de configuração não são específicos para VPN MPLS e são aplicáveis para redes MPLS simples e redes CsC (Carrier supported Carrier). Também é possível que sua rede possa variar de outra rede—muitos parâmetros de QoS e modos de tunelamento diferentes podem ser usados.

[Modo Uniforme](#)

O Modo Uniforme de Tunelamento DiffServ tem apenas uma camada de QoS, que alcança de ponta a ponta. O roteador PE de entrada (PE1) copia o DSCP do pacote IP de entrada para os bits EXP MPLS dos rótulos impostos. À medida que os bits EXP trafegam pelo núcleo, eles podem ou não ser modificados por roteadores IP intermediários. Neste exemplo, o roteador P1 modifica os bits EXP do rótulo superior. No roteador P de saída (P2), copiamos os bits EXP para os bits EXP do novo rótulo exposto após o PHP (Penúltimos-Saltos-Pop). Finalmente, no roteador PE de saída (PE2), copiamos os bits EXP para os bits DSCP do pacote IP recém-exposto.



Configurações do modo uniforme:

```

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-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
!

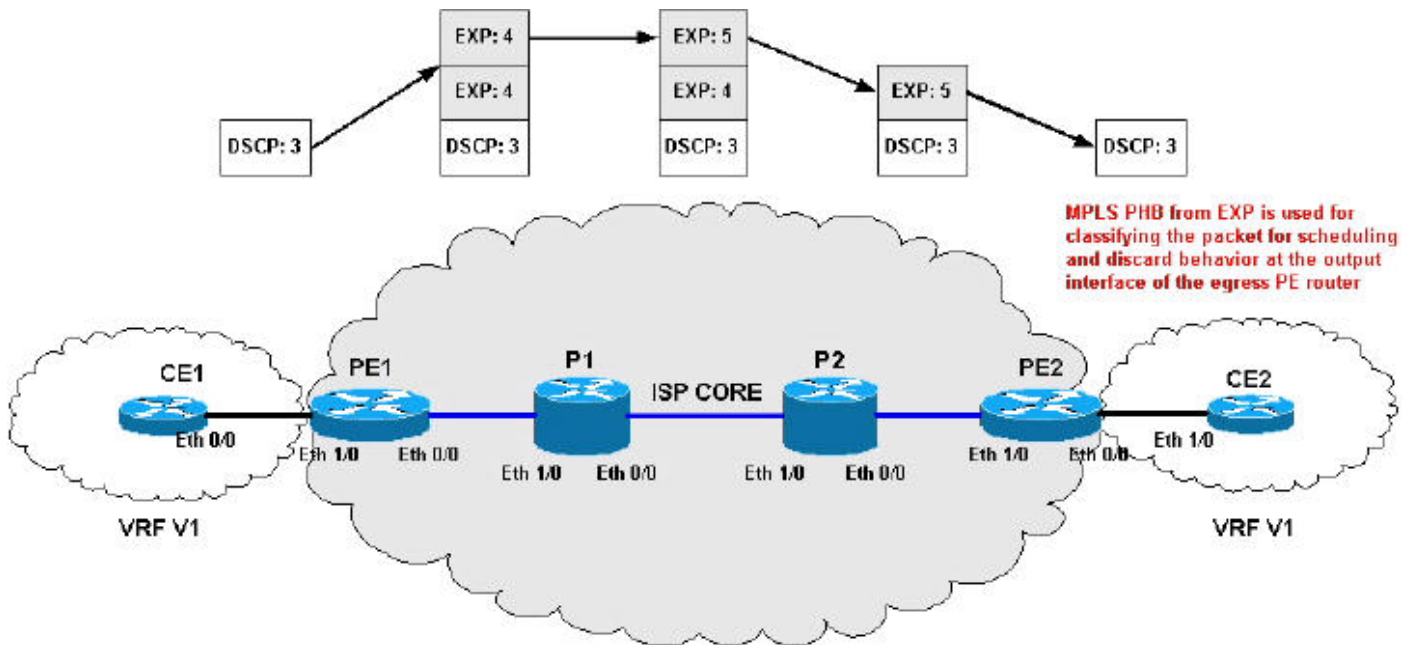
```

Modo Pipe

O modo Pipe de Tunelamento Diffserv usa duas camadas de QoS:

1. Uma QoS subjacente para os dados, que permanece inalterada ao atravessar o núcleo.
2. Uma QoS por núcleo, separada da dos pacotes IP subjacentes. Esse PHB QoS por núcleo permanece transparente para os usuários finais.

Quando um pacote chega à borda do núcleo MPLS, o roteador PE de saída (PE2) classifica os pacotes IP recentemente expostos para enfileiramento de saída com base no PHB MPLS dos bits EXP do rótulo recentemente removido.



Configurações do modo Pipe:

```

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

```

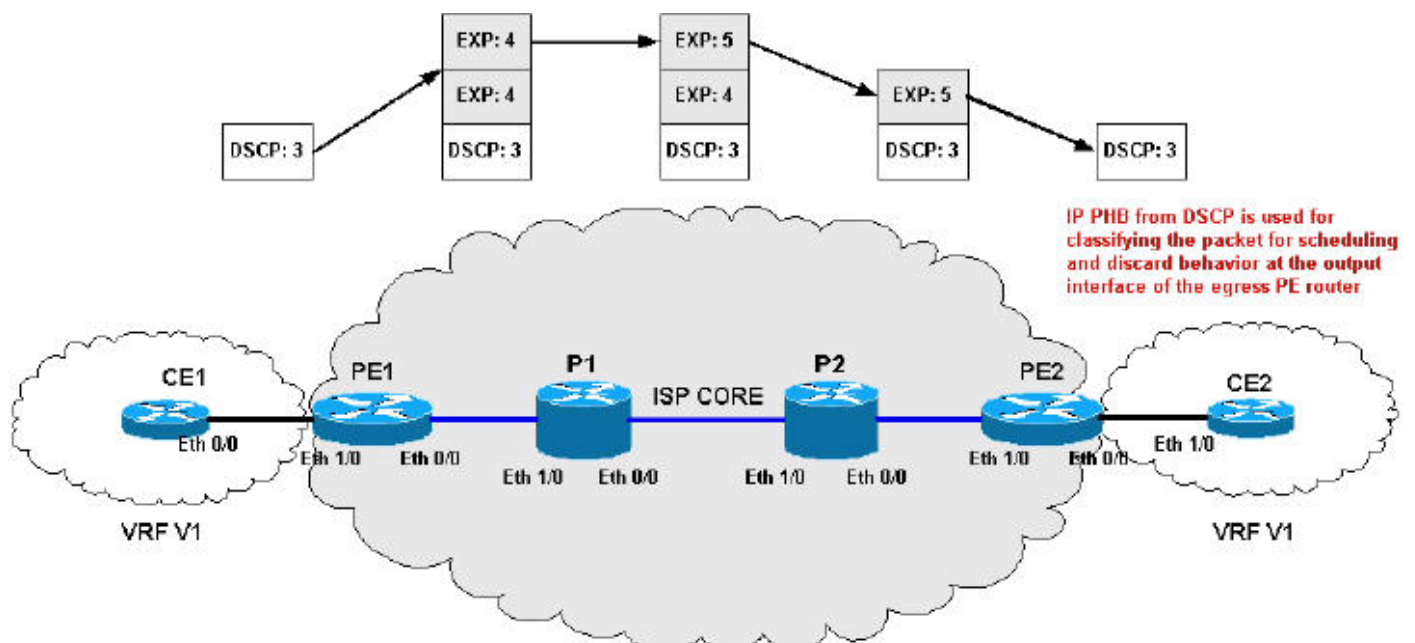
```

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
!

```

Modo de tubulação curta

O Diffserv Tunneling Short-pipe Mode utiliza as mesmas regras e técnicas através do núcleo. A diferença está no roteador PE de saída (PE2)—você classifica os pacotes IP recém-expostos para enfileiramento de saída com base no PHB IP do valor DSCP desse pacote IP.



Configurações do modo de tubulação curta:

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
!

```

Resumo do modo de túnel

Esta tabela resume as várias ações que se aplicaram a pacotes IP ou rotulados em vários estágios na rede:

Modo de tunelamento	IP -> Rótulo	Rótulo -> Rótulo	Rótulo -> IP
Uniforme	Copiar IP/DiffServ para MPLS EXP (também pode ser alterado pela controladora de armazenamento)	O MPLS EXP pode ser alterado pelo SP	MPLS EXP copiado para IP/DiffServ
Cano	MPLS EXP definido pela política de QoS do SP		IP/Diffserv original preservado (enfileiramento de saída baseado em MPLS EXP)
Cano			IP/Diffserv original

curto			preservado (enfileiramento de saída baseado em IP/DiffServ)
-------	--	--	--

Informações Relacionadas

- [Software Cisco IOS - MPLS](#)
- [P&R - Qualidade do serviço para redes de switching de rótulo multiprotocolo](#)
- [Referência de comando do Cisco IOS Switching Services, versão 12.2](#)
- [Guia de configuração de soluções de qualidade de serviço do Cisco IOS, versão 12.2](#)
- [Referência de comando de soluções de qualidade de serviço do Cisco IOS, versão 12.2](#)
- [MPLS Support Page](#)
- [página de suporte de QoS](#)
- [Suporte Técnico - Cisco Systems](#)