

Modos para las redes MPLS de la Tunelización del DiffServ

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedentes](#)

[Comportamiento predeterminado](#)

[Pre comando usage y comportamiento del Cisco IOS Software 12.2\(13\)T](#)

[Comando usage y comportamiento del Cisco IOS Software 12.2\(13\)T del poste](#)

[Modos de la Tunelización del DiffServ](#)

[Modo uniforme](#)

[Modo del tubo](#)

[Modo del Cortocircuito-tubo](#)

[Resumen del modo túnel](#)

[Información Relacionada](#)

[Introducción](#)

Este documento describe la implementación de los modos de tunelización de los servicios diferenciados (DiffServ) disponibles para los entornos de red basados en Multiprotocol Label Switching (MPLS).

[prerrequisitos](#)

[Requisitos](#)

Quienes lean este documento deben tener conocimiento de los siguientes temas:

- MPLS y MPLS para el Redes Privadas Virtuales (VPN)
- Conceptos referentes la Prioridad IP, el Tipo de servicio (ToS) y al DiffServ
- Marcación de paquetes de Calidad de Servicio (QoS) y clasificación usando la interfaz de línea CLI (MQC) del comando modular qos

[Componentes Utilizados](#)

La información que contiene este documento se basa en las siguientes versiones de software y

hardware.

- Versión 12.2(13)T del [®] del Cisco IOS Software para las nuevas características de QoS. La versión 12.1(5)T incluye las características originales de QoS.
- Cualquier router Cisco de las 3600 Series o de más adelante, por ejemplo el Cisco 3660 o los 7206 que soportan la funcionalidad del router del borde del router/del proveedor de la base del proveedor MPLS (p) (PE).

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

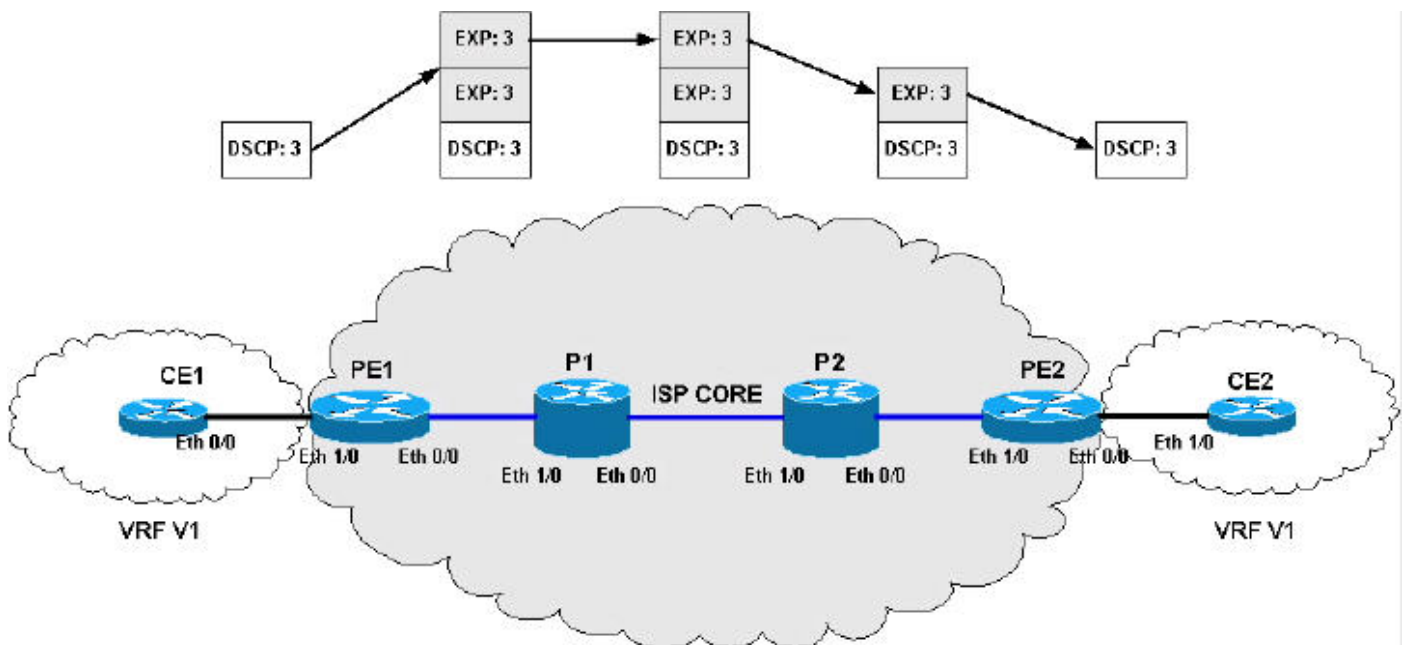
Convenciones

Para obtener más información sobre las convenciones del documento, consulte las [Convenciones de Consejos Técnicos de Cisco](#).

Antecedentes

Comportamiento predeterminado

Este diagrama describe el comportamiento predeterminado de los bits experimentales del DiffServ Code Point (DSCP) /MPLS (EXP) mientras que un paquete viaja a partir de un router de la frontera del cliente (CE) a otro router CE a través de una base MPLS:



Este resto de esta sección describe la actividad dentro del diagrama del comportamiento predeterminado.

Imposición de la escritura de la etiqueta (IP -> escritura de la etiqueta):

- La Prioridad IP del paquete del IP entrante se copia a los bits MPLS EXP de todas las escrituras de la etiqueta avanzadas.
- Los 1ros tres bits del bit DSCP se copian a los bits MPLS EXP de todas las escrituras de la

etiqueta avanzadas.

- Esta técnica también se conoce como reflexión de ToS.

Expedición MPLS (escritura de la etiqueta – > escritura de la etiqueta):

- El EXP se copia a las nuevas escrituras de la etiqueta que se intercambian/se avanzan durante la expedición o la imposición.
- En la imposición de etiqueta, las escrituras de la etiqueta que son la base no se modifican con el valor de la nueva escritura de la etiqueta que es agregada a la pila de etiquetas actual.
- En la disposición de etiqueta, los bits EXP no se copian abajo a los bits nuevamente expuestos de la escritura de la etiqueta EXP.

Disposición de la escritura de la etiqueta (escritura de la etiqueta – > IP):

- En la disposición de etiqueta los bits EXP no se copian abajo al campo IP precedence/DSCP del paquete del IP nuevamente expuesto.

[Pre comando usage y comportamiento del Cisco IOS Software 12.2\(13\)T](#)

Antes de la versión del IOS 12.2(13)T, el [comando set mpls experimental](#) era el único método disponible modificar los bits MPLS EXP.

[Comando usage y comportamiento del Cisco IOS Software 12.2\(13\)T del poste](#)

Con la versión del IOS 12.2(13)T y posterior, modifican al **comando set mpls experimental** de permitir estas opciones:

- [fije el MPLS Experimental superior {MPLS-exp-valor | \[table table-map-name\] del qos-grupo}](#)
- [fije la imposición del MPLS Experimental {MPLS-exp-valor | \[table table-map-name\] del qos-grupo}](#)

Nota: El nuevo **comando set mpls experimental topmost** es equivalente al viejo **comando set mpls imposition**.

Estos dos comandos, conjuntamente con un poco de Switches del comando new, permiten un mejor control de la manipulación de bits MPLS EXP durante el empuje, el intercambio, y las operaciones pop de la escritura de la etiqueta. Estos dos comandos permiten que usted utilice los modos del Tunelización del DiffServ.

[Modos del Tunelización del DiffServ](#)

Los modos del Tunelización del DiffServ introducen un nuevo Per-Hop Behavior (PHB), que permite QoS distinguido en una red de proveedores. El modo del Tunelización se define en el borde de la red, normalmente en los Label Switch Router PE (LSR) (ingreso y salida). Usted puede necesitar realizar los cambios en el Routers P; usted debe también considerar qué ocurre cuando la etiqueta superior se quita de un paquete debido al Penultimate Hop Popping (PHP). Puede ser necesario copiar el valor MPLS EXP de la escritura de la etiqueta superior que se está haciendo estallar a la escritura de la etiqueta nuevamente expuesta; esto no se aplica siempre a todos los modos del Tunelización.

En algunos casos (por ejemplo, una red MPLS llana NON-VPN), la acción de PHP en el router final P puede exponer un paquete del IP llano cuando un paquete con solamente una escritura de

la etiqueta se recibe. Cuando este paquete del IP es recibido por la salida LSR (PE), no es posible clasificar el paquete basado en los bits MPLS EXP porque ahora no hay escritura de la etiqueta. En estas situaciones, usted debe configurar al router de la salida PE para hacer publicidad de una **Etiqueta NULL explícita**. Cuando la acción de PHP se realiza en el router P, una escritura de la etiqueta con un valor de cero se envía, y con esta escritura de la etiqueta especial usted puede marcar los bits EXP como normalmente paquetes etiquetados, permitiendo la clasificación correcta en el router de la salida PE.

Compatibilidad de red MPLS de especificación Diffserv define estos modos del Tunelización:

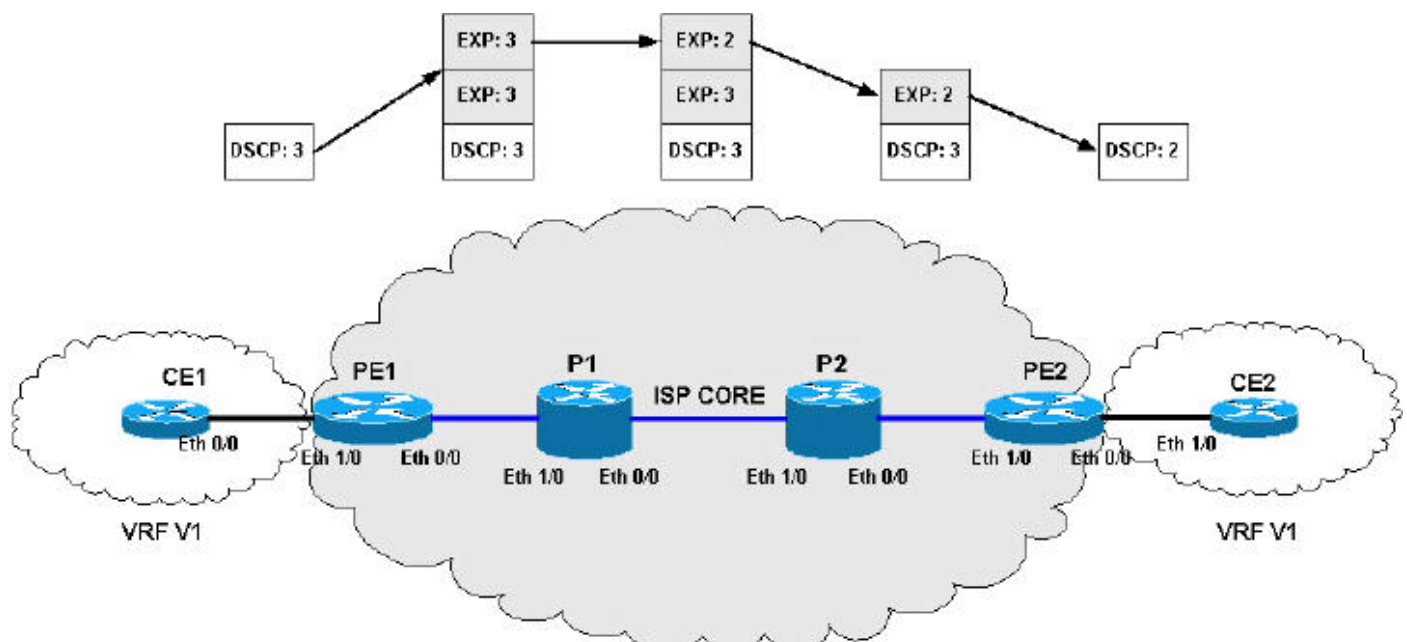
- [Uniforme](#)
- [Tubo](#)
- [Cortocircuito-tubo](#)

Las siguientes secciones examinan cada modo del Tunelización por separado, y proporcionan los ejemplos para mostrar cómo cada modo del Tunelización puede ser configurado. Los ejemplos incluyen una asignación completa de la Prioridad IP a los bits MPLS EXP. Es posible tener varios diversos parámetros de QoS y modos del Tunelización para cada cliente.

Nota: Los ejemplos de configuración no son específicos para el MPLS VPN, y son aplicables para las redes llanas de las redes MPLS y de la Portadora que admite portadora (csc). Es también posible que su red puede variar de otra red — muchos diversos parámetros de QoS y modos del Tunelización pueden ser utilizados.

Modo uniforme

El modo uniforme del Tunelización del DiffServ tiene solamente una capa de QoS, que alcanza de punta a punta. El router del ingreso PE (PE1) copia el DSCP del paquete del IP entrante en los bits MPLS EXP de las escrituras de la etiqueta impuestas. Mientras que los bits EXP viajan con la base, pueden o no pueden ser modificados por el Routers intermedio P. En este ejemplo, el router P1 P modifica los bits EXP de la escritura de la etiqueta superior. En el router de la salida P (P2) copiamos los bits EXP a los bits EXP de la escritura de la etiqueta nuevamente expuesta después del PHP (Penúltimo-Salto-estallido). Finalmente en el router de la salida PE (PE2) entonces copiamos los bits EXP a los bits DSCP del paquete del IP nuevamente expuesto.



Configuraciones del 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-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
!

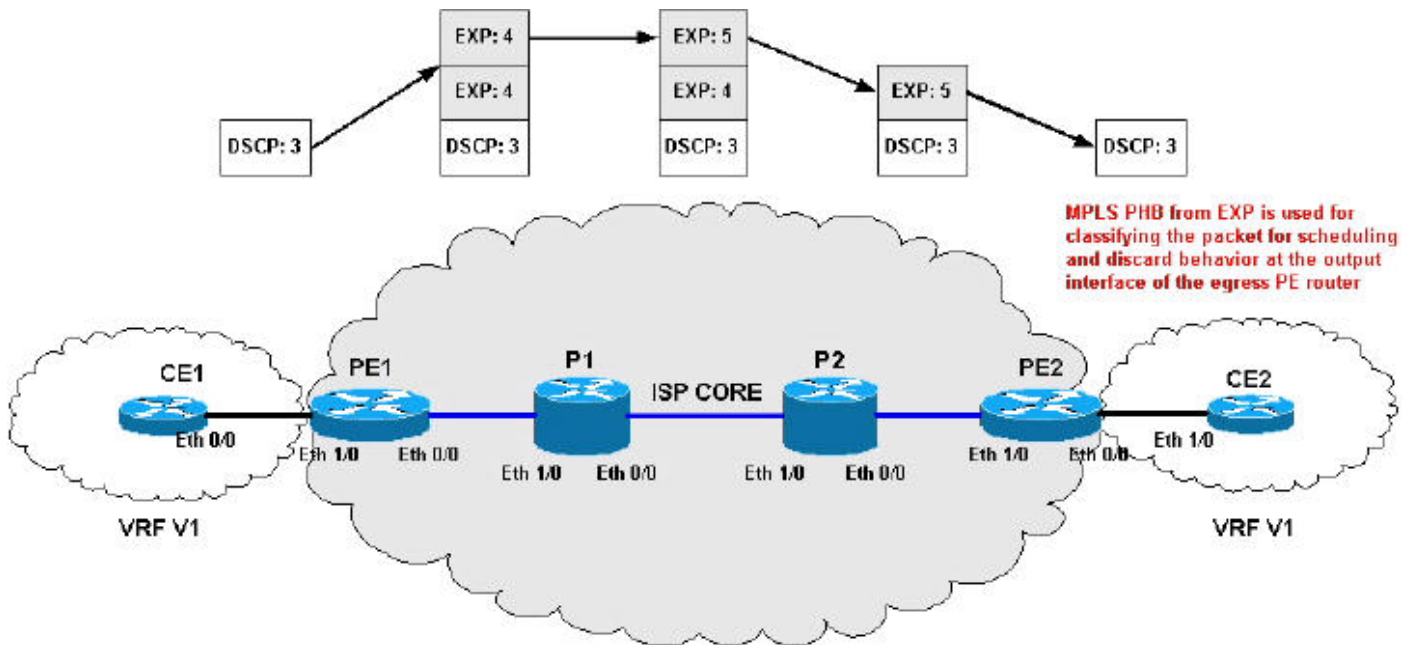
```

Modo del tubo

El modo del tubo de la Tunelización del DiffServ utiliza dos capas de QoS:

1. Un QoS subyacente para los datos, que sigue habiendo sin cambiar al atravesar la base.
2. Una por-memoria QoS, que está a parte de la de los paquetes del IP subyacentes. Esta por-memoria QoS PHB sigue siendo transparente a los usuarios finales.

Cuando un paquete alcanza el borde de la base MPLS, el router de la salida PE (PE2) clasifica los paquetes del IP nuevamente expuestos para los Datos en espera salientes basados en el MPLS PHB de los bits EXP de la escritura de la etiqueta recientemente quitada.



Configuraciones de modo del tubo:

```

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

```

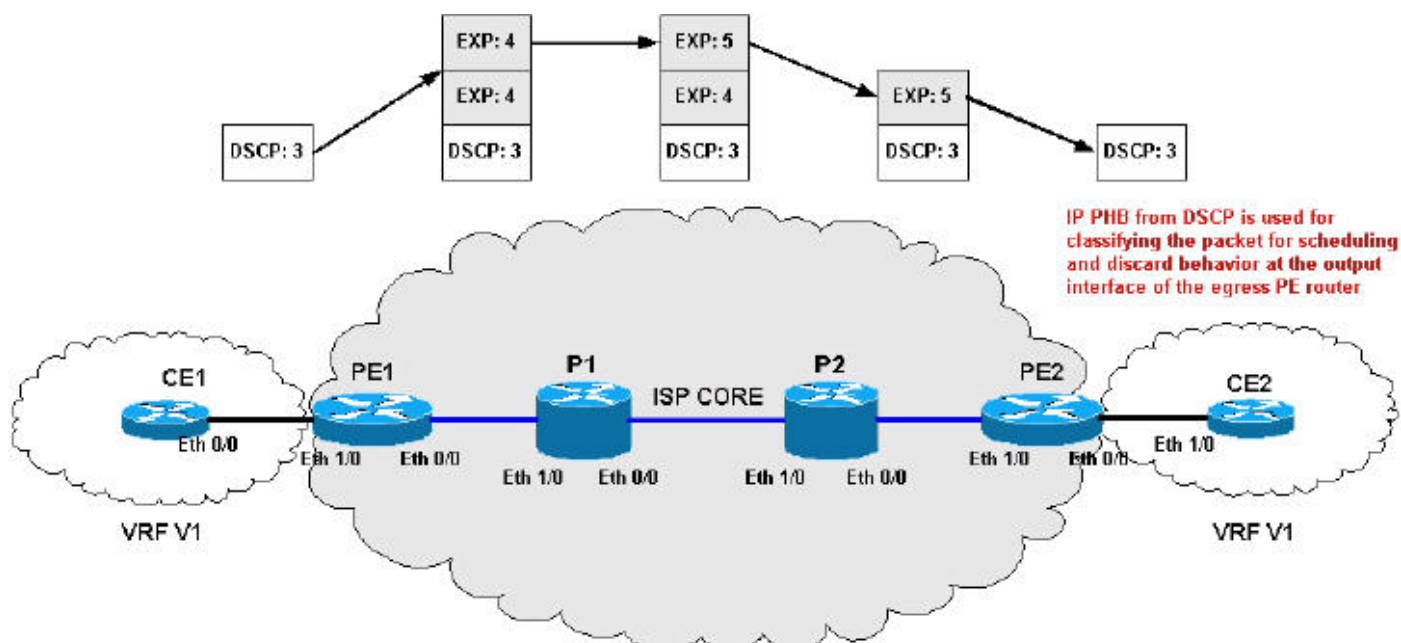
```

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 del Cortocircuito-tubo

El modo del Cortocircuito-tubo del Tunelización del DiffServ utiliza las mismas reglas y técnicas a través de la base. La diferencia está en el router de la salida PE (PE2) — usted clasifica los paquetes del IP nuevamente expuestos para los Datos en espera salientes basados en el IP PHB del valor DSCP de este paquete del IP.



configuraciones de modo del Cortocircuito-tubo:

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
!

```

Resumen del modo túnel

Esta tabla resume las diversas acciones que se aplicaron al IP o a los paquetes etiquetados en las diversas etapas en la red:

Modo el hacer un túnel	IP -> escritura de la etiqueta	Escritura de la etiqueta -> escritura de la etiqueta	Escritura de la etiqueta -> IP
Uniforme	Copie el Prec/DiffServ IP en MPLS EXP (puede ser cambiado por el SP también)	El MPLS EXP se puede cambiar por el SP	MPLS EXP copiado al Prec/DiffServ IP
Tubo	MPLS EXP fijado por el SP política de calidad de servicio (QoS)		IP original Prec/DiffServ preservado (Datos en espera de la salida basados en MPLS EXP)
Cortocircuito-tubo			IP original Prec/DiffServ preservado (Datos en espera de la salida basados en el

			Prec/DiffServ (P)
--	--	--	----------------------

Información Relacionada

- [Cisco IOS Software - MPLS](#)
- [Q&A - Calidad de servicio para las redes del Multi-Protocol Label Switching](#)
- [Referencia de comandos del Cisco IOS Switching Services, Release12.2](#)
- [Guía de configuración de las soluciones de la Calidad de servicio de Cisco IOS, Release12.2](#)
- [Referencia del comando solutions de la Calidad de servicio de Cisco IOS, Release12.2](#)
- [Página de soporte de MPLS](#)
- [Página de Soporte de Qos \(Calidad de Servicio\)](#)
- [Soporte Técnico - Cisco Systems](#)