

DiffServ-Tunneling-Modi für MPLS-Netzwerke

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[Standardverhalten](#)

[Befehlsverwendung und -verhalten der Cisco IOS Software 12.2\(13\)T](#)

[Verwendung und Verhalten der Befehle nach Cisco IOS Software 12.2\(13\)T](#)

[DiffServ-Tunneling-Modi](#)

[Einheitlicher Modus](#)

[Pipe-Modus](#)

[Short Pipe-Modus](#)

[Zusammenfassung des Tunnelmodus](#)

[Zugehörige Informationen](#)

Einführung

In diesem Dokument wird die Implementierung von Differentiated Services (DiffServ) Tunneling-Modi beschrieben, die für MPLS-basierte Netzwerkumgebungen (Multiprotocol Label Switching) verfügbar sind.

Voraussetzungen

Anforderungen

Die Leser dieses Dokuments sollten folgende Themen kennen:

- MPLS und MPLS für Virtual Private Networks (VPN)
- Konzepte bezüglich IP-Rangfolge, Type of Service (ToS) und DiffServ
- QoS-Paketkennzeichnung und -klassifizierung mit modularer QoS-Befehlszeilenschnittstellen-CLI (MQC)

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Cisco IOS Software® Version 12.2(13)T für neue QoS-Funktionen Version 12.1(5)T enthält ursprüngliche QoS-Funktionen.
- Jeder Cisco Router der Serie 3600 oder höher, z. B. der Cisco Router 3660 oder 7206, der die Routerfunktionalität des MPLS-Provider-Core (P)-Routers/Provider Edge (PE) unterstützt.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

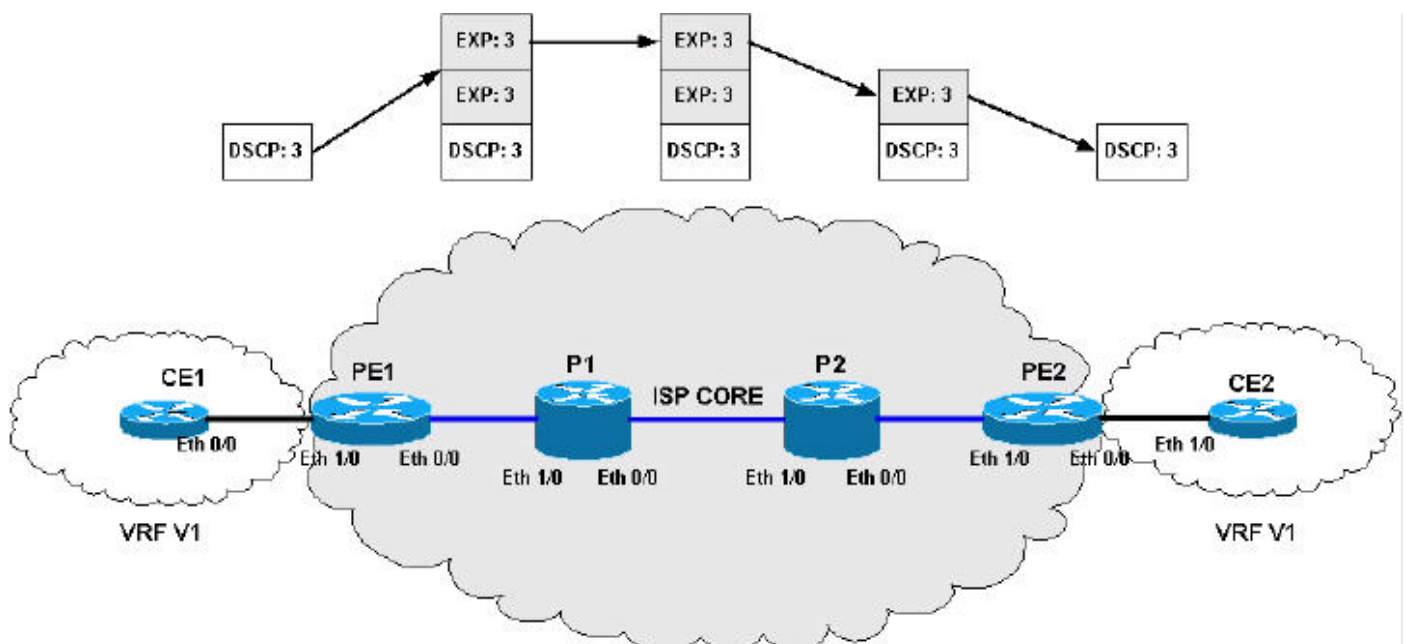
Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie in den [Cisco Technical Tips Conventions](#).

Hintergrundinformationen

Standardverhalten

Dieses Diagramm beschreibt das Standardverhalten der DSCP-/EXP-Bits (DiffServ Code Point)/MPLS Experimental (EXP), wenn ein Paket von einem CE-Router (Customer Edge) über einen MPLS-Core zu einem anderen CE-Router wechselt:



In diesem Abschnitt wird die Aktivität im Diagramm für das Standardverhalten beschrieben.

Imposition of Label (IP -> Label):

- Die IP-Rangfolge des eingehenden IP-Pakets wird in die MPLS EXP-Bits aller Push-Label(s) kopiert.
- Die 1-3 Bit des DSCP-Bits werden in die MPLS EXP-Bits aller Push-Label(s) kopiert.
- Diese Technik wird auch als ToS-Reflektion bezeichnet.

MPLS-Weiterleitung (Label -> Label):

- Der EXP wird auf die neuen Labels kopiert, die während der Weiterleitung oder der Installation ausgetauscht/geschoben werden.
- Bei der Label-Erstellung werden die zugrunde liegenden Labels nicht geändert, wobei der Wert des neuen Labels dem aktuellen Label-Stack hinzugefügt wird.
- Bei Label-Disposition werden die EXP-Bits nicht in die neu verfügbar gemachten EXP-Bits kopiert.

Entsorgung des Labels (Label -> IP):

- Bei der Label-Disposition werden die EXP-Bits nicht in das IP-Rangfolge-/DSCP-Feld des neu verfügbar gemachten IP-Pakets kopiert.

Befehlsverwendung und -verhalten der Cisco IOS Software 12.2(13)T

Vor der IOS-Version 12.2(13)T war der Befehl `set mpls experimental` die einzige Möglichkeit, die MPLS EXP-Bits zu ändern.

Verwendung und Verhalten der Befehle nach Cisco IOS Software 12.2(13)T

Mit IOS Release 12.2(13)T und höher wird der Befehl `set mpls experimental` geändert, um folgende Optionen zu ermöglichen:

- [MPLS experimentelle topmost `{mpls-exp-Wert | qos-group \[Tabellenname-Kartename\]}`](#)
- [MPLS experimentelle Auferlegung `{mpls-exp-Wert | qos-group \[Tabellenname-Kartename\]}`](#)

Hinweis: Der Befehl `experimentelle oberste MPLS experimentell` entspricht dem Befehl `alte mpls-Anweisung`.

Diese beiden Befehle ermöglichen in Kombination mit einigen neuen Befehls-Switches eine bessere Kontrolle der Manipulation von MPLS EXP-Bits während Label-Push-, Swap- und Pop-Vorgängen. Mit diesen beiden Befehlen können Sie DiffServ-Tunneling-Modi verwenden.

DiffServ-Tunneling-Modi

Mit den Diffserv-Tunneling-Modi wird ein neues Per-Hop-Behavior (PHB) eingeführt, das eine differenzierte QoS in einem Anbieternetzwerk ermöglicht. Der Tunneling-Modus wird am Netzwerk-Edge definiert, normalerweise in den PE-Label-Switch-Routern (LSRs) (ein- und ausgehend). Möglicherweise müssen Sie Änderungen an den P-Routern vornehmen. Sie müssen auch bedenken, was geschieht, wenn das oberste Label aus einem Paket entfernt wird, weil Penultimate-Hop-Popping (PHP). Möglicherweise muss der MPLS EXP-Wert aus dem oberen Label, das auf das neu exponierte Label gehoben wird, kopiert werden. Dies gilt nicht immer für alle Tunneling-Modi.

In einigen Fällen (z. B. bei einem einfachen MPLS-Netzwerk ohne VPN) kann die PHP-Aktion auf dem letzten P-Router ein einfaches IP-Paket verfügbar machen, wenn ein Paket mit nur einem Label empfangen wird. Wenn dieses IP-Paket vom Ausgangs-LSR (PE) empfangen wird, ist es nicht möglich, das Paket anhand der MPLS EXP-Bits zu klassifizieren, da es derzeit kein Label gibt. In diesen Situationen müssen Sie den Egress-PE-Router so konfigurieren, dass ein **explizites Null-Label** angekündigt wird. Wenn die PHP-Aktion auf dem P-Router ausgeführt wird, wird ein Label mit dem Wert 0 gesendet, und mit diesem speziellen Label können Sie die EXP-Bits als normal bezeichnete Pakete markieren, was die korrekte Klassifizierung auf dem Egress-PE-

Router ermöglicht.

Die MPLS-Netzwerkunterstützung der Diffserv-Spezifikation definiert die folgenden Tunneling-Modi:

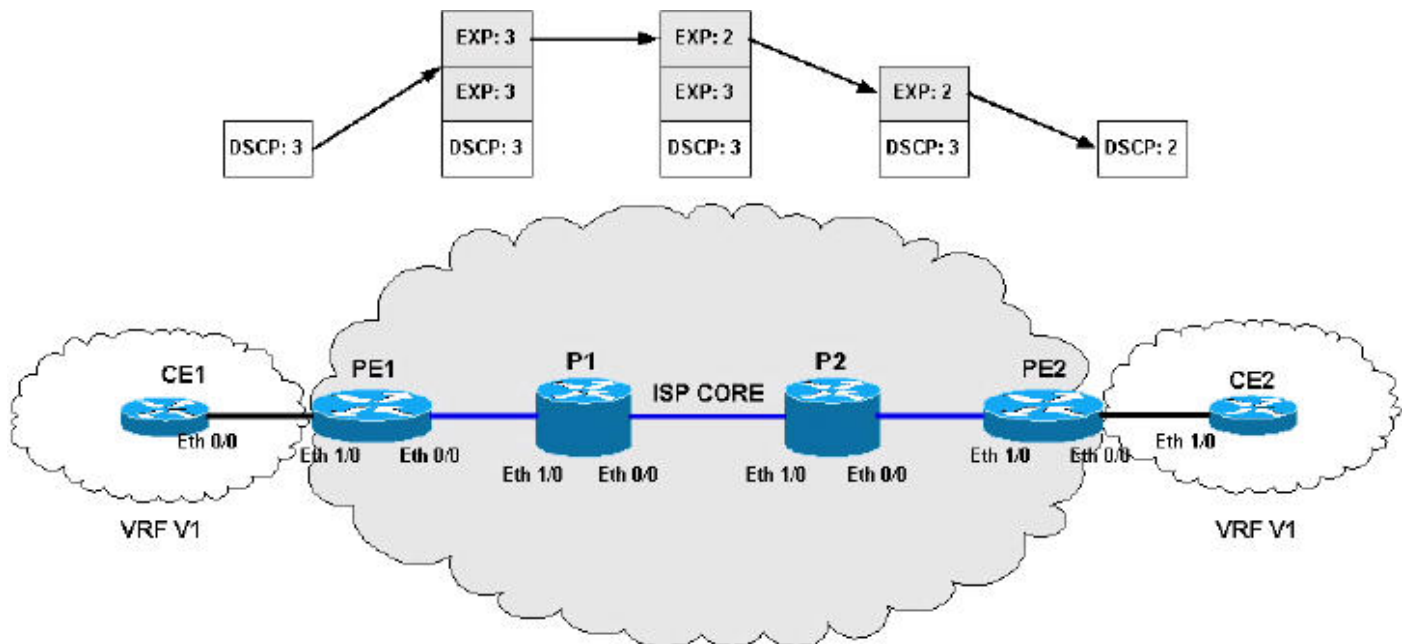
- [Einheitlich](#)
- [Rohrleitung](#)
- [Kurzwahl](#)

In den nächsten Abschnitten werden die einzelnen Tunneling-Modi einzeln untersucht. Außerdem werden Beispiele für die Konfiguration der einzelnen Tunneling-Modi bereitgestellt. Die Beispiele umfassen eine vollständige Zuordnung der IP-Rangfolge zu MPLS EXP-Bits. Für jeden Kunden können verschiedene QoS-Parameter und Tunneling-Modi verwendet werden.

Hinweis: Die Konfigurationsbeispiele sind nicht spezifisch für MPLS VPN und gelten für einfache MPLS-Netzwerke und CsC-Netzwerke (Carrier supported Carrier). Es ist auch möglich, dass Ihr Netzwerk von einem anderen Netzwerk abweichen kann. Es können viele verschiedene QoS-Parameter und Tunneling-Modi verwendet werden.

Einheitlicher Modus

Der Uniform-Modus für DiffServ-Tunneling verfügt nur über eine QoS-Ebene, die End-to-End erreicht. Der Eingangs-PE-Router (PE1) kopiert das DSCP aus dem eingehenden IP-Paket in die MPLS EXP-Bits der angegebenen Labels. Da die EXP-Bits den Core durchlaufen, können sie durch zwischengeschaltete P-Router modifiziert werden oder nicht. In diesem Beispiel ändert P-Router P1 die EXP-Bits des oberen Labels. Am Egress-P-Router (P2) kopieren wir die EXP-Bits nach dem PHP (Penultimate-Hop-Pop) in die EXP-Bits des neu verfügbar gemachten Labels. Am Egress-PE-Router (PE2) kopieren wir anschließend die EXP-Bits in die DSCP-Bits des neu verfügbar gemachten IP-Pakets.



Einheitliche Moduskonfigurationen:

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
!

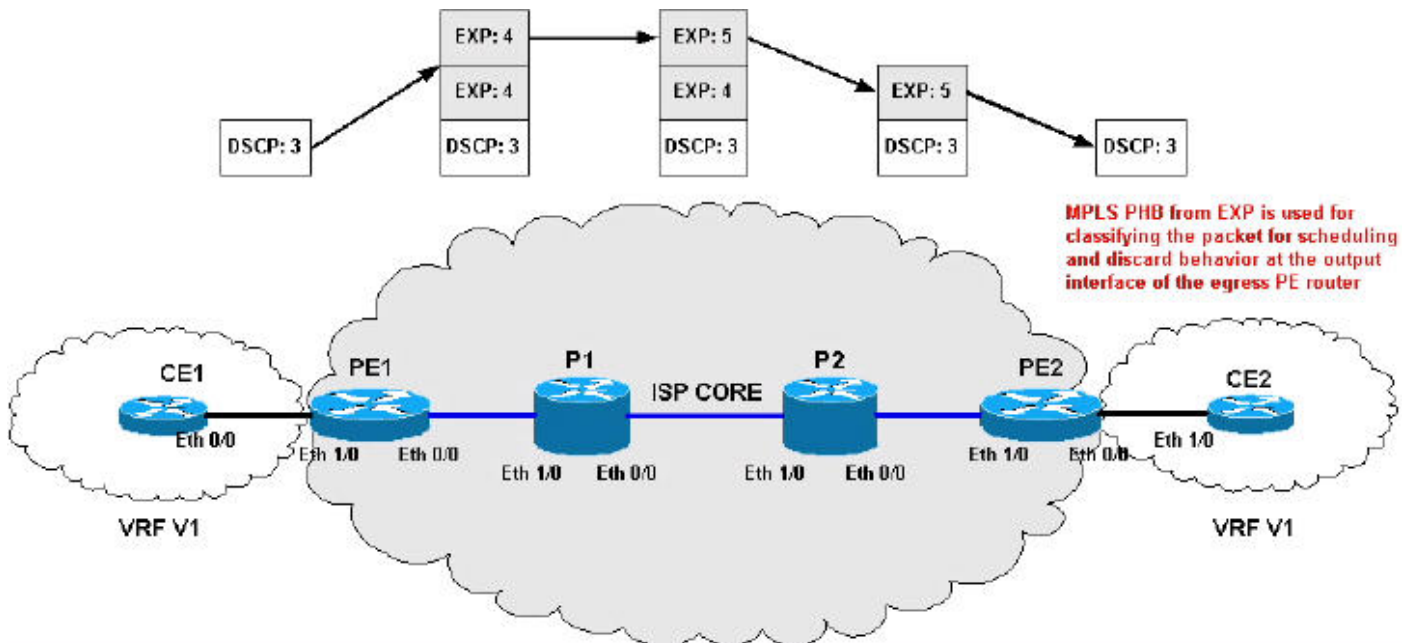
```

Pipe-Modus

Der Diffserv-Tunneling-Pipe-Modus verwendet zwei Ebenen von QoS:

1. Eine zugrunde liegende QoS für die Daten, die beim Durchlaufen des Kerns unverändert bleibt.
2. Eine Pro-Core-QoS, die von der zugrunde liegenden IP-Pakete getrennt ist. Dieses QoS-PHB pro Kern bleibt für Endbenutzer transparent.

Wenn ein Paket den Edge des MPLS-Core erreicht, klassifiziert der Egress-PE-Router (PE2) die neu verfügbar gemachten IP-Pakete für die Warteschlangenverwaltung für ausgehende Anrufe basierend auf dem MPLS-PHB aus den EXP-Bits des kürzlich entfernten Labels.



Konfiguration des Pipe-Modus:

```

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

```

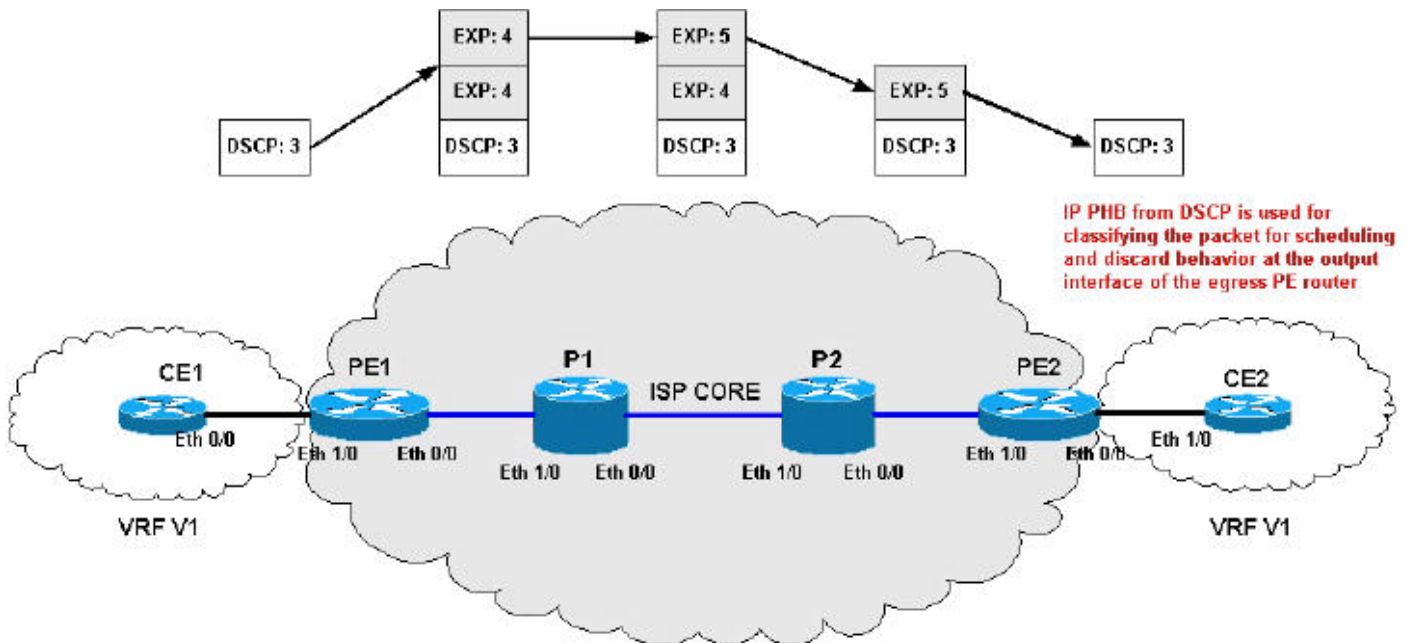
```

    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
!

```

Short Pipe-Modus

Der Short-Pipe-Modus für Diffserv-Tunneling verwendet im Core dieselben Regeln und Techniken. Der Unterschied liegt beim Egress-PE-Router (PE2). Sie klassifizieren die neu verfügbar gemachten IP-Pakete für die Warteschlangenverwaltung für ausgehende Anrufe basierend auf dem IP-PHB aus dem DSCP-Wert dieses IP-Pakets.



Konfigurationen für den Short-Pipe-Modus:

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 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 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
!

```

Zusammenfassung des Tunnelmodus

In dieser Tabelle sind die verschiedenen Aktionen zusammengefasst, die auf IP-Pakete oder Pakete mit Labels in verschiedenen Phasen des Netzwerks angewendet werden:

Tunneling-Modus	IP -> Label	Label -> Label	Label -> IP
Einheitlich	Kopieren von IP Prec/DiffServ in MPLS EXP (kann auch vom SP geändert werden)	MPLS EXP kann von SP geändert werden.	MPLS EXP in IP Prec/DiffServ kopiert
Rohrleitung	MPLS EXP wird von der SP QoS-Richtlinie festgelegt		Original IP Prec/Diffserv konserviert (Ausgangswarteschlange basierend auf MPLS EXP)
Kurzwahl			Original IP Prec/Diffserv konserviert (Ausgangswarteschlange basierend auf IP Prec/DiffServ)

Zugehörige Informationen

- [Cisco IOS-Software - MPLS](#)
- [Q&A - Quality of Service für Multi-Protocol Label Switching-Netzwerke](#)
- [Befehlsreferenz zu Cisco IOS Switching Services, Version 12.2](#)
- [Cisco IOS-Konfigurationsleitfaden für Quality of Service-Lösungen, Version 12.2](#)
- [Cisco IOS Quality of Service Solutions Command Reference, Version 12.2](#)
- [MPLS-Support-Seite](#)
- [QoS-Support-Seite](#)
- [Technischer Support - Cisco Systems](#)