

# MPLS 네트워크에 대한 DiffServ 터널링 모드

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기규칙](#)

[배경 정보](#)

[기본 동작](#)

[Cisco IOS Software 12.2\(13\)T 이전 명령 사용 및 동작](#)

[Post Cisco IOS Software 12.2\(13\)T 명령 사용 및 동작](#)

[DiffServ 터널링 모드](#)

[균일 모드](#)

[파이프 모드](#)

[짧은 파이프 모드](#)

[터널 모드 요약](#)

[관련 정보](#)

## 소개

이 문서에서는 MPLS(Multiprotocol Label Switching) 기반 네트워크 환경에 사용할 수 있는 DiffServ(Differentiated Services) 터널링 모드의 구현에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

이 문서의 독자는 다음 주제에 대해 알고 있어야 합니다.

- VPN(Virtual Private Network)용 MPLS 및 MPLS
- IP 우선 순위, 서비스 유형(ToS) 및 DiffServ와 관련된 개념
- 모듈식 QoS MQC(Command Line Interface CLI)를 사용한 QoS(Quality of Service) 패킷 마킹 및 분류

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco IOS Software® Release 12.2(13)T - 새로운 QoS 기능 릴리스 12.1(5)T에는 원래 QoS 기능이 포함되어 있습니다.

- MPLS P(Provider Core) 라우터/PE(Provider Edge) 라우터 기능을 지원하는 Cisco 3660 또는 7206 등 3600 시리즈 이상의 모든 Cisco 라우터

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

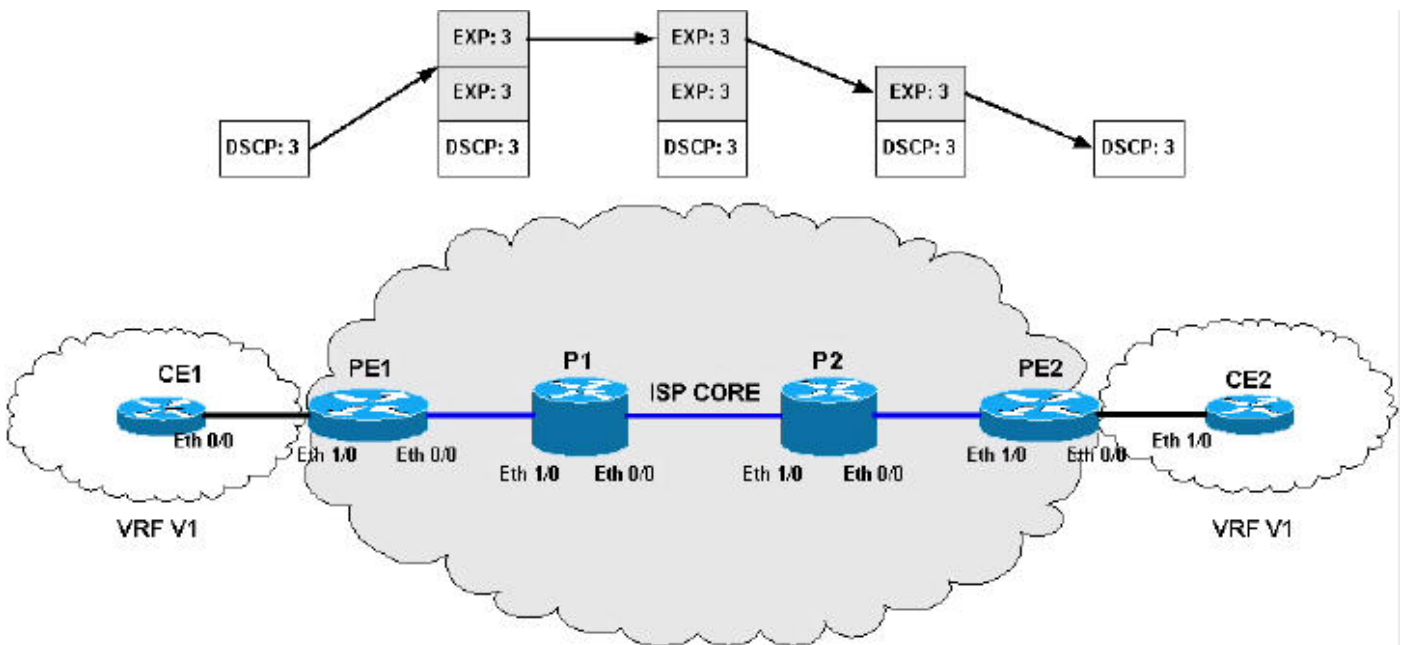
## 표기 규칙

문서 규칙에 대한 자세한 내용은 [Cisco 기술 팁 표기 규칙](#)을 참조하십시오.

## 배경 정보

### 기본 동작

이 다이어그램은 패킷이 MPLS 코어를 통해 한 고객 에지(CE) 라우터에서 다른 CE 라우터로 이동하는 DiffServ 코드 포인트(DSCP)/MPLS 실험(EXP) 비트의 기본 동작을 설명합니다.



이 섹션의 나머지 부분에서는 기본 동작 다이어그램 내의 활동에 대해 설명합니다.

레이블 부과(IP -> 레이블):

- 들어오는 IP 패킷의 IP 우선 순위는 모든 푸시된 레이블의 MPLS EXP 비트로 복사됩니다.
- DSCP 비트 1번째 3비트가 모든 푸시된 레이블의 MPLS EXP 비트로 복사됩니다.
- 이 기법은 ToS Reflection이라고도 합니다.

MPLS 전달(레이블 -> 레이블):

- EXP는 전달 또는 부과 중에 스왑/푸시되는 새 레이블에 복사됩니다.
- 레이블 적용 시 기본 레이블은 현재 레이블 스택에 추가되는 새 레이블의 값으로 수정되지 않습니다.
- 레이블 처리 시 EXP 비트는 새로 노출된 레이블 EXP 비트로 복사되지 않습니다.

레이블 속성(레이블 -> IP):

- 레이블 분류 시 EXP 비트는 새로 노출된 IP 패킷의 IP precedence/DSCP 필드에 복사되지 않습니다.

## Cisco IOS Software 12.2(13)T 이전 명령 사용 및 동작

IOS 릴리스 12.2(13)T 이전에는 MPLS EXP 비트를 수정하는 데 사용할 수 있는 유일한 방법은 `set mpls experimental` 명령이었습니다.

## Post Cisco IOS Software 12.2(13)T 명령 사용 및 동작

IOS 릴리스 12.2(13)T 이상에서 `set mpls experimental` 명령이 수정되어 다음 옵션을 허용합니다.

- [mpls experimental topmost {mpls-exp-value 설정 | qos-group \[table-map-name\]}](#)
- [mpls 실험적 부과 {mpls-exp-value 설정 | qos-group \[table-map-name\]}](#)

참고: 새 `set mpls experimental topmost` 명령은 이전 `set mpls imposition` 명령과 같습니다.

이 두 명령은 일부 새로운 명령 스위치와 함께 레이블 푸시, 스왑 및 팝업 작업 중에 MPLS EXP 비트 조작을 보다 효과적으로 제어할 수 있습니다. 이 두 명령을 사용하면 DiffServ 터널링 모드를 사용할 수 있습니다.

## DiffServ 터널링 모드

DiffServ Tunneling Mode에는 사업자 네트워크에서 차별화된 QoS를 허용하는 새로운 PHB(Per-Hop-Behavior)가 도입되었습니다. 터널링 모드는 네트워크 에지에서 정의되며 일반적으로 PE LSR(Label Switch Router)(인그레스 및 이그레스 모두)에서 정의됩니다. P 라우터를 변경해야 할 수도 있습니다. 또한 PHP(Penultimate-Hop-Popting)로 인해 패킷에서 맨 위 레이블이 제거될 때 어떤 일이 발생하는지 고려해야 합니다. 팝업되는 맨 위 레이블에서 새로 노출된 레이블로 MPLS EXP 값을 복사해야 할 수 있습니다. 이는 모든 터널링 모드에 적용되는 것은 아닙니다.

경우에 따라(예: 일반 비 VPN MPLS 네트워크), 최종 P 라우터의 PHP 작업은 레이블이 하나만 있는 패킷을 수신할 때 일반 IP 패킷을 노출할 수 있습니다. 이 IP 패킷이 이그레스 LSR(PE)에 의해 수신되는 경우 현재 레이블이 없으므로 MPLS EXP 비트를 기반으로 패킷을 분류할 수 없습니다. 이러한 경우 명시적 null 레이블을 알리도록 이그레스 PE 라우터를 구성해야 합니다. PHP 작업이 P 라우터에서 수행되면 값이 0인 레이블이 전송되고 이 특수 레이블이 있는 EXP 비트를 일반적으로 레이블이 지정된 패킷으로 표시하여 이그레스 PE 라우터에서 올바른 분류를 허용합니다.

DiffServ 사양의 MPLS 네트워크 지원은 다음 터널링 모드를 정의합니다.

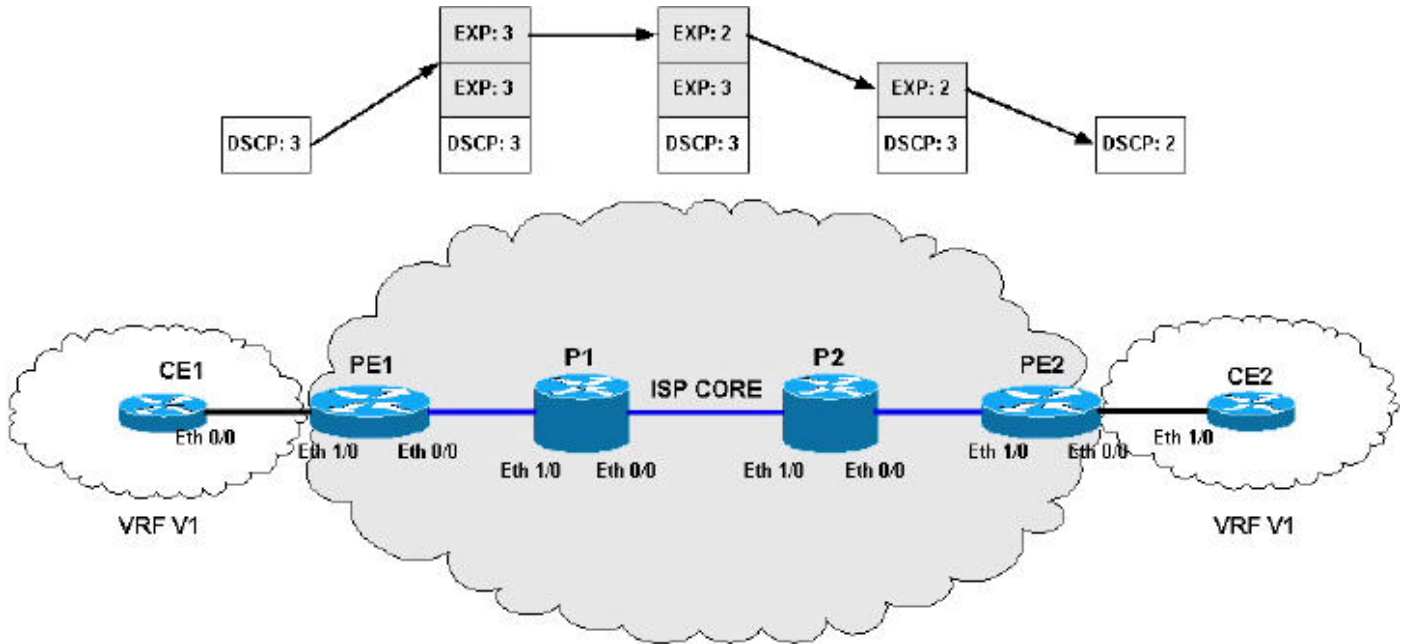
- [유니폼](#)
- [파이프](#)
- [짧은 파이프](#)

다음 섹션에서는 각 터널링 모드를 개별적으로 검토하고 각 터널링 모드를 구성할 수 있는 방법을 보여 주는 예제를 제공합니다. 이 예에는 IP Precedence를 MPLS EXP 비트에 대한 전체 매핑이 포함됩니다. 각 고객에 대해 다양한 QoS 매개변수 및 터널링 모드가 있을 수 있습니다.

참고: 컨피그레이션 예제는 MPLS VPN에 한정되지 않으며 일반 MPLS 네트워크 및 CsC(Carrier Supported Carrier) 네트워크에 적용됩니다. 네트워크는 다른 네트워크와 다를 수 있습니다. 다양한 QoS 매개변수 및 터널링 모드를 사용할 수 있습니다.

## 균일 모드

DiffServ 터널링 Uniform Mode에는 엔드 투 엔드 QoS 레이어가 하나만 있습니다. 인그레스 PE 라우터(PE1)는 수신 IP 패킷의 DSCP를 지정된 레이블의 MPLS EXP 비트로 복사합니다. EXP 비트가 코어를 통과하면서 중간 P 라우터에 의해 수정되거나 수정되지 않을 수 있습니다. 이 예에서 P 라우터 P1은 상위 레이블의 EXP 비트를 수정합니다. 이그레스 P 라우터(P2)에서는 PHP(Penultimate-Hop-Pop) 다음에 새로 노출된 레이블의 EXP 비트에 EXP 비트를 복사합니다. 마지막으로 이그레스 PE 라우터(PE2)에서 EXP 비트를 새로 노출된 IP 패킷의 DSCP 비트로 복사합니다.



균일 모드 컨피그레이션:

```

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-imp-
    position-transmit
    0 exceed-
    action drop
  class IP-AF12
    police 10000 conform-action set-mpls-exp-imp-
    position-transmit
    1 exceed-action set-mpls-exp-imp-
    position-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
!

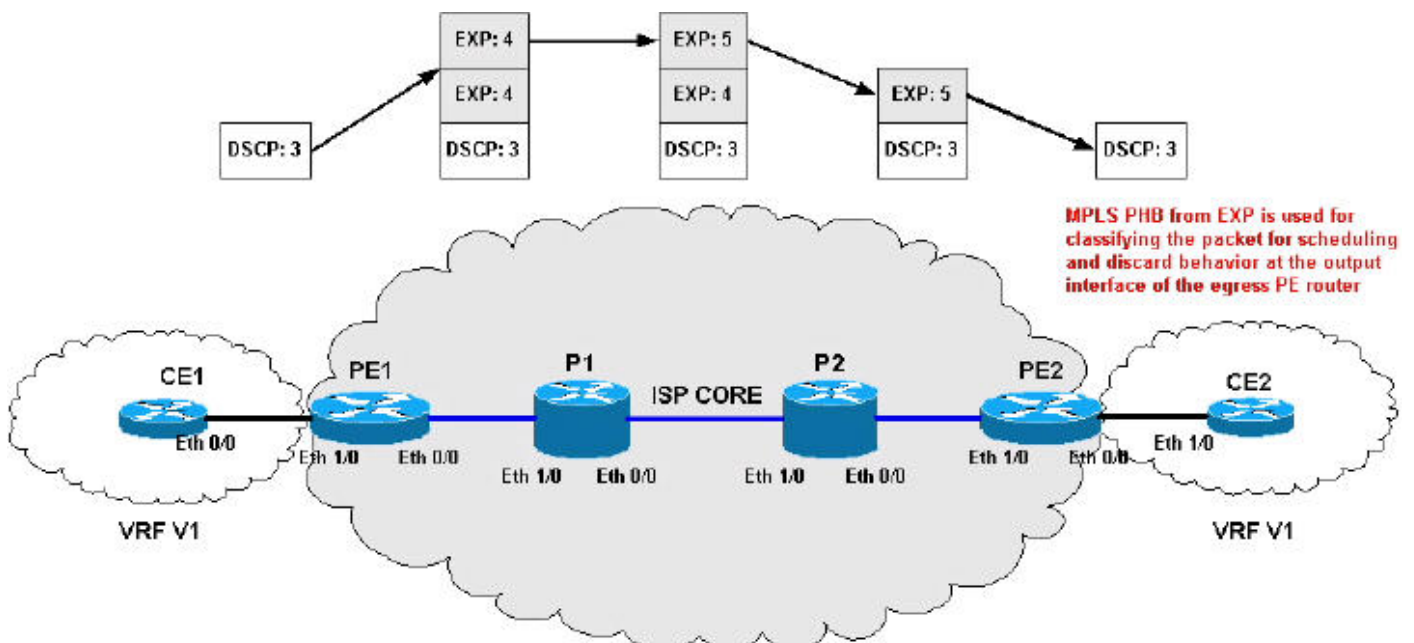
```

## 파이프 모드

Diffserv 터널링 파이프 모드에서는 두 개의 QoS 레이어를 사용합니다.

1. 데이터에 대한 기본 QoS로, 코어를 이동할 때 변경되지 않습니다.
2. 기본 IP 패킷과 별개인 코어별 QoS. 이 코어별 QoS PHB는 엔드 유저에게 투명하게 유지됩니다.

패킷이 MPLS 코어의 에지에 도달하면 이그레스 PE 라우터(PE2)는 최근에 제거된 레이블의 EXP 비트에서 MPLS PHB를 기반으로 아웃바운드 대기열의 새로 노출된 IP 패킷을 분류합니다.



파이프 모드 컨피그레이션:

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
```

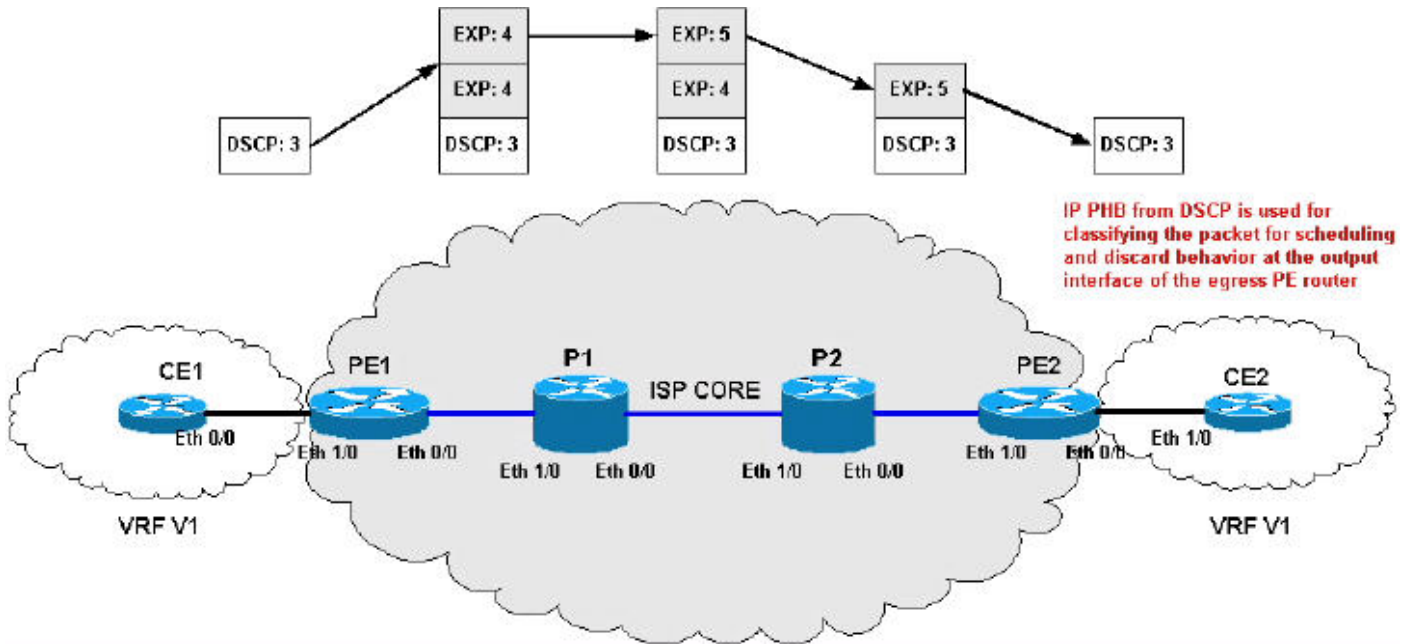
```

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
!

```

## 짧은 파이프 모드

Diffserv Tunneling Short-pipe Mode는 코어에서 동일한 규칙 및 기술을 사용합니다. 차이점은 이그레스 PE 라우터(PE2)입니다. 이 IP 패킷의 DSCP 값과 IP PHB를 기반으로 아웃바운드 큐잉에 대해 새로 노출된 IP 패킷을 분류합니다.



짧은 파이프 모드 컨피그레이션:

### 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-gos
    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-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 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
!

```

## 터널 모드 요약

이 표에는 네트워크의 다양한 단계에서 IP에 적용되거나 레이블이 지정된 패킷에 적용되는 다양한 작업이 요약되어 있습니다.

터널 모드	IP -> 레이블	레이블 -> 레이블	레이블 -> IP
패킷	IP Prec/DiffServ를 MPLS EXP에 복사(SP에서 변경할 수도 있음)	MPLS EXP는 SP에서 변경할 수 있습니다.	MPLS EXP가 IP Prec/DiffServ에 복사됨
파일	SP QoS 정책에 의해 설정된 MPLS EXP		원래 IP Prec/Difserv는 보존됨(MPLS EXP를 기반으로 하는 이그레스 대기열)
원래 파일			원래 IP Prec/DiffServ가 보존됨(IP Prec/DiffServ에 기반한 이그레스 대기열)

## 관련 정보

- [Cisco IOS 소프트웨어 - MPLS](#)
- [Q&A - 멀티 프로토콜 레이블 스위칭 네트워크의 QoS\(Quality of Service\)](#)
- [Cisco IOS Switching Services 명령 참조, 릴리스 12.2](#)
- [Cisco IOS Quality of Service 솔루션 구성 가이드, 릴리스 12.2](#)
- [Cisco IOS Quality of Service Solutions 명령 참조, 릴리스 12.2](#)
- [MPLS 지원 페이지](#)
- [QoS 지원 페이지](#)
- [Technical Support - Cisco Systems](#)