

# MPLS ネットワーク用の DiffServ トンネリングモード

## 目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[背景説明](#)

[デフォルト動作](#)

[Cisco IOS ソフトウェア 12.2\(13\)T よりも前のコマンドの使用方法和動作](#)

[Cisco IOS ソフトウェア 12.2\(13\)T 以降のコマンドの使用方法和動作](#)

[DiffServ トンネリング モード](#)

[均一モード](#)

[パイプ モード](#)

[ショートパイプ モード](#)

[トンネルモードのまとめ](#)

[関連情報](#)

## 概要

このドキュメントでは、Multiprotocol Label Switching ( MPLS; マルチプロトコル ラベル スイッチング ) ベースのネットワーク環境で使用できる Differentiated Services ( DiffServ; デイファレンシエテッド サービス ) トンネリング モードの実装について説明しています。

## 前提条件

### 要件

このドキュメントの読者は次のトピックについて理解している必要があります。

- MPLS および Virtual Private Network ( VPN; バーチャル プライベート ネットワーク ) 用 MPLS
- IP 優先順位、Type of Service ( ToS; タイプ オブ サービス ) および DiffServ に関連する概念
- Modular QoS Command Line Interface ( MQC; モジュラ QoS コマンドライン インターフェイス ) を使用した Quality of Service ( QoS ) のパケット マーキングとクラス分け

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

- 新しい QoS 機能のための Cisco IOSソフトウェア<sup>®</sup> リリース 12.2(13)T。リリース 12.1(5)T には元の QoS 機能が含まれています。
- MPLS プロバイダー コア ( P ) ルータやプロバイダー エッジ ( PE ) ルータの機能をサポートする Cisco 3660 や 7206 などの 3600 シリーズ以降の任意の Cisco ルータ。

本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、初期 ( デフォルト ) 設定の状態から起動しています。稼働中のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してください。

## [表記法](#)

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

## [背景説明](#)

### [デフォルト動作](#)

次の図には、1つのカスタマー エッジ ( CE ) ルータから別の CE ルータへ MPLS コアを通過してパケットが転送される際の、DiffServ Code Point ( DSCP; DiffServ コード ポイント ) /MPLS Experimental ( EXP ) ビットのデフォルト動作が説明されています。

このセクションでは、以後、「デフォルト動作」の図の中の動作について説明しています。

ラベルのインポジション ( IP → ラベル ) :

- 着信 IP パケットの IP 優先順位が、プッシュされたすべてのラベルの MPLS EXP ビットにコピーされます。
- DSCP ビットの最初の 3 ビットが、プッシュされたすべてのラベルの MPLS EXP ビットにコピーされます。
- この手法は ToS Reflection ( TOS の反映 ) とも呼ばれています。

MPLS 転送 ( ラベル → ラベル ) :

- 転送やインポジションでスワップまたはプッシュされた新しいラベルに EXP がコピーされません。
- ラベルのインポジション ( imposition ) では、現在のラベル スタックに付加される新しいラベルの値で、元のラベルが変更されることはありません。
- ラベルのディスポジション ( disposition ) では、新たに表れるようになったラベルの EXP ビットに、取り外したラベルの EXP ビットはコピーされません。

ラベルのディスポジション ( ラベル → IP ) :

- ラベルのディスポジションを行う際には、新たに表れるようになった IP パケットの IP 優先順位/DSCP フィールドに、取り外したラベルの EXP ビットはコピーされません。

## [Cisco IOS ソフトウェア 12.2\(13\)T よりも前のコマンドの使用方法和動作](#)

IOS リリース 12.2(13)T よりも前のリリースでは、MPLS EXP ビットを変更するには、[set mpls](#)

[experimental](#) コマンドが唯一の方法でした。

## [Cisco IOS ソフトウェア 12.2\(13\)T 以降のコマンドの使用方法和動作](#)

IOS リリース 12.2(13)T 以降のリリースでは、次のオプションを使用できるように `set mpls experimental` コマンドが変更されています。

- [MPLS Experimental を最上設定して下さい](#){*MPLS Exp.値 / qos-group [表表マップ名前]*}
- [MPLS Experimental インポジション](#){*MPLS Exp.値を設定して下さい / qos-group [表表マップ名前]*}

注: 新しい `set mpls experimental topmost` コマンドは、古い `set mpls imposition` コマンドと同等です。

これら 2 つのコマンドを一部の新しいコマンドスイッチとともに使用すると、ラベルのプッシュ、スワップ、ポップ処理時の MPLS EXP ビットの操作がさらに改善されます。これら 2 つのコマンドにより、DiffServ トンネリング モードが使用できます。

## [DiffServ トンネリング モード](#)

Diffserv トンネリング モードには、差別化された QoS をプロバイダーのネットワークで使用可能にする新しい Per-Hop-Behavior ( PHB ) が導入されています。トンネリング モードはネットワークのエッジで定義されるもので、通常は PE のラベル スイッチ ルータ ( LSR ) ( 入口と出口の両方 ) で定義されます。P ルータの変更を行なう必要がある場合もあります; 最高のラベルが Penultimate Hop Popping ( PHP ) によるパケットから取除かれるときまた発生するものが考慮して下さい。最近露出されたラベルにぼんと鳴らされている上ラベルから MPLS EXP 値をコピーすることは必要かもしれません; これはすべてのトンネリング モードに常に適用しません。

一部のケース ( VPN でない普通の MPLS ネットワークなど ) では、ラベルが 1 つだけ付いたパケットが受信されると、最後の P ルータでの PHP 処理で、素の IP パケットが表出される場合があります。この IP パケットが出口 LSR ( PE ) で受信されると、ラベルが付いていないので、MPLS EXP ビットに基づいたパケットの分類ができません。そのような場合には、`explicit-null` ラベルをアドバタイズするように、出口 PE ルータを設定する必要があります。PHP 処理が P ルータで実行されると、ゼロ値のラベルが送信されます。この特別なラベルを使用すれば、通常のラベルが付いたパケットとして EXP ビットをマーキングできるので、出口 PE ルータで正しく分類できるようになります。

Diffserv 仕様の MPLS ネットワーク サポートでは、次のトンネリング モードが定義されます。

- [均一](#)
- [パイプ](#)
- [ショートパイプ](#)

次のセクションでは、各トンネリング モードを個別に詳しく検証し、各トンネリング モードの設定方法を示す例を紹介しています。IP 優先順位を MPLS EXP ビットに完全にマッピングする例も含まれています。顧客ごとに異なる多数の QoS パラメータやトンネリング モードを設定することも可能です。

注: この設定例は MPLS VPN 固有のものではなく、普通の MPLS ネットワークや Carrier supported Carrier ( CsC ) ネットワークにも適用できます。また、使用するネットワークが別のネットワークとは異なったり、多数の異なる QoS パラメータやトンネリング モードが使用されたりする場合があります。

## 均一モード

DiffServ トンネリングの均一モードには、エンドツーエンドに到達する QoS のレイヤが 1 つだけ存在します。入口 PE ルータ (PE1) では、付加されたラベルの MPLS EXP ビットに着信 IP パケットの DSCP がコピーされます。EXP ビットがコアを通過すると、中継 P ルータによって EXP ビットが変更される場合もあれば、そうでない場合もあります。この例では、P1 という P ルータで最上部のラベルの EXP ビットが変更されています。出口 P ルータ (P2) では、PHP (Penultimate-Hop-Pop) の後に、新しく表れたラベルの EXP ビットに EXP ビットをコピーします。最後に、出口 PE ルータ (PE2) で、新しく表れた IP パケットの DSCP ビットに EXP ビットをコピーします。

均一モードの設定例 :

```
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
!
```

## パイプモード

DiffServ トンネリングのパイプモードでは、QoS のレイヤを 2 つ使用します。

1. データの元の QoS。この QoS はコアを通過しても変わりません。
2. コアごとの QoS。この QoS は元の IP パケットの QoS とは別の QoS です。このコアごとの QoS の PHB は、エンドユーザに対しては常に透過的です。

MPLS コアのエッジにパケットが到達すると、最近取り外されたラベルの EXP ビットから取得した MPLS PHB に基づいて、出口 PE ルータ (PE2) が、新たに表れた IP パケットを出力キューに分類します。

パイプモードの設定例：

### 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
!

```

## ショートパイプモード

DiffServ トンネリングのショートパイプモードでは、同じ規則と手法がコア全体で使用されます。出口 PE ルータ ( PE2 ) の処理に違いがあります。この処理では、新たに表れた IP パケットを、この IP パケットの DSCP 値から取得した IP PHB に基づいて発信キューに分類します。

ショートパイプモードの設定例：

### 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
!

```

## トンネルモードのまとめ

IP パケットやラベルの付いたパケットにネットワークのさまざまな段階で適用されるさまざまな処理を次の表にまとめておきます。

トンネルモード	IP →ラベル	ラベル→ラベル	ラベル→ IP
均一	IP 優先順位や DiffServ を MPLS EXP にコピーする ( SP によって変更される場合もある )	MPLS EXP が SP によって変更される場合がある	MPLS EXP が IP 優先順位や DiffServ にコピーされる
パイプ	MPLS EXP が SP の QoS ポリシーによって設定される		元の IP 優先順位や DiffServ が維持される ( MPLS EXP に基づく出口キューイング )
ショートパイプ			元の IP 優先順位や DiffServ が維持される ( IP 優先順位や DiffServ に基づく出口キューイング )

## 関連情報

- [Cisco IOS ソフトウェア : MPLS](#)
- [Q&A : Multi-Protocol Label Switching \( MPLS; マルチプロトコル ラベル スイッチング \) ネットワークの QoS](#)
- [Cisco IOS スイッチング サービス コマンド リファレンス、リリース 12.2](#)
- [Cisco IOS QoS ソリューション コンフィギュレーション ガイド、リリース 12.2](#)
- [Cisco IOS QoS ソリューション コマンド リファレンス、リリース 12.2](#)

- [MPLS に関するサポートページ](#)
- [QoS に関するサポート ページ](#)
- [テクニカルサポート - Cisco Systems](#)