

DiffServ Tunneling Modes for MPLS Networks

Document ID: 47815

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Background Information

- Default Behavior
- Pre Cisco IOS Software 12.2(13)T Command Usage and Behavior
- Post Cisco IOS Software 12.2(13)T Command Usage and Behavior

DiffServ Tunneling Modes

- Uniform Mode
- Pipe Mode
- Short-Pipe Mode
- Tunnel Mode Summary

Related Information

Introduction

This document describes the implementation of Differentiated Services (DiffServ) Tunneling Modes available for Multiprotocol Label Switching (MPLS) based network environments.

Prerequisites

Requirements

Readers of this document should have knowledge of these topics:

- MPLS and MPLS for Virtual Private Networks (VPN)
- Concepts relating to IP precedence, Type of Service (ToS) and DiffServ
- Quality of Service (QoS) Packet Marking and Classification using Modular QoS Command Line Interface CLI (MQC)

Components Used

The information in this document is based on these software and hardware versions:

- Cisco IOS Software® Release 12.2(13)T for new QoS features. Release 12.1(5)T includes original QoS features.
- Any Cisco router from the 3600 series or later, such as the Cisco 3660 or 7206 that support MPLS provider core (P) router/provider edge (PE) router functionality.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

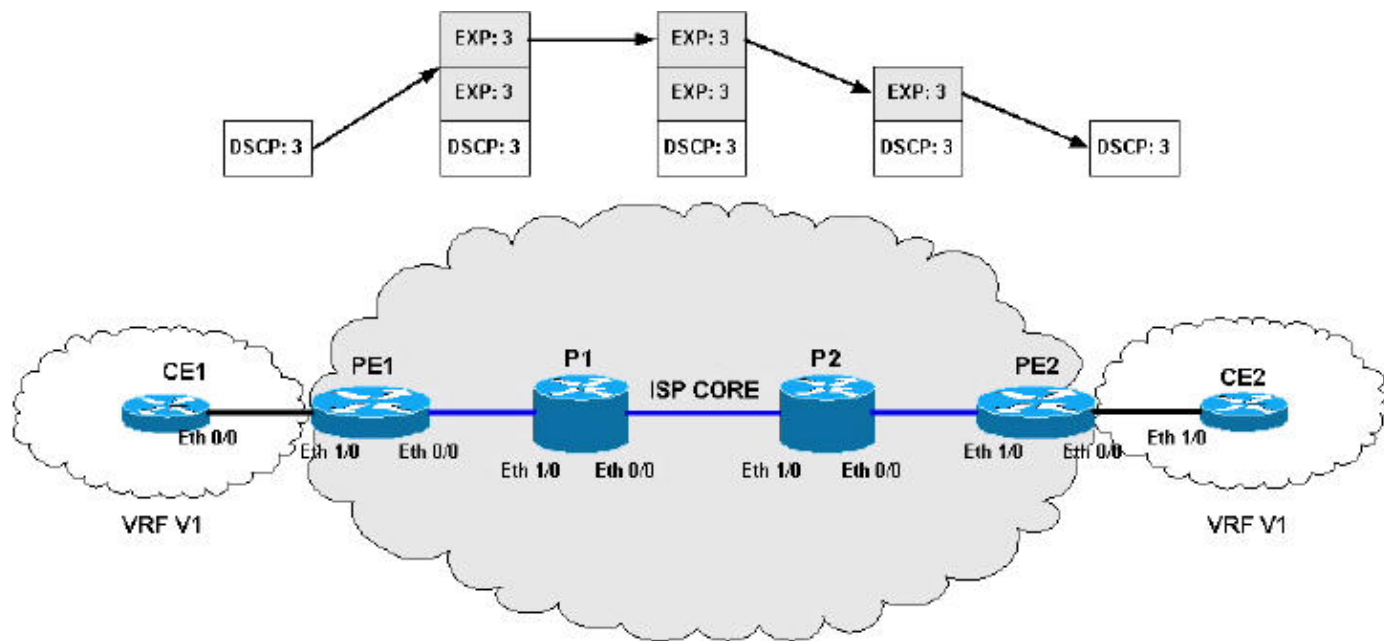
Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Background Information

Default Behavior

This diagram describes the default behavior of the DiffServ Code Point (DSCP)/MPLS Experimental (EXP) bits as a packet travels from one customer edge (CE) router to another CE router across an MPLS core:



This remainder of this section describes the activity within the Default Behavior diagram.

Imposition of Label (IP > Label):

- The IP precedence of the incoming IP packet is copied to the MPLS EXP bits of all pushed label(s).
- The 1st three bits of the DSCP bit is copied to the MPLS EXP bits of all pushed label(s).
- This technique is also known as ToS Reflection.

MPLS forwarding (Label > Label):

- The EXP is copied to the new labels that are swapped/pushed during forwarding or imposition.
- At label imposition, the underlying labels are not modified with the value of the new label being added to the current label stack.
- At label disposition, the EXP bits are not copied down to the newly exposed label EXP bits.

Disposition of Label (Label > IP):

- At label disposition the EXP bits are not copied down to the IP precedence/DSCP field of the newly exposed IP packet.

Pre Cisco IOS Software 12.2(13)T Command Usage and Behavior

Before IOS Release 12.2(13)T, the **set mpls experimental** command was the only method available to modify the MPLS EXP bits.

Post Cisco IOS Software 12.2(13)T Command Usage and Behavior

With IOS Release 12.2(13)T and later, the **set mpls experimental** command is modified to allow these options:

- **set mpls experimental topmost** *{mpls-exp-value | qos-group [table table-map-name]}*
- **set mpls experimental imposition** *{mpls-exp-value | qos-group [table table-map-name]}*

Note: The new **set mpls experimental topmost** command is equivalent to the old **set mpls imposition** command.

These two commands, in combination with some new command switches, allow better control of MPLS EXP bits manipulation during label push, swap, and pop operations. These two commands allow you to use DiffServ Tunneling Modes.

DiffServ Tunneling Modes

Diffserv Tunneling Modes introduces a new Per-Hop-Behavior (PHB), which allows differentiated QoS in a providers network. The tunneling mode is defined at the edge of the network, normally in the PE label switch routers (LSRs) (both ingress and egress). You may need to make changes in the P routers; you must also consider what occurs when the topmost label is removed from a packet due to Penultimate-Hop-Popping (PHP). It may be necessary to copy the MPLS EXP value from the top label that is being popped to the newly exposed label; this does not always apply to all tunneling modes.

In some cases (for example, a plain non-VPN MPLS network), the PHP action on the final P router can expose a plain IP packet when a packet with only one label is received. When this IP packet is received by the egress LSR (PE), it is not possible to classify the packet based on the MPLS EXP bits because there is no label now. In these situations, you must configure the egress PE router to advertise an **explicit-null** label. When the PHP action is performed on the P router, a label with a value of zero is sent, and with this special label you can mark the EXP bits as normally labeled packets, allowing the correct classification on the egress PE router.

The MPLS network support of Diffserv specification defines these tunneling modes:

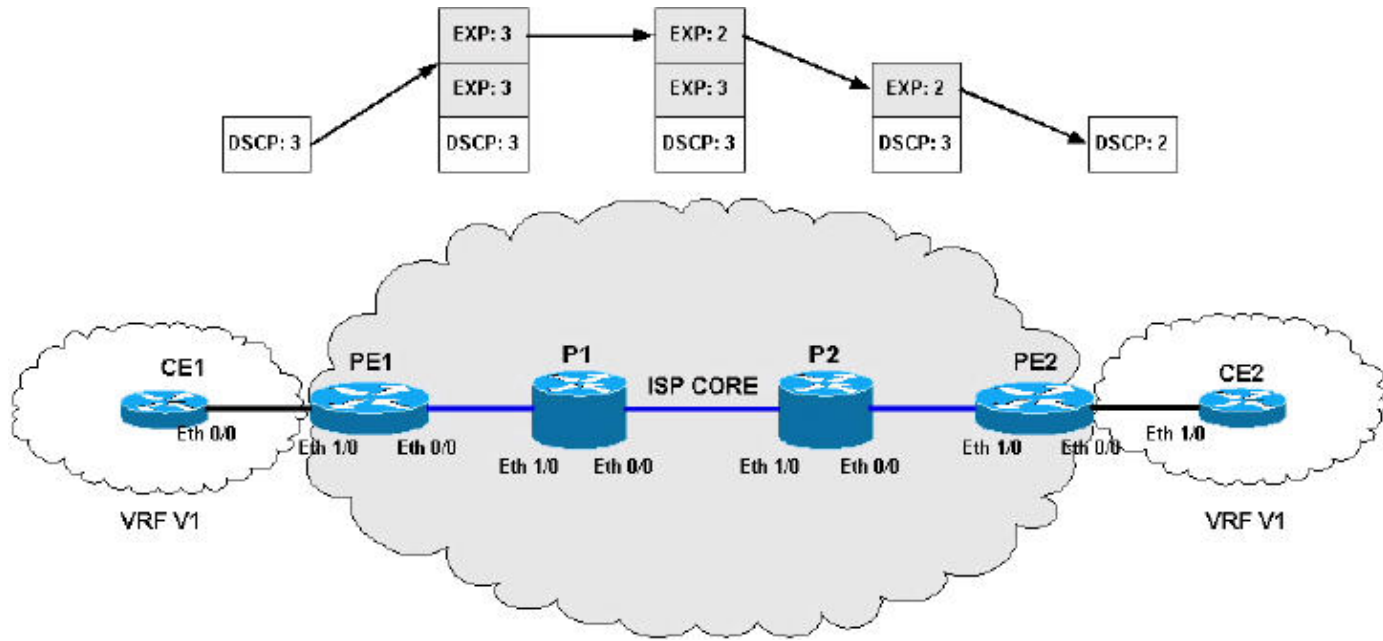
- Uniform
- Pipe
- Short-Pipe

The next sections examine each tunneling mode separately, and provide examples to show how each tunneling mode can be configured. The examples include a full mapping of IP Precedence to MPLS EXP bits. It is possible to have a number of different QoS parameters and tunneling modes for each customer.

Note: The configuration examples are not specific for MPLS VPN, and are applicable for plain MPLS networks and Carrier supported Carrier (CsC) networks. It is also possible that your network can vary from another network many different QoS parameters and tunneling modes can be used.

Uniform Mode

DiffServ Tunneling Uniform Mode has only one layer of QoS, which reaches end-to-end. The ingress PE router (PE1) copies the DSCP from the incoming IP packet into the MPLS EXP bits of the imposed labels. As the EXP bits travel through the core, they may or may not be modified by intermediate P routers. In this example, P router P1 modifies the EXP bits of the top label. At the egress P router (P2) we copy the EXP bits to the EXP bits of the newly exposed label after the PHP (Penultimate-Hop-Pop). Finally at the egress PE router (PE2) we then copy the EXP bits to the DSCP bits of the newly exposed IP packet.



Uniform mode configurations:

```

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
!

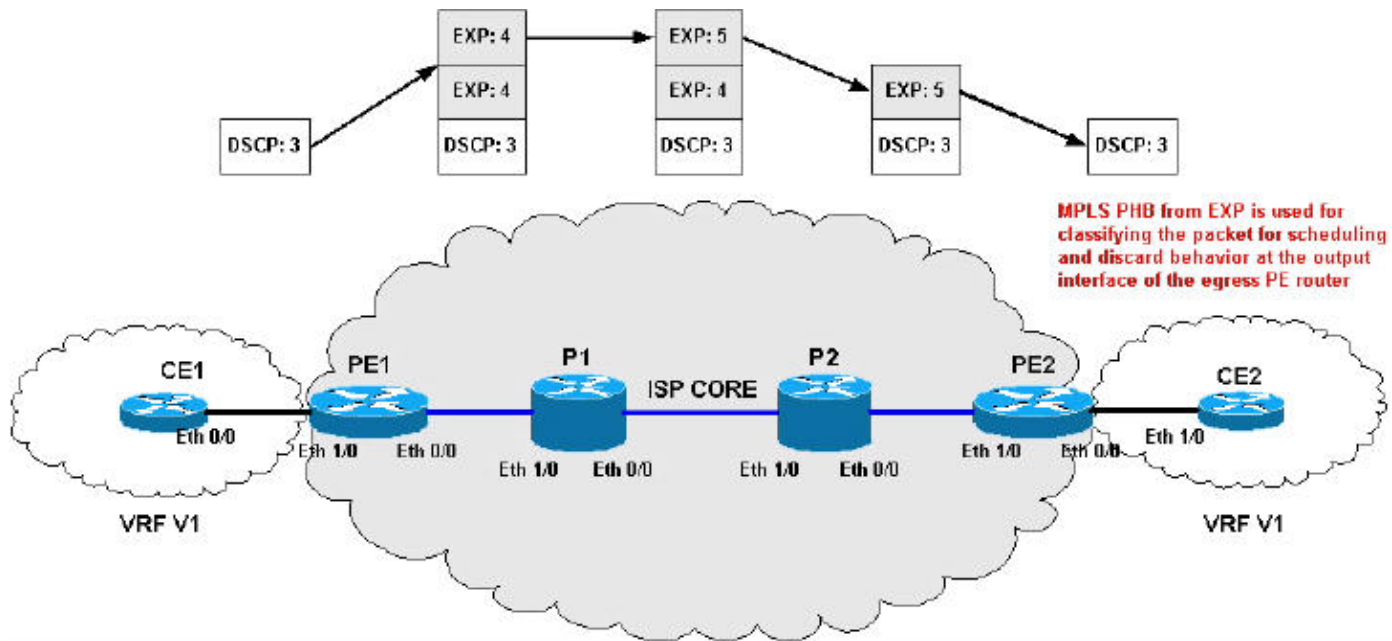
```

Pipe Mode

Diffserv Tunneling Pipe Mode uses two layers of QoS:

1. An underlying QoS for the data, which remains unchanged when traversing the core.
2. A per-core QoS, which is separate from that of the underlying IP packets. This per-core QoS PHB remains transparent to end users.

When a packet reaches the edge of the MPLS core, the egress PE router (PE2) classifies the newly exposed IP packets for outbound queuing based on the MPLS PHB from the EXP bits of the recently removed label.



Pipe mode configurations:

```

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

```

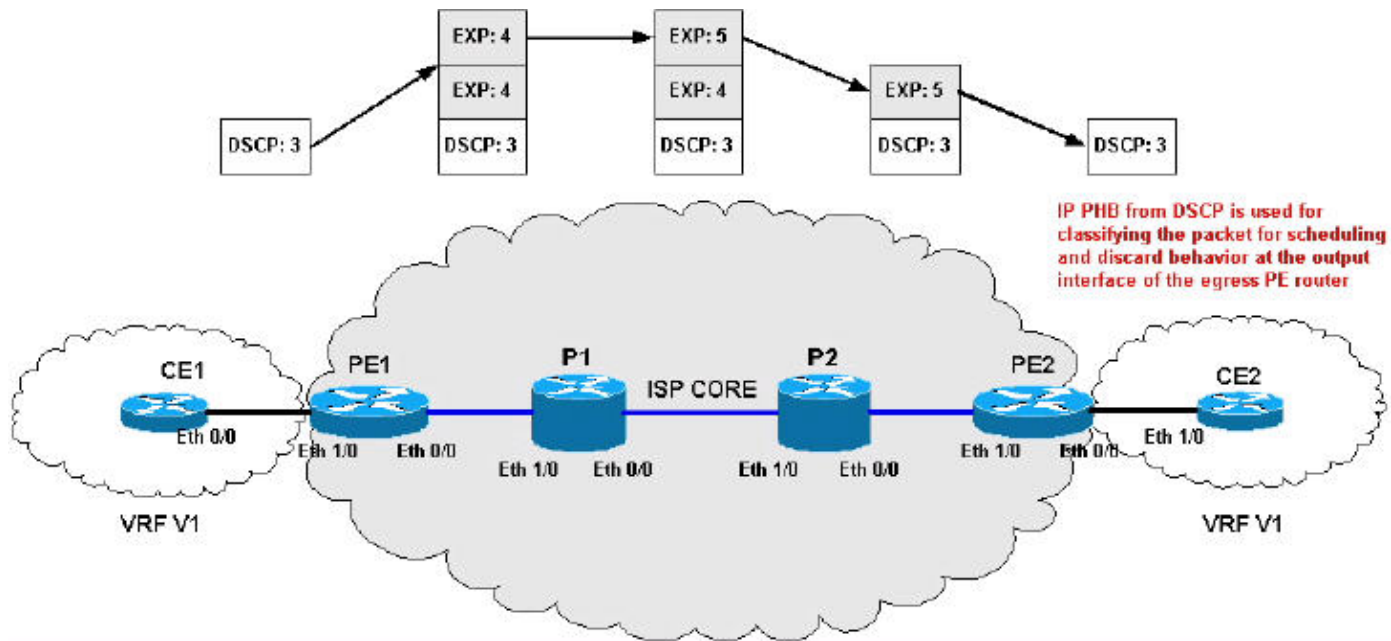
```

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
!

```

Short-Pipe Mode

Diffserv Tunneling Short-pipe Mode uses the same rules and techniques across the core. The difference is at the egress PE router (PE2) you classify the newly exposed IP packets for outbound queuing based on the IP PHB from the DSCP value of this IP packet.



Short-pipe mode configurations:

```

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
!

```

Tunnel Mode Summary

This table summarizes the various actions that applied to IP or labeled packets at various stages in the network:

Tunneling Mode	IP > Label	Label > Label	Label > IP
Uniform	Copy IP Prec/DiffServ into MPLS EXP (may be changed by the SP also)		MPLS EXP copied to IP Prec/DiffServ
Pipe		MPLS EXP may be changed by SP	Original IP Prec/Diffserv preserved (egress queuing based on MPLS EXP)
Short-Pipe	MPLS EXP set by the SP QoS policy		Original IP Prec/Diffserv preserved (egress queuing based in IP Prec/DiffServ)

Related Information

- [Cisco IOS Software – MPLS](#)
- [Q &A – Quality of Service for Multi-Protocol Label Switching Networks](#)
- [Cisco IOS MPLS Configuration Guide, Release 12.2](#)
- [Cisco IOS Switching Services Command Reference, Release 12.2](#)
- [Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2](#)
- [Cisco IOS Quality of Service Solutions Command Reference, Release 12.2](#)
- [MPLS Support Page](#)
- [QoS Support Page](#)
- [Technical Support – Cisco Systems](#)

