# QoS: Enhancement to Subinterface QoS Policy Definitions (L3/nC/nD) Including Proportional Bandwidth Allocation

L3/nC/nD is a service provider Quality of Service (QoS) reference model that describes methods by which a physical interface can be provisioned by a service provider to multiple customers with one connection provisioned per customer. This feature document describes how a subset of Cisco 12000 Series ISE line cards implement elements of L3/nC/nD, in particular, proportional bandwidth allocation for sharing excess bandwidth between customers.

**Feature History for L3/nC/nD Including Proportional Bandwidth Allocation**

| Release | Modification |
|---|---|
| 12.0(28)S | This feature was introduced for Cisco 12000 Series ISE linecards. |

### Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

## Contents




Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

# Prerequisites

- Knowledge of the Cisco Modular Quality of Service Command line Interface (MQC) to create Quality of Service class maps and policy maps. See the "Related Documents" section on page 15 for further information.
- Understanding of Frame Relay interface configuration, traffic shaping and policing.

The QoS: Enhancement to Subinterface QoS Policy Definitions (L3/nC/nD) Including Proportional Bandwidth Allocation is supported on the following Cisco 12000 Series ISE line cards:

### Supported Cisco 12000 Series Internet Services Engine (E3) Line Cards

- 1-Port Channelized OC-12/STM-4 (DS1/E1) ISE Line Card
  Product Number: CHOC12/DS1-IR-SC=
- 1-Port Channelized OC-48/STM-16 (DS3/E3, OC-3c/STM-1c, OC-12c/STM-4c) POS/SDH ISE Line Card
  Product Number: CHOC48/DS3-SR-SC=
- 4-Port Channelized OC-12/STM-4 (DS3/E3, OC-3c/STM-1c) POS/SDH ISE Line Card
  Product Number: 4CHOC12/DS3-I-SCB=
- 16-Port OC-3c/STM-1c POS/SDH ISE Line Card
  Product Numbers: 6OC3X/POS-I-LC-B=, 16OC3X/POS-M-MJ-B=

# Information About QoS: Enhancement to Subinterface QoS Policy Definitions (L3/nC/nD) Including Proportional Bandwidth Allocation

The L3/nC/nD is a reference model developed for service provider customers to describe methods to provide a physical interface for Layer 3 services (L3) to any number of customers (nCustomers) with one logical subinterface connection provisioned per customer (nDLCIs). The customer-specific Modular Quality of Service Command line (MQC) configuration is also attached to the subinterface.

Each customer can classify their traffic into one or more classes defined by MQC class-maps with match criteria (such as match ip precedence, or match access-group). Each of these classes may have queuing and non-queuing QoS features applied to them individually.

This L3/nC/nD model is the most widely deployed model for Layer 3 services today. The service characteristics of the model are as follows:

- Each customer buys an aggregate service (across all classes) at a defined rate; the service provider guarantees this service at the edge by shaping each Datalink Connection Identifier (DLCI) to the contracted per-customer aggregate rate.
- The customer aggregate can then be divided into bandwidth available for the different classes. When the aggregate per-customer rate is exceeded, backpressure is directed into a queuing scheme that governs the per-class packet differentiation and drop behaviour.

# How to Implement L3/nC/nD Including Proportional Bandwidth Allocation with Frame Relay

The L3/nC/nD reference model advances the following configuration scenario for Frame Relay subinterface traffic shaping. The following configuration is for the Customer Edge (CE) outbound towards the Provider Edge (PE) and on the PE outbound towards the CE:

### L3nCnD Frame Relay Subinterface Traffic-Shaping Reference Configuration Model

```
policy-map parent
  class class-default
    shape average <cir> <nb> <be>
    service-policy child
!
policy-map child
  class voip
    {voip-sub-model}
  class bus
    {data-sub-model}
  class class-default
    {data-sub-model}
!
map-class frame-relay frts
  service-policy output parent
!
interface SerialX/Y
  encapsulation frame-relay IETF
!
interface SerialX/Y.1 point-to-point
  frame-relay class frts
  frame-relay interface-dlci <dlci>
```

In the reference configuration, traffic shaping on a Frame Relay subinterface is governed by an hierarchical service policy named “parent” which is applied to every subinterface configured to be in the Frame Relay class “frts.” No policy is applied to the main interface. In this example, the “child” policy map defines three typical traffic classes. It is assumed that each class in the child policy is configured to police its corresponding traffic.

The Cisco 12000 Series ISE line cards specified in the “Prerequisites” section on page 2, do not support the implementation of hierarchical service policies applied to subinterfaces. The following configuration is supported instead:

### Cisco 12000 ISE Line Card Frame Relay Subinterface Traffic-Shaping Configuration Model

```
policy-map child-dummy
!
policy-map port-shape
  class class-default
    shape average <cir> <nb> <be>
    service-policy child-dummy
!
policy-map subinterface-policy
  class voip
    {voip-sub-model}
  class bus
    {data-sub-model}
  class class-default
    {data-sub-model}
!
map-class frame-relay frts
```

```
  service-policy output subinterface-policy
!
interface SerialX/Y
  encapsulation frame-relay IETF
  service-policy output port-shape
!
interface SerialX/Y.1 point-to-point
  frame-relay class frts
  frame-relay interface-dlci <dlci>
```

In the ISE Line Card configuration example, a service policy is applied to shape the main interface, with the subinterface configured as a member of a Frame Relay class. Alternatively, the subinterface service policy can be applied directly to each subinterface.

The null service policy (child-dummy) specified in the service policy attached to the main interface (port-shape) signals the software that port-shaping rather than class shaping is being configured.

The following restrictions for the specified ISE linecards follow from this example:

- Shaping in the parent class on a subinterface is not supported.
- Bandwidth sharing of excess bandwidth is possible between all classes on a main interface, but it is not possible to share excess bandwidth within classes in a policy.

# Configuring Class Maps

An implementation of the L3/nC/nD feature may require configuring multiple class maps. To configure class maps, complete the following procedure.

## Restrictions

- If a service policy is attached to a Frame Relay subinterface, then a valid policy, flat or hierarchical, cannot be added to the main interface.
- Only a hierarchical policy with a dummy child policy can be attached to the main interface (port-shaping).
- If a policy map is configured with a bandwidth percentage method, then the subinterface bandwidth should be configured on the Frame Relay subinterface.
- The **bandwidth remaining percent** command is not supported on the Frame Relay subinterface.
- Configuration of port-shaping for L3/nC/nD is not supported with map-class configuration.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match ip precedence** *number*
5. **match mpls experimental exp**
6. **exit**
7. Repeat Steps 3 through 6 for each class that you want to create.
8. **exit**

9. **show class-map**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **class-map** [**match-all** \| **match-any**] *class-map-name*<br><br>**Example:**<br>`Router(config)# class-map match-all prec1` | Specifies the name of the class map to be created and enters class-map configuration mode.<br>• Enter class map name.<br>If the match-all or match-any keyword is not specified, traffic must match all the match criteria to be classified as part of the traffic class. |
| **Step 4** | **match ip precedence** *number*<br><br>**Example:**<br>`Router#(config-cmap)# match ip precedence 1` | Configures the class map created above to match traffic based on the IP precedence number or the packet. |
| **Step 5** | **match mpls experimental exp**<br><br>**Example:**<br>`Router#(config-cmap)# match mpls experimental 1` | Designates the value to which the MPLS bits are set if the packets match the specified policy map. |
| **Step 6** | **exit**<br><br>**Example:**<br>`Router#(config-cmap)# exit` | (Optional). Exits class-map configuration mode. |
| **Step 7** | Repeat Steps 3 through 6 for each class that you want to create. | — |
| **Step 8** | **show class-map**<br><br>**Example:**<br>`Router# show class-map` | Displays all class maps and their match criteria. |

# Configuring the Policy Maps with Port-Shaping and Attaching them to a Main and SubInterface

After a policy map is created, the next step is to attach the policy map to an interface. Policy maps can be attached to either the input or output direction of the interface.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *name*
4. **exit**
5. **policy-map** *name*
6. **class** *name*
7. **shape average**
8. **service-policy name**
9. **exit**
10. **exit**
11. **policy-map**
12. **class**
13. **bandwidth**
    `bandwidth percent 15`
14. **queue-limit**
    `queue-limit 1500 packets`
15. **exit**

To add additional classes, repeat steps 12 through 15.

16. **map-class frame-relay frts**
17. **service-policy output port-shape**
18. **exit**
19. **exit**
20. **interface pos1/0**
21. **no ip address** *address*
22. **encapsulation {frame-relay | hdlc | ppp}**
23. **no keepalive**
24. **service-policy** {**history** | **input** | **output**} *name*
25. **exit**
26. **interface pos1/0.1** {**point to point** | **mulitpoint**}
27. **ip address 172.16.1.1 255.255.255.0**
28. **frame-relay class** *name*
29. **frame-relay interface-dlci** *dlci*

30. **exit**
31. **show running-config interface serial0**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-name*<br><br>**Example:**<br>`Router(config)# policy-map child-dummy` | Specifies the name of the policy map to be created or modified. Enters policy-map configuration mode.<br>• Enter the policy map name.<br>In this example, the policy map, child-dummy, is an empty policy map, that when called in a parent policy map signals the software that port-shaping is being configured rather than class shaping. |
| **Step 4** | **exit**<br><br>**Example:**<br>`Router(config-pmap)# exit` | Exits policy-map configuration mode. |
| **Step 5** | **policy-map** *policy-name*<br><br>**Example:**<br>`Router(config)# policy-map port-shape` | Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.<br>• Enter the policy map name.<br>In this example, the policy map port-shape is the service policy to be applied to the main interface. |
| **Step 6** | **class** *class-name*<br><br>**Example:**<br>`Router(config-pmap)# class class-default` | Specifies a class map name. Enters policy-map class configuration mode.<br>• Enter the class name. |
| **Step 7** | **shape** {**average** \| **peak**} **mean-rate** [**burst-size** [**excess-burst-size**]]<br><br>**Example:**<br>`Router(config-pmap-c)# shape average 5000000` | Specifes the traffic shaping method for the class being configured. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | **service-policy** *policy-map-name*<br><br>**Example:**<br>`Router(config-pmap-c)# service-policy child-dummy` | Specifies the name of a traffic policy to be used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another).<br>• Enter the class name.<br>In this example, specifying the empty service policy, child-dummy, indicates to that this policy map is for port-shaping rather than class shaping. |
| **Step 9** | **exit**<br><br>**Example:**<br>`Router#(config-pmap-c)# exit` | (Optional) Exits policy-map class configuration mode. |
| **Step 10** | **exit**<br><br>**Example:**<br>`Router#(config-pmap-c)# exit` | (Optional) Exits policy-map configuration mode. |
| **Step 11** | **policy-map** *policy-name*<br><br>**Example:**<br>`Router#(config-pmap)# policy-map foo` | Specifies the name of the policy map to be created or modified. Enters policy-map configuration mode.<br>• Enter the policy map name.<br>In this example, the policy map, child-dummy, is an empty policy map, that when called in a parent policy map signals the software that port-shaping is being configured rather than class shaping. |
| **Step 12** | **class** {*class-name* \| **class-default**}<br><br>**Example:**<br>`Router#(config-pmap)# class C4` | Specifies the name of a traffic policy to be used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another).<br>• Enter a class name. |
| **Step 13** | **bandwidth** {*bandwidth-kbps* \| **percent** *percent* \| **remaining percent** *percent*}<br><br>**Example:**<br>`Router#(config-pmap-c)# bandwidth percent 15` | Specifies a minimum bandwidth guarantee to a traffic class in periods of congestion. A minimum bandwidth guarantee can be specified in kbps , by a percentage of the overall available bandwidth, or a pecentage of the remaining available bandwidth.<br>• Enter a bandwidth or a percentage parameter. |
| **Step 14** | **queue-limit** {**cells** \| **ms** \| **packets** \| **us**}<br><br>**Example:**<br>`Router#(config-pmap-c)# queue-linit packets 1500 packets` | Specifies the maximum size of the queue for the traffic class in number of packets, number of cells, in milliseconds, or in microseconds of traffic. The default unit is packets. |
| **Step 15** | **exit**<br><br>**Example:**<br>`Router#(config-pmap-c)# exit` | (Optional) Exits policy-map configuration mode. |
| **Step 16** | To add more classes to the policy map, repeat steps 12 through 15. | |

| | Command or Action | Purpose |
|---|---|---|
| **Step 17** | **map-class frame-relay** *map-class-name*<br><br>**Example:**<br>`Router#(config)# map-class frame-relay frts` | Specifies a Frame Relay map class, and enters map-class configuration mode. |
| **Step 18** | **service-policy** {**input** \| **output**} *policy-map-name*<br><br>**Example:**<br>`Router#(config-map-class)# map-class output frts` | Specifies the name of the policy map to be attached to the input or output direction of the interface.<br><br>**Note** Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according your network configuration. When using the service-policy command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for your network configuration. |
| **Step 19** | **exit**<br><br>**Example:**<br>`Router#(config-map-class)# exit` | (Optional) Exits map-class configuration mode. |
| **Step 20** | **interface pos 1/0**<br><br>**Example:**<br>`Router(config)# interface pos 1/0` | Configures an interface (or subinterface) type and enters interface configuration mode.<br><br>• Enter the interface type number. |
| **Step 21** | **no ip address**<br><br>**Example:**<br>`Router(config-inf)# no ip address` | Deletes the IP address of the main interface. |
| **Step 22** | **encapsulation {frame-relay** \| **hdlc** \| **ppp}**<br><br>**Example:**<br>`Router(config-if)# encapsulation frame-relay` | Specifies the level 2 encapsulation method. |
| **Step 23** | **no keepalive**<br><br>**Example:**<br>`Router(config-if)# no keepalive` | Disables Local Management Interface (LMI) processing. |
| **Step 24** | **service-policy** {**input** \| **output**} *policy-map-name*<br><br>**Example:**<br>`Router(config-inf)# service-policy output parent` | Attaches the specified service policy map to the output or the input interface and enables the Low Latency Queue (LLQ) for Frame Relay.<br><br>• Enter the policy map name. |
| **Step 25** | **exit**<br><br>**Example:**<br>`Router(config-inf)# exit` | (Optional). Exits interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 26** | **interface pos1/0.1** {**point-to-point** \| **multipoint**}<br><br>**Example:**<br>`Router(config)# interface pos1/0.1 point-to-point` | Configures subinterface and specifies the connection type. Enters interface configuration mode.<br>• Enter the subinterface type number.<br>**Note** The serial subinterface configuration syntax varies with the line card. See the software configuration guide of your line card to determine the correct syntax. |
| **Step 27** | **ip address** *address mask*<br><br>**Example:**<br>`Router(config-inf)# ip address 172.16.1.1 255.255.255.0` | Specifies the IP address and subnet mask of the interface. |
| **Step 28** | **frame-relay class** *name*<br><br>**Example:**<br>`Router(config-inf)# frame-relay class frts` | Associates a map class with an interface or subinterface.s |
| **Step 29** | **frame-relay interface-dlci** *dlci*<br><br>**Example:**<br>`Router(config-inf)# frame-relay interface-dlci 202` | Specifies the data-link connection identifier (DLCI) associated with the subinterface. |
| **Step 30** | **exit**<br><br>**Example:**<br>`Router(config-inf)# exit` | (Optional). Exits interface configuration mode to global configuration mode. |
| **Step 31** | **exit**<br><br>**Example:**<br>`Router(config)# exit` | (Optional). Exits global configuration mode to priveleged EXEC mode. |
| **Step 32** | **show running-config interface serial0**<br><br>**Example:**<br>`Router# show running-config interface serial0` | (Optional). Displays parameters and the current policies attached to the specified interface.<br>• Use this command to verify the configuration. |

# Examples

### Configuring the Class Maps

```
Router(config)# class-map match-any C1
Router#(config-cmap)# match ip precedence 1
Router#(config-cmap)# match mpls experimental 1

Router(config)# class-map match-any C6
Router#(config-cmap)# match ip precedence 6
Router#(config-cmap)# match access-group 1

Router(config)# class-map match-any C5
```

```
Router#(config-cmap)# match ip precedence 5

Router(config)# class-map match-any C4
Router#(config-cmap)# match ip precedence 4
Router#(config-cmap)# match ip precedence 3
Router#(config-cmap)# match ip precedence 2
```

### Configuring the Policy Maps With Port-Shaping:

```
Router(config)# Policy Map child-dummy

Router(config)# policy map port-shape /* policy on main-interface */
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 500000000
Router(config-pmap-c)# service-policy child-dummy

Router(config)# policy map foo /* policy attached to individual FRSI */
Router(config-pmap)# Class C4
Router(config-pmap-c)# bandwidth percent 15
Router(config-pmap-c)# queue-limit 1500 packets
Router(config-pmap-c)# exit

Router(config-pmap)# Class C5
Router(config-pmap-c)# match ip precedence 1
Router(config-pmap-c)# exit

Router(config-pmap)# Class C6
Router(config-pmap-c)# bandwidth percent 45
Router(config-pmap-c)# queue-limit 2500 packets
Router(config-pmap-c)# exit

Router(config-pmap)# Class class-default
Router(config-pmap-c)# bandwidth percent 10

Router(config)# map-class frame-relay frts
Router(config-map-class)# service-policy output foo
Router(config-map-class)# exit
```

### Attaching Policies to the Main and Subinterfaces

```
Router(config)# interface pos0/1
Router(config-if)# no ip address
Router(config-if)# encapsulation frame-relay IETF
Router(config-if)# no keepalive
Router(config-if)# service-policy output port-shape
Router(config-if)# exit
!
Router(config)# interface pos0/1.1 point-to-point
Router(config-subif)# ip address 172.16.1.1 255.255.255.0
Router(config-subif)# frame-relay class frts
Router(config-subif)# frame-relay interface-dlci 102
Router(config-subif)# cntl-Z
```

### Verifing the Configuration

```
Router# show running-config interface pos1/0
Building configuration...
Current configuration : 156 bytes
!
interface POS1/0
no ip address
no ip directed-broadcast
encapsulation frame-relay
service-policy output port-shape
```

```
end

Router# show running-config interface pos1/0.1
Building configuration...
Current configuration : 156 bytes
!
interface POS1/0.1 point-to-point
bandwidth 6000
!
! Notice the bandwidth configuration under the Subinterface
!
!ip address 192.168.14.1 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 16
service-policy output foo
end
Without port-shaping (policy attached only on FRSI):
Policy Map test
Class C6
bandwidth percent 60
queue-limit 2000 packets
Class C5
bandwidth percent 30
queue-limit 1500 packets
Class C4
bandwidth percent 10
queue-limit 1000 packets
Verify
GSR# show run int pos1/0.1
Building configuration...
Current configuration : 156 bytes
!
interface POS1/0.1 point-to-point
bandwidth 6000 ..................> Notice the bandwidth configuration
under the Sub-interface
ip address 192.168.14.1 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 16
service-policy output test
end
```

### Other Show Commands

```
show class-map
show policy-map
show policy-map interface <x>
```

# Configuration Examples for L3/nC/nD Including Proportional Bandwidth Allocation

The sample configurations in this section are derived from actual service provider configurations. The following CLIs were altered to display the new syntax available for the supported line cards in Cisco IOS Release 12.0(28)S.

- shape average percent <%> <time> ms
- random-detect <num> <time> ms <time> ms
- random-detect <num> <time> ms <time> ms
- police cir percent <%> [bc <ms> ms]

**Note** The subinterface **bandwidth** *kbps* command is essential for correct functionality of the percent and time-based configurations for the supported line card subinterfaces. The subinterface **bandwidth** command does not guarantee a minimum bandwidth to the subinterface. It is used as a hint.

# L3/nC/nD Service Provider Configuration Example 1

```
class-map match-any real-time
  match ip dscp 46
  match access-group <x>
class-map match-any bursty-low
  match ip dscp 18 20
  match access-group <x>
class-map match-any bursty-high
  match ip dscp 26 28
  match access-group <x>
!
policy child-dummy
!
policy parent-out
  class class-default
    shape average percent <%> <ms> ms
    service-policy child-dummy
policy content-vpn
  class real-time
    priority
    police cir percent <%> [bc <ms> ms] ! using subinterface bandwidth
  class bursty-high
    bandwidth percent <%> ! using subinterface bandwidth
    random-detect [precedence-based | dscp-based | discard-class-based
    random-detect {precedence | dscp | discard-class} <num> <min_th> ms <max_th> ms [mpd]
! using subinterface bandwidth
  class bursty-low
    bandwidth percent <%>
    random-detect [precedence-based | dscp-based | discard-class-based
    random-detect {precedence | dscp | discard-class} <num> <min_th> ms <max_th> ms [mpd]
! using subinterface bandwidth
  class class-default ! best effort
    bandwidth percent <%>
    shape average percent <%> <ms> ms
    random-detect [precedence-based | dscp-based | discard-class-based
    random-detect {precedence | dscp | discard-class} <num> <min_th> ms <max_th> ms [mpd]
! using subinterface bandwidth
map-class frame-relay content-vpn
  service-policy output content-vpn-out
map-class frame-relay internet
  service-policy output internet-out
map-class frame-relay nms-vpn
  service-policy output nms-vpn-out
interface POS3/0
  encapsulation frame-relay
  service-policy output parent-out
end
interface POS3/0.100 point-to-point
  frame-relay interface-dlci 100
  description e.g. Content VPN
  bandwidth <kbps>
  ip vrf forwarding content-vpn
  frame-relay class content-vpn
end
```

```
interface POS3/0.101 point-to-point
  frame-relay interface-dlci 101
  description e.g. Internet service
  bandwidth <kbps>
  frame-relay class internet
end
interface POS3/0.102 point-to-point
  frame-relay interface-dlci 102
  description e.g. NMS VPN
  bandwidth <kbps>
  ip vrf forwarding nms-vpn
  frame-relay class nms-vpn
end
```

# L3/nC/nD Service Provider Configuration Example 2

```
class-map match-any real-time
  match ip dscp 46
  match qos-group <x>
class-map match-any bursty-low
  match ip dscp 18 20
  match qos-group <x>
class-map match-any bursty-high
  match ip dscp 26 28
  match qos-group <x>
policy-map child-dummy
policy-map port-shape
  class class-default
    shape average percent <%> <ms> ms
    service-policy child-dummy
policy-map content-vpn
  class real-time
    priority
    police cir percent <%> [bc <ms> ms] ! using bandwidth
  class bursty-high
    bandwidth percent <%>  ! using subinterface bandwidth
    random-detect [precedence-based | dscp-based | discard-class-based]
    random-detect {precedence | dscp | discard-class} <num> <min_th> ms <max_th> ms [mpd]
! using subinterface bandwidth
  class bursty-low
    bandwidth percent <%>  ! using subinterface bandwidth
    random-detect [precedence-based | dscp-based | discard-class-based]
    random-detect {precedence | dscp | discard-class} <num> <min_th> ms <max_th> ms [mpd]
! using subinterface bandwidth
  class class-default  ! best effort
    bandwidth percent <%>  ! using subinterface bandwidth
    shape average percent <%> <ms> ms ! using subinterface bandwidth
    random-detect [precedence-based | dscp-based | discard-class-based]
    random-detect {precedence | dscp | discard-class} <num> <min_th> ms <max_th> ms [mpd]
! using subinterface bandwidth
!
map-class frame-relay content-vpn
  service-policy output content-vpn-out
  service-policy input input-policer
map-class frame-relay internet
  service-policy output internet-out
map-class frame-relay nms-vpn
  service-policy output nms-vpn-out
interface POS3/0
  encapsulation frame-relay
  service-policy output port-shape
end
```

```
interface POS3/0.100 point-to-point
  frame-relay interface-dlci 100
  description e.g. Content VPN
  bandwidth <kbps>
  ip vrf forwarding content-vpn
  frame-relay class content-vpn
end
interface POS3/0.101 point-to-point
  frame-relay interface-dlci 101
  description e.g. Internet service
  bandwidth <kbps>
  frame-relay class internet
end
interface POS3/0.102 point-to-point
  frame-relay interface-dlci 102
  description e.g. NMS VPN
  bandwidth <kbps>
  ip vrf forwarding nms-vpn
  frame-relay class nms-vpn
end
```

# Additional References

The following sections provide references related to QoS: Enhancement to Subinterface QoS Policy Definitions (L3/nC/nD) Including Proportional Bandwidth Allocation.

## Related Documents

| Related Topic | Document Title |
|---|---|
| Modular QoS Command-Line Interface (CLI) (MQC) | *Cisco IOS Quality of Service Solutions Configuration Guide* |
| Information about attaching policy maps to interfaces | *Cisco IOS Quality of Service Solutions Configuration Guide* |
| Information about attaching policy maps to Frame Relay DLCIs | *Cisco IOS Wide-Area Networking Configuration Guide* |
| Additional match criteria that can be used for packet classification | *Cisco IOS Quality of Service Solutions Configuration Guide* |
| Frame Relay configuration information and information about DLCIs | *Cisco IOS Wide-Area Networking Configuration Guide* |
| Frame Relay commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples | *Cisco IOS Wide-Area Networking Command Reference, Release 12.3T* |

# Standards

| Standards | Title |
|---|---|
| No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature. | — |

# MIBs

| MIBs | MIBs Link |
|---|---|
| • CISCO-CLASS-BASED-QOS-MIB<br>• CISCO-CLASS-BASED-QOS-CAPABILITY-MIB | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

# RFCs

| RFCs | Title |
|---|---|
| No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature. | — |

# Technical Assistance

| Description | Link |
|---|---|
| Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content. | http://www.cisco.com/public/support/tac/home.shtml |