

IP to ATM CoS, per-VC WFQ and CBWFQ

This feature module describes the IP to ATM Class of Service, per-Virtual Circuit Weighted Fair Queueing and Class-Based Weighted Fair Queueing (IP to ATM CoS, per-VC WFQ and CBWFQ) feature. It contains the following sections:

- Feature Overview on page 1
- Configuration Tasks on page 4
- Configuration Tasks on page 4
- Feature Overview on page 1
- Command Reference on page 9

Feature Overview

The IP to ATM CoS, per-VC WFQ and CBWFQ feature allows you to apply CBWFQ functionality—normally applicable at the interface or subinterface levels only—to an individual VC configured for IP to ATM CoS. You can use this extension to the IP to ATM CoS feature to apply either class-based WFQ (CBWFQ) or flow-based WFQ on a per-VC basis.

Before you look at how IP to ATM CoS, per-VC WFQ and CBWFQ functions, consider the following summaries of CBWFQ and IP to ATM CoS.

About CBWFQ

CBWFQ extends the flow-based WFQ functionality to provide support for user-defined classes. CBWFQ allows you to define traffic classes that are based on certain match criteria such as access control lists, input interfaces names, protocols, and quality of service (QoS) labels. Once a class has been defined according to its match criteria, you can assign it characteristics. To characterize a class, you assign it bandwidth, weight, and maximum packet limit. The bandwidth assigned to a class is the minimum bandwidth delivered to the class during congestion. Also, to characterize a class, you specify the queue limit for that class, which is the maximum number of packets allowed to accumulate in its queue. Packets belonging to a class are subject to the bandwidth and queue limits that characterize the class.

After you define traffic classes, you can configure one or more of them in a policy map to be attached as a service policy. CBWFQ allows you to create policy maps and attach them to interfaces or subinterfaces as service policies. The IP to ATM CoS, per-VC WFQ and CBWFQ feature allows you to create a policy map using standard CBWFQ, then apply the map to a VC to be used as a service policy for that VC. For complete information on CBWFQ, refer to the Cisco IOS Release 12.0(5)T feature module titled *Class-Based Weighted Fair Queueing*.

About IP to ATM CoS

Before considering how per-VC WFQ and CBWFQ functions as part of IP to ATM CoS, it is helpful to have some understanding of the IP to ATM CoS feature itself, the main function of which is to provide for consistent QoS between IP and ATM interworked networks. IP to ATM CoS implements a solution for coarse-grained mapping of QoS characteristics between IP and ATM, using Cisco's PA-A3 ATM port adapters on Cisco 7200 series routers. (This category of coarse-grained QoS is often referred to as CoS.) The resulting feature makes it possible to support differential services in network service provider environments.

IP to ATM CoS is designed to provide a true working solution to class-based services, without the investment of new ATM network infrastructures. Now networks can offer different service classes (sometimes termed *differential service classes*) across the entire WAN, not just the routed portion. Mission-critical applications can be given exceptional service during periods of high network usage and congestion. In addition, noncritical traffic can be restricted in its network usage, which ensures greater QoS for more important traffic and user types. IP to ATM CoS supports configuration of both a single ATM VC and VC bundles. For complete information on IP to ATM CoS, refer to the 12.0(3)T feature module, *IP to ATM Class of Service*.

The IP to ATM CoS, per-VC WFQ and CBWFQ feature allows you to apply a policy map to a VC to specify a service policy for that VC so that all traffic sent on that VC is categorized according to the classes and their match criteria defined by the service policy. In other words, IP to ATM CoS, per-VC WFQ and CBWFQ takes the functionality defined for standard CBWFQ and makes it available for application and use at the discrete VC level.

IP to ATM CoS allows you to configure a single, standalone VC or individual VCs belonging to a bundle. You also can configure collectively all VCs belonging to a bundle. However, for per-VC WFQ and CBWFQ, you can configure individual VCs only. That is, you can configure a standalone VC or a VC that belongs to a bundle, but you cannot use per-VC WFQ and CBWFQ to configure a bundle of VCs collectively.

Per-VC WFQ and CBWFQ allows you to differentiate the use of individual VCs within a bundle. For instance, you can apply one service policy to one VC belonging to a VC bundle and apply a different service policy to another VC belonging to the same bundle. You can also apply the same policy map to multiple VCs—whether standalone or bundle members—but each VC can have only one service policy. To concatenate service policies, you must create a third policy map and include in it all the classes that you want to use from policy maps you would have concatenated.

The following is a summary of how you configure a VC to use CBWFQ:

- You define traffic classes to specify the classification policy (class maps).

This process determines how many types of packets are to be differentiated from one another. (The Cisco IOS Release 12.0(5)T feature module titled *Class-Based Weighted Fair Queueing* explains how to do this.)

Note For per-VC WFQ and CBWFQ, the total amount of bandwidth you can allocate for all classes included in a service policy to be attached to a VC cannot exceed 75 percent of the available bandwidth for that VC. The remaining 25 percent of available bandwidth is used for encapsulation, such as the ATM cell overhead (also referred to as ATM tax cut), and other functions that assume overhead.

- You configure policy maps containing classes that specify the policy for each traffic class.
- You attach a policy map to a VC that uses IP to ATM CoS to specify the service policy for the VC. (This feature module explains how to do so.)

To apply flow-based WFQ on a per-VC basis, you configure WFQ in the predefined CBWFQ default class, which is called class-default, but you do not ascribe bandwidth to the default class. How to configure the default class to specify flow-based fair queueing is explained in “Configuring a VC to Use Flow-Based WFQ” on page 6.

Benefits

Bandwidth Allocation

CBWFQ allows you to specify the exact amount of bandwidth to be allocated for a specific class of traffic. As long as the total bandwidth for all classes included in a service policy does not exceed 75 percent of the available bandwidth for the VC, you can configure up to 64 classes and control bandwidth distribution among them.

Coarser Granularity and Scalability

CBWFQ allows you to define what constitutes a class based on criteria that exceed the confines of flow. CBWFQ allows you to use access control lists and input interface names or protocols to define how traffic will be classified, thereby providing coarser granularity. You need not maintain traffic classification on a flow basis. The ability to apply a service policy specified by various classes at the VC level gives you greater control and differentiation over how traffic sent over the VC is classified and treated, distinguishing it from how traffic sent over other VCs is classified and treated.

Related Features and Technologies

This feature is an extension to the IP to ATM CoS feature. This feature includes use of the standard CBWFQ implementation. To use per-VC WFQ and CBWFQ, you must use the commands documented in this feature module and the standard CBWFQ commands documented in the feature module titled 12.0(5)T *Class-Based Weighted Fair Queueing* feature module; you use these commands to create classes and define policy maps to be attached to individual VCs as service policies.

Related Documents

For related information on this feature, refer to the following documents:

- Cisco IOS Release 12.0(3)T *IP to ATM Class of Service* feature module
- Cisco IOS Release 12.0(5)T *Class-Based Weighted Fair Queueing* feature module
- Cisco IOS Release 12.0 *Quality of Service Solutions Configuration Guide*
- Cisco IOS Release 12.0 *Quality of Service Solutions Command Reference*

CBWFQ supports standard and extended numbered access lists only. For information on creating access lists, see the appropriate Cisco IOS Release 12.0 configuration guides and command references.

Supported Platforms

The IP to ATM CoS, per-VC WFQ and CBWFQ feature is supported on any Cisco 7200 series routers equipped with the following hardware:

- NPE-200 or higher
- One of the following PA-A3 port adapters: DS3 or OC-3

Supported Standards, MIBs and RFCs

None

Prerequisites

The IP to ATM CoS feature requires ATM PVC management and Cisco Express Forwarding (CEF) switching functionality. It also requires that the remote router run a version of Cisco IOS software that supports IP to ATM CoS with VC bundle management.

To use the feature described in this feature module, you should be familiar with the following QoS features:

- ATM Forum Traffic Management
- CAR
- PBR
- WRED (and DWRED)

Documentation for these features can be found on the Documentation CD-ROM and on Cisco Connection Online (CCO).

To use this feature, you should also be able to install a port adapter in a Cisco 7200 series router and to configure the router.

Access Control Lists

Because you can specify a numbered access list as the match criterion for any class that you create, you should know how to configure access lists.

Configuration Tasks

See the following sections for configuration tasks for the IP to ATM CoS, per-VC WFQ and CBWFQ feature. The first two sections are required; the remaining sections are optional.

- Configuring Class-Based Weighted Fair Queueing (Required)
- Attaching a Service Policy and Enabling CBWFQ for a VC (Required)
- Configuring a VC to Use Flow-Based WFQ (Optional)
- Monitoring Per-VC WFQ and CBWFQ (Optional)
- Enabling Logging of Error Messages to the Console (Optional)

Configuring Class-Based Weighted Fair Queueing

Before configuring CBWFQ for a VC, you must perform the following tasks using standard CBWFQ commands:

- Create one or more classes to be used to classify traffic sent across the VC
- Define a policy map containing the classes to be used as the service policy

Note You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes included in a policy map to be attached to a VC must not exceed 75 percent of the available bandwidth of the VC. The remaining 25 percent of available bandwidth is used for encapsulation, such as the ATM cell overhead (also referred to as ATM cell tax), routing and best-effort traffic, and other functions that assume overhead. See the 12.0(5)T feature module, *Class-Based Weighted Fair Queueing* for directions on how to perform these tasks and for the syntax of the commands that you must use.

Attaching a Service Policy and Enabling CBWFQ for a VC

Because CBWFQ gives you minimum bandwidth guarantee, you can only apply CBWFQ to VCs having these classes of service: available bit rate (ABR) and variable bit rate (VBR). You cannot apply per-VC WFQ and CBWFQ to UBR and UBR+ VCs because both of these service classes are best-effort classes that do not guarantee minimum bandwidth. When CBWFQ is enabled for a VC, all classes configured as part of the service policy are installed in the fair queueing system.

To attach a policy map to a standalone VC to be used as its service policy and to enable CBWFQ on that VC, use the following command in VC submode:

Command	Purpose
Router(config-if-atm-vc)# service-policy output <i>policy-map</i>	Enables CBWFQ and attaches the specified service policy map to the VC being created or modified.

To attach a policy map to an individual VC bundle member to be used as its service policy and to enable CBWFQ on that VC, use the following command in bundle-vc configuration mode:

Command	Purpose
Router(config-if-atm-member)# service-policy output <i>policy-map</i>	Enables CBWFQ and attaches the specified service policy map to the VC being created or modified.

Note The **service-policy output** and **random-detect-group** commands are mutually exclusive; you cannot apply a WRED group to a VC for which you have enabled CBWFQ through application of a service policy. Moreover, before you can configure one command, you must disable the other if it is configured.

Configuring a VC to Use Flow-Based WFQ

In addition to configuring CBWFQ at the VC level, the IP to ATM CoS feature allows you to configure flow-based WFQ at the VC level. Because flow-based WFQ gives you best-effort class of service—that is, it does not guarantee minimum bandwidth—you can configure per-VC WFQ for all types of CoS VC: ABR, VBR, UBR, and UBR+.

Per-VC WFQ uses the class-default class. Therefore, to configure per-VC WFQ, you must first create a policy map and configure the class-default class. (You need not create the class-default class, which is predefined, but you must configure it.) For per-VC WFQ, the class-default class must be configured with the **fair-queue** policy-map configuration command.

In addition to configuring the **fair-queue** policy-map class configuration command, you can configure the default class with either the **queue-limit** command or the **random-detect** command, but not both. Moreover, if you want the default class to use flow-based WFQ, you cannot configure the default class with the **bandwidth** policy-map class configuration command—to do so would disqualify the default class as flow-based WFQ, and therefore limit application of the service policy containing the class to ABR and VBR VCs.

To create a policy map and configure the class-default class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-map</i>	Specifies the name of the policy map to be created or modified.
Step 2	Router(config-pmap)# class class-default <i>default-class-name</i>	Specifies the default class so that you can configure or modify its policy.
Step 3	Router(config-pmap-c)# fair-queue <i>number-of-dynamic-queues</i>	Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class.
Step 4	Router(config-pmap-c)# queue-limit <i>number-of-packets</i> or Router(config-pmap-c)# random-detect	Specifies the maximum number of packets that can be enqueued for the class. Enables WRED. The class policy will drop packets using WRED instead of tail drop.

For more information about creating a policy map and configuring the class-default class, see the 12.0(5)T feature module, *Class-Based Weighted Fair Queueing*.

By default—that is, even if you do not configure the class-default class with the **fair-queue** policy-map class configuration command and you do not configure it with the **bandwidth** policy-map class configuration command—the default class is defined as flow-based WFQ.

Note that you can include other classes in the same policy map as the one that contains the flow-based WFQ class. Packets not otherwise matched are selected by the default class-default class match criteria.

To attach the policy map containing the class-default class to a standalone VC so that it becomes the service policy enabling WFQ for that VC, use the following command in VC submode:

Command	Purpose
Router(config-if-atm-vc)# service-policy output <i>policy-map</i>	Enables WFQ for the VC by attaching the specified policy map containing the class-default class to the VC being created or modified.

To attach the policy map containing the class-default class to an individual VC bundle member so that the policy map becomes the service policy enabling WFQ for that VC, use the following command in bundle-vc configuration mode:

Command	Purpose
Router (config-if-atm-membe) # service-policy output <i>policy-map</i>	Enables WFQ for the VC bundle member by attaching the specified policy map containing the class-default class to the VC bundle member.

Monitoring Per-VC WFQ and CBWFQ

To monitor per-VC WFQ and CBWFQ in your network, use one or more of the following commands in EXEC mode:

Command	Purpose
Router# show queue <i>interface-name interface-number [vc [vpi/] vci]</i>	Displays the contents of packets inside a queue for a particular interface or VC.
Router# show queueing interface <i>interface-number [vc [[vpi/] vci]]</i>	Displays the queueing statistics of a specific VC on an interface.

Enabling Logging of Error Messages to the Console

When you configure a VC in order to create or modify it, the router performs the task in interrupt mode. For this reason, the router cannot issue printf statements to inform you of error conditions, if errors occur. Rather, the router logs all error messages to the console. To accommodate these circumstances, you should enable logging of error messages to the console.

To enable logging of error messages to the console, use the following command in global configuration mode:

Command	Purpose
Router (config) # logging console <i>level</i>	Limits messages logged to the console based on severity.

For information on the **logging console** command, including configuration tasks and command syntax, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and the *Cisco IOS Configuration Fundamentals Command Reference*.

Configuration Examples

This section provides the following configuration examples:

- Per-VC WFQ and CBWFQ on a Standalone VC Example
- Per VC WFQ and CBWFQ on Bundle-Member VCs Example

Per-VC WFQ and CBWFQ on a Standalone VC Example

The following example creates two class maps and defines their match criteria. For the first map class, called class1, the numbered access control list (ACL) 101 is used as the match criterion. For the second map class, called class2, the numbered ACL 102 is used as the match criterion.

Next, the example includes these classes in a policy map called policy1. For class1, the policy includes a minimum bandwidth allocation request of 500 Mbps and maximum packet count limit of 30 for the queue reserved for the class. For class2, the policy specifies only the minimum bandwidth allocation request of 1000 Mbps, so the default queue limit of 64 packets is assumed. Note that the sum of the bandwidth requests for the two classes comprising policy1 is 75 percent of the total amount of bandwidth (2000 Mbps) for PVC cisco to which the policy map is attached.

The example attaches the policy map called policy1 to the PVC called cisco. Once the policy map policy1 is attached to PVC cisco, its classes constitute the CBWFQ service policy for that PVC. Packets sent on this PVC will be checked for matching criteria against ACLs 101 and 102 and classified accordingly.

Because **class-default** is not explicitly configured for this policy map, all traffic that does not meet the match criteria of the two classes comprising the service policy is handled by the predefined class-default class, which provides best-effort flow-based WFQ.

```
class-map class1
  match access-group 101

class-map class2
  match access-group 102

policy-map policy1
  class class1
    bandwidth 500
    queue-limit 30

  class class2
    bandwidth 1000

interface ATM1/1/0.46 multipoint
  ip address 200.126.186.2 255.255.255.0
  pvc cisco 46
  vbr-nrt 2000 2000
  encaps aal5snap
  service policy output policy1
```

Per VC WFQ and CBWFQ on Bundle-Member VCs Example

The following example shows a PVC bundle called san-francisco with members for which per-VC WFQ and CBWFQ are enabled and service policies configured. The example assumes that the classes included in the following policy maps have been defined and that the policy maps have been created: policy1, policy2, and policy4. For each PVC, the IP to ATM CoS **pvc-bundle** command is used to specify the PVC to which the specified policy map is to be attached.

Note that PVC 0/34 and 0/31 have the same policy map attached to them, policy2. Although you can assign the same policy map to multiple VCs, each VC can have only one policy map attached at an output PVC.

```
bundle san-francisco
  protocol ip 1.0.2.20 broadcast
  encapsulation aal5snap
  pvc-bundle 0/35
    service policy output policy1
    vrt-nrt 5000 3000 500
    precedence 4-7
  pvc-bundle 0/34
    service policy output policy2
    vrt-nrt 5000 3000 500
    precedence 2-3
  pvc-bundle 0/33
    vrt-nrt 4000 3000 500
    precedence 2-3
  service policy output policy4
  pvc-bundle 0/31
    service policy output policy2
```

Command Reference

This section documents commands modified to support this feature. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publications and the Cisco IOS Release 12.0(5)T feature modules.

- **pvc**
- **pvc-bundle**
- **service-policy**
- **show policy interface**
- **show queue**
- **show queueing interface**

pvc

Use the **pvc** interface configuration command to perform one or more of the following tasks:

- Create an ATM permanent virtual circuit (PVC) on a main interface or subinterface
- Assign a name to an ATM PVC
- Specify ILMI, QSAAL, or SMDS as the encapsulation type on an ATM PVC (To configure other encapsulations types, see the **encapsulation** command).
- Enter interface-ATM-VC configuration mode

To remove an ATM PVC, use the **no** form of this command.

```
pvc [name] vpi/vci [ilmi | qsaal | smds]
no pvc [name] vpi/vci [ilmi | qsaal | smds]
```

Syntax Description

<i>name</i>	(Optional) The name of the PVC or map. The name can be up to 16 characters long.
<i>vpi/vci</i>	<p>ATM network virtual path identifier (VPI) for this PVC. The absence of the “/” and a <i>vpi</i> value defaults the <i>vpi</i> value to 0. On the Cisco 7200 and 7500 series routers, this value ranges from 0 to 255; on the Cisco 4500 and 4700 routers, this value ranges from 0 to 1 less than the quotient of 8192 divided by the value set by the atm vc-per-vc command.</p> <p>The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0.</p>
<i>vci</i>	<p>ATM network virtual channel identifier (VCI) for this PVC. This value ranges from 0 to 1 less than the maximum value set for this interface by the atm vc-per-vc command. Typically, lower values 0 to 31 are reserved for specific traffic (for example, F4 OAM, SVC signalling, ILMI, and so on) and should not be used.</p> <p>The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has local significance only. The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0.</p>
ilmi	(Optional) Used to set up communication with the ILMI; the associated <i>vpi</i> and <i>vci</i> values ordinarily are 0 and 16, respectively.
qsaal	(Optional) A signalling-type PVC used for setting up or tearing down SVCs; the associated <i>vpi</i> and <i>vci</i> values ordinarily are 0 and 5, respectively.
smds	(Optional) Encapsulation for SMDS networks. If you are configuring an ATM PVC on the ATM Interface Processor (AIP), you must configure AAL3/4SMDS using the atm aal aal3/4 command before specifying smds encapsulation. If you are configuring an ATM network processor module (NPM), the atm aal aal3/4 command is not required. SMDS encapsulation is not supported on the ATM port adapter.

Defaults

No PVC is defined. When a PVC is defined, the global default of the **encapsulation** command applies (*aal-encap* = **aal5snap**).

Command Modes

Interface or subinterface configuration

Command History

Release	Modification
11.3T	This command was introduced.

Usage Guidelines

The **pvc** command replaces the **atm pvc** command, which, although still supported and available, will be obsoleted in the near future. Use the **pvc** command to configure a single ATM VC only, not a VC that is a bundle member. We recommend that you use the **pvc** command in conjunction with the **encapsulation** and **random-detect attach** commands instead of the **atm pvc** command.

The Cisco IOS software dynamically creates rate queues as necessary to satisfy the requests of the **pvc** commands.

The **pvc** command creates a PVC and attaches it to the VPI and VCI specified. Both *vpi* and *vci* cannot be simultaneously specified as 0; if one is 0, the other cannot be 0.

When configuring an SVC, use the **pvc** command to configure the PVC that handles SVC call setup and termination. In this case, specify the **qsaal** keyword. See the second example that follows.

Once you specify a *name* for a PVC, you can reenter the interface-ATM-VC configuration mode by simply entering **pvc name**. You can remove a PVC and any associated parameters by entering **no pvc name** or **no pvc vpi/vci**.

Note After configuring the parameters for an ATM PVC, you must exit the interface-ATM-VC configuration mode in order to create the PVC and enable the settings.

If **ilmi**, **qsaal**, or **smds** encapsulation is not explicitly configured on the ATM PVC, the PVC inherits the following default configuration (listed in order of next highest precedence):

- Configuration of the **encapsulation** command in a VC class assigned to the PVC itself.
- Configuration of the **encapsulation** command in a VC class assigned to the ATM subinterface of the PVC.
- Configuration of the **encapsulation** command in a VC class assigned to the ATM main interface of the PVC.
- Global default: The global default of the **encapsulation** command applies: *aal-encap* = **aal5snap**.

Examples

The following example creates a PVC with VPI 0 and VCI 16, and communication is set up with the ILMI:

```
pvc cisco 0/16 ilmi
exit
```

The following example creates a PVC used for ATM signalling for an SVC. It specifies VPI 0 and VCI 5:

```
pvc cisco 0/5 qsaal
exit
```

The following example configures the PVC called cisco to use CBWFQ. It attaches a policy map called policy1 to the PVC. The classes comprising policy1 determine the service policy for the PVC:

```
pvc cisco 0/5
  service-policy output policy1
vbr-nrt 2000 2000
encap aal5snap
```

Related Commands

Command	Description
atm vc-per-vp	Sets the maximum number of VCIs to support per VPI.

pvc-bundle

To add a virtual circuit (VC) to a bundle as a member of the bundle and enter bundle-vc configuration mode in order to configure that VC bundle member, use the **pvc-bundle** bundle configuration command. The **no** form of this command removes the VC from the bundle.

```
pvc-bundle pvc-name [vpi/] [vci]
no pvc-bundle pvc-name [vpi/] [vci]
```

Syntax Description

<i>pvc-name</i>	The name of the PVC.
vpi/	<p>(Optional) ATM network virtual path identifier (VPI) for this permanent virtual circuit (PVC). The absence of the “/” and a <i>vpi</i> value defaults the <i>vpi</i> value to 0.</p> <p>On the Cisco 7200 and 7500 series routers, this value ranges from 0 to 255; on the Cisco 4500 and 4700 routers, this value ranges from 0 to 1 less than the quotient of 8192 divided by the value set by the atm vc-per-vc command.</p> <p>The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0.</p>
vci	<p>(Optional) ATM network virtual channel identifier (VCI) for this PVC. This value ranges from 0 to 1 less than the maximum value set for this interface by the atm vc-per-vc command. Typically, lower values 0 to 31 are reserved for specific traffic (for example, F4 OAM, SVC signalling, ILMI, and so on) and should not be used.</p> <p>The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has local significance only.</p> <p>The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0.</p>

Defaults

None

Command Modes

Bundle configuration

Command History

Release	Modification
12.0(3)T	This command was introduced.

Usage Guidelines

Each bundle can contain multiple VCs having different quality of service (QoS) attributes. This command associates a VC with a bundle, making it a member of that bundle. Before you can add a VC to a bundle, the bundle must exist. Use the **bundle** command to create a bundle. You can also use this command to configure a VC that already belongs to a bundle. You enter the command in the same way, giving the name of the VC bundle member.

The **pvc-bundle** command enters into bundle-vc configuration mode in which you can specify VC-specific and VC class attributes for the VC.

Examples

The following example specifies an existing bundle named *chicago*, and enters into bundle configuration mode. Then, it adds two VCs to the bundle. For each added VC, bundle-vc mode is entered and a VC class is attached to the VC to configure it.

```
bundle chicago
pvc-bundle chicago-control 207
class control-class
pvc-bundle chicago-premium 206
class premium-class
```

The following example configures the PVC called *chicago-control*, an existing member of the bundle called *chicago*, to use Class-Based Weighted Fair Queueing (CBWFQ). The example configuration attaches the policy map *policy1* to the PVC. Once the policy map is attached, the classes comprising *policy1* determine the service policy for the PVC *chicago-control*.

```
bundle chicago
pvc-bundle chicago-control 207
class control-class
service-policy output policy1
```

Related Commands

Command	Description
atm vc-per-vp	Sets the maximum number of VCI to support per VPI.
bump	Configures bumping rules for a virtual circuit class that can be assigned to a virtual circuit bundle.
class-bundle	Configures a VC bundle with the bundle-level commands contained in the specified VC class.
class	Assigns a VC class to an ATM PVC, SVC, or VC bundle member.
precedence (VC bundle)	Configures precedence levels for a VC
protect	Configures a VC class with protected group or protected VC status for application to a VC bundle.

ubr	Configures unspecified bit rate (UBR) QoS and specifies the output peak cell rate for an ATM PVC, SVC, VC class, or VC bundle member.
ubr+	Configures UBR QoS and specifies the output peak cell rate and output minimum guaranteed cell rate for an ATM PVC, SVC, VC class, or VC bundle member.
vbr-nrt	Configures the variable bit rate nonreal time (VBR-NRT) QoS and specifies output peak cell rate, output sustainable cell rate, and output maximum burst cell size for an ATM PVC, SVC, VC class, or VC bundle member.

service-policy

To attach a policy map to an input interface or virtual circuit (VC) or an output interface or VC to be used as the service policy for that interface or VC, use the **service-policy** global configuration command. To remove a service policy from an input or output interface or input or output VC, use the **no** form of this command.

```
service-policy {input | output} policy-map
no service-policy {input | output}
```

Syntax Description

input	Attaches the specified policy map to the input interface or input VC.
output	Attaches the specified policy map to the output interface or output VC.
<i>policy-map</i>	The name of a service policy map (created using the policy-map command) to be attached.

Defaults

No service policy is specified.

Command Modes

- Global configuration
- VC submode (for a standalone VC)
- Bundle-vc configuration (for ATM VC bundle members)

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

You can attach a single policy map to one or more interfaces or one or more VCs to specify the service policy for those interfaces or VCs.

Currently a service policy specifies Class-Based Weighted Fair Queueing (CBWFQ). The class policies comprising the policy map are then applied to packets that satisfy the class map match criteria for the class.

To successfully attach a policy map to an interface or a VC, the aggregate of the configured minimum bandwidths of the classes comprising the policy map must be less than or equal to 75 percent of the interface bandwidth or the bandwidth allocated to the VC.

Attaching a service policy and enabling CBWFQ on an interface renders ineffective any commands related to fancy queueing such as commands pertaining to fair queueing, custom queueing, priority queueing, and Weighted Random Early Detection (WRED). You can configure these features only after you remove the policy map from the interface.

You can modify a policy map attached to an interface or a VC, changing the bandwidth of any of the classes comprising the map. Bandwidth changes that you make to an attached policy map are effective only if the aggregate of the bandwidth amounts for all classes comprising the policy map, including the modified class bandwidth, is less than or equal to 75 percent of the interface bandwidth or the VC bandwidth. If the new aggregate bandwidth amount exceeds 75 percent of the interface bandwidth or VC bandwidth, the policy map is not modified.

Examples

The following example attaches the service policy map called policy9 to the input interface Serial1:

```
interface Serial1
  service-policy input policy9
```

The following example attaches the service policy map called policy9 to the input permanent virtual circuit (PVC) called cisco:

```
pvc cisco 0/34
  service-policy input policy9
vbr-nt 5000 3000 500
precedence 4-7
```

The following example attaches the policy called policy9 to the output interface serial1 to specify the service policy for the interface and enable CBWFQ on it:

```
interface serial1
  service-policy output policy9
```

The following example attaches the service policy map called policy9 to the output PVC called cisco:

```
pvc cisco 0/5
  service-policy output policy9
vbr-nt 4000 2000 500
precedence 2-3
```

Related Commands

Command	Description
policy-map	Specifies a policy map to be assigned to an interface.
show policy	Displays configuration of all classes for a policy map.

show policy interface

To display the configuration of all classes configured for all service policies on the specified interface or to display the classes for the service policy for a specific permanent virtual circuit (PVC) on the interface, use the **show policy interface** global configuration command.

```
show policy interface interface-name [vc [vpi] vci ] ]
```

Syntax Description

<i>interface-name</i>	Name of the interface or subinterface whose policy configuration is to be displayed.
vc	(Optional) For ATM interfaces only, shows the policy configuration for a specified PVC. The name can be up to 16 characters long.
<i>vpi</i>	(Optional) ATM network virtual path identifier (VPI) for this PVC. The absence of the “/” and a <i>vpi</i> value defaults the <i>vpi</i> value to 0. On the Cisco 7200 and 7500 series routers, this value ranges from 0 to 255. The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0. If this value is omitted, information for all VCs on the specified ATM interface or subinterface is displayed.
<i>vci</i>	(Optional) ATM network virtual channel identifier (VCI) for this PVC. This value ranges from 0 to 1 less than the maximum value set for this interface by the atm vc-per-vp command. Typically, lower values 0 to 31 are reserved for specific traffic (for example, F4 OAM, SVC signalling, ILMI, and so on) and should not be used. The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has local significance only. The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0.

Defaults

There is no default behavior.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The **show policy interface** command displays the configuration for classes on the specified interface or the specified PVC only if a service policy has been attached to the interface or the PVC.

You can use the *pvc-name* argument to display output for a PVC only for PA-A3 ATM port adapters that support per-VC queueing.

Examples

The following example displays configurations for classes on the output interface e1/1:

```
Router# show policy interface output e1/1
Ethernet1/1 output : pol
  Weighted Fair Queueing
    Class class1
      Output Queue: Conversation 264
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11548/0/0
    Class class2
      Output Queue: Conversation 265
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11546/0/0
    Class class3
      Output Queue: Conversation 266
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11546/0/0
    Class class4
      Output Queue: Conversation 267
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11702/0/0
    Class class5
      Output Queue: Conversation 268
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11701/0/0
    Class class6
      Output Queue: Conversation 269
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11702/0/0
    Class class7
      Output Queue: Conversation 270
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11857/0/0
    Class class8
      Output Queue: Conversation 271
      Bandwidth 937 (kbps) Max Threshold 64 (packets)
      (total/discards/tail drops) 11858/1/0
```

show policy interface

The following example displays configurations for classes comprising the service policy for the output VC 0/101 on the output interface atm2/0.6:

```
qos4-72a#show policy interface atm2/0.6
ATM2/0.6: VC 0/101 - output : p1
  Weighted Fair Queueing
    Class c-vc1-c1
      Output Queue: Conversation 264
      Bandwidth 31 (kbps)
      mean queue depth: 1
      drops: class random tail min-th max-th mark-prob
            0      0      0    100   200   1/10
            1      0      0    105   200   1/10
            2      0      0    110   200   1/10
            3      0      0    115   200   1/10
            4      0      0    120   200   1/10
            5      0      0    125   200   1/10
            6      0      0    130   200   1/10
            7      0      0    135   200   1/10
            rsvp  0      0    140   200   1/10
    Class c-vc1-c2
      Output Queue: Conversation 265
      Bandwidth 54 (kbps)
      mean queue depth: 1
      drops: class random tail min-th max-th mark-prob
            0      0      0     60   100   1/10
            1      0      0     65   100   1/10
            2      0      0     70   100   1/10
            3      0      0     75   100   1/10
            4      0      0     80   100   1/10
            5      0      0     83   100   1/10
            6      0      0     85   100   1/10
            7      0      0     87   100   1/10
            rsvp  0      0     90   100   1/10
    Class c-vc1-c3
      Output Queue: Conversation 266
      Bandwidth 77 (kbps)
      mean queue depth: 0
      drops: class random tail min-th max-th mark-prob
            0      0      0      1    10   1/10
            1      0      0      2    10   1/10
            2      0      0      3    10   1/10
            3      0      0      4    10   1/10
            4      0      0      5    10   1/10
            5      0      0      6    10   1/10
            6      0      0      7    10   1/10
            7      0      0      7    10   1/10
            rsvp  0      0      7    10   1/10
    Class c-vc1-c4
      Output Queue: Conversation 267
      Bandwidth 100 (kbps)
      mean queue depth: 9
      drops: class random tail min-th max-th mark-prob
            0      0      0      1    10   1/10
            1      9      220    2    10   1/10
            2     24     645    3    10   1/10
            3     22     844    4    10   1/10
            4      0      0      5    10   1/10
            5     23     351    6    10   1/10
            6     28     213    7    10   1/10
            7     59     540    7    10   1/10
            rsvp  0      0      7    10   1/10
```

```

Class c-vc1-c5
  Output Queue: Conversation 268
    Bandwidth 123 (kbps)
  mean queue depth: 150
  drops: class  random  tail    min-th  max-th  mark-prob
         0      120    1777   50     150    1/50
         1      136    1549   60     150    1/50
         2       88    2354   70     150    1/50
         3      121    1569   80     150    1/50
         4      122    1717   80     150    1/50
         5       0      0      90     150    1/50
         6       0      0     100    150    1/50
         7      105    2058  110     150    1/50
         rsvp    0      0     120    150    1/50

Class c-vc1-c6
  Output Queue: Conversation 269
    Bandwidth 146 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 50216/32696/0
Class c-vc1-c7
  Output Queue: Conversation 270
    Bandwidth 216 (kbps) Max Threshold 64 (packets)
  (total/discards/tail drops) 74577/51994/0
Class class-default
  Flow Based Fair Queueing
  Number of Hashed Queues 256
  drops: class  random  tail    min-th  max-th  mark-prob
         0      101    828   50     150    1/50
         1       87    1154  60     150    1/50
         2      115    476   70     150    1/50
         3      116    444   80     150    1/50
         4      123    338   80     150    1/50
         5       92    1042  90     150    1/50
         6       79    1068  100    150    1/50
         7      110    740   110    150    1/50
         rsvp    0      0     120    150    1/50

```

Related Commands

Command	Description
show policy	Displays the configuration for all classes of the specified service policy map.
show policy class	Displays the configuration for the specified class of the specified service policy map.

show queue

To list fair queueing configuration and statistics for a particular interface or a specific virtual circuit (VC) on an ATM interface, use the **show queue** privileged EXEC command.

```
show queue interface-name interface-number [vc [vpi/] vci ] ]
```

Syntax Description

<i>interface-name</i>	The name of the interface.
<i>interface-number</i>	The number of the interface.
vc	(Optional) For ATM interfaces only, shows the fair queueing configuration for a specified permanent virtual circuit (PVC). The name can be up to 16 characters long.
<i>vpi/</i>	(Optional) ATM network virtual path identifier (VPI) for this PVC. The absence of the “/” and a <i>vpi</i> value defaults the <i>vpi</i> value to 0. On the Cisco 7200 and 7500 series routers, this value ranges from 0 to 255. The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0. If this value is omitted, information for all VCs on the specified ATM interface or subinterface is displayed.
<i>vci</i>	(Optional) ATM network virtual channel identifier (VCI) for this PVC. This value ranges from 0 to 1 less than the maximum value set for this interface by the atm vc-per-vc command. Typically, lower values 0 to 31 are reserved for specific traffic (for example, F4 OAM, SVC signaling, ILMI, and so on) and should not be used. The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has local significance only. The arguments <i>vpi</i> and <i>vci</i> cannot both be set to 0; if one is 0, the other cannot be 0.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.1(22)CC	This command was introduced.

Usage Guidelines

This command displays statistics for interfaces or VCs configured with the fair queueing strategy.

You can use the **vc** keyword and its arguments to display output for a PVC only for PA-A3 ATM port adapters that support per-VC queueing.

Examples

The following is sample output from the **show queue** command. Two conversations are active on the serial 1 interface. Weighted fair queueing ensures that both of these IP data streams, one TCP and the other UDP, receive equal bandwidth on the interface while they have messages in the pipeline.

```
Router# show queue serial1

Input queue: 0/75/0 (size/max/drops); Total output drops: 303628
Queueing strategy: weighted fair
Output queue: 64/1000/64/303628 (size/max total/threshold/drops)
  Conversations 2/2/256 (active/max active/max total)
  Reserved Conversations 0/0 (allocated/max allocated)

(depth/weight/discards/tail drops/interleaves) 45/4096/1123/0/0
Conversation 244, linktype: ip, length: 50
source: 55.1.1.1, destination: 66.1.1.2, id: 0x0000, ttl: 59,
TOS: 0 prot: 6, source port 55, destination port 55

(depth/weight/discards/tail drops/interleaves) 19/4096/302541/0/0
Conversation 185, linktype: ip, length: 118
source: 55.1.1.1, destination: 66.1.1.2, id: 0x0000, ttl: 59,
TOS: 0 prot: 17, source port 20, destination port 20
```

The following is sample output from the **show queue** command for PVC 33 on the atm2/0.33 ATM subinterface. Two conversations are active on the this interface. WFQ ensures that both data streams receive equal bandwidth on the interface while they have messages in the pipeline.

```
Router# show queue atm2/0.33 vc 33

Interface ATM2/0.33 VC 0/33
Queueing strategy: weighted fair
Total output drops per VC: 18149
Output queue: 57/512/64/18149 (size/max total/threshold/drops)
  Conversations 2/2/256 (active/max active/max total)
  Reserved Conversations 3/3 (allocated/max allocated)

(depth/weight/discards/tail drops/interleaves) 29/4096/7908/0/0
Conversation 264, linktype: ip, length: 254
source: 15.1.1.1, destination: 1.0.2.20, id: 0x0000, ttl: 59,
TOS: 0 prot: 17, source port 1, destination port 1

(depth/weight/discards/tail drops/interleaves) 28/4096/10369/0/0
Conversation 265, linktype: ip, length: 254
source: 15.1.1.1, destination: 1.0.2.20, id: 0x0000, ttl: 59,
TOS: 32 prot: 17, source port 1, destination port 2
```

Table 2 describes the fields shown in this display.

Table 2 show queue Field Descriptions

Field	Description
Input Queue	Input queue size in packets.
Total output drops per VC	Total output packet drops.
Queueing strategy	Type of queueing active on this interface.
Output queue	Output queue size in packets.
Conversations	WFQ conversation number.
Reserved Conversations	Total number of reserved WFQ conversations. Default is 256.

Table 2 show queue Field Descriptions (continued)

Field	Description
depth	Queue depth for the conversation in packets.
weight	Weight used in WFQ.
discards	Number of packet discards for the conversation.
tail drops	Number of tail drop packets for the conversation.
interleaves	Number of packets interleaved.
linktype	Protocol name.
length	Packet length.
source	Source IP address.
destination	Destination IP address.
id	Packet ID.
ttl	Time to live count.
TOS	IP type of service.
prot	Layer 4 protocol number.

show queueing interface

To show the queueing statistics of an interface or a virtual circuit (VC), use the **show queueing interface** privileged EXEC command.

```
show queueing interface interface-number [vc [[vpi/] vci]]
```

Syntax Description

<i>interface-number</i>	Specifies the interface.
vc	(Optional) Shows the weighted fair queueing (WFQ) and Weighted Random Early Detection (WRED) parameters associated with a specific VC. If desired, both the virtual path identifier (VPI) and virtual channel identifier (VCI) values, or just the VCI value, can be specified.
<i>vpi</i>	(Optional) Specifies the VPI. If <i>vpi</i> is omitted, 0 is used as the VPI value for locating the permanent virtual circuit (PVC). If <i>vpi</i> is specified, the / separator is required.
<i>vci</i>	(Optional) Specifies the VCI.

Command Mode

Privileged EXEC

Command History

Release	Modification
11.1(22)CC	This command was introduced.

Usage Guidelines

This command first appeared in Cisco IOS Release 11.1(22)CC.

Examples

The following is sample output from the **show queueing interface** command. It shows the WFQ configuration for PVC 0/101 on the ATM subinterface ATM2/0.6. Eight conversations are active on the interface. WFQ ensures that all of these data stream receive equal bandwidth on the interface while they have messages in the pipeline.

```
Router# show queueing interface atm2/0 serial1

Interface ATM2/0.6 VC 0/101
Queueing strategy: weighted fair
Total output drops per VC: 111883
Output queue: 311/512/64/111883 (size/max total/threshold/drops)
  Conversations  8/9/256 (active/max active/max total)
  Reserved Conversations 7/7 (allocated/max allocated)
```

Related Commands

Command	Description
exponential-weighting-constant	Configures the exponential weight factor for the average queue size calculation for a WRED parameter group.
precedence (WRED group)	Configures a WRED group for a particular IP precedence.
random-detect-group	Defines the WRED or Distributed WRED parameter group.
show queueing red	Shows the configured WRED parameters.