Hewlett Packard
Enterprise

HPE FlexNetwork 6600/HSR6600 Routers

ACL and QoS Configuration Guide

Part number: 5998-1491R
Software version: A6600_HSR6602-CMW520-R3303P25
Document version: 6W105-20151231

Acknowledgments

Contents

Configuring ACLs

Overview

An access control list (ACL) is a set of rules (or permit or deny statements) for identifying traffic based on criteria such as source IP address, destination IP address, and port number.

ACLs are primarily used for traffic identification. The packet drop or forwarding decisions varies with the modules that use ACLs.

ACL categories

Category	ACL number	IP version	Match criteria
Basic ACLs	2000 to 2999	IPv4	Source IPv4 address
		IPv6	Source IPv6 address
Advanced ACLs	3000 to 3999	IPv4	Source IPv4 address, destination IPv4 address, packet priority, protocol number, and other Layer 3 and Layer 4 header fields
		IPv6	Source IPv6 address, destination IPv6 address, packet priority, protocol number, and other Layer 3 and Layer 4 header fields
Ethernet frame header ACLs	4000 to 4999	N/A	Layer 2 header fields, such as source and destination MAC addresses, 802.1p priority, and link layer protocol type

Numbering and naming ACLs

Each ACL category has a unique range of ACL numbers. You can assign the ACL a name for ease of identification. After creating an ACL with a name, you cannot rename it or delete its name.

For an IPv4 basic or advanced ACLs, its ACL number and name must be unique in IPv4, and for an IPv6 basic or advanced ACL, its ACL number and name must be unique in IPv6.

Match order

The rules in an ACL are sorted in a specific order. When a packet matches a rule, the device stops the match process and performs the action defined in the rule. If an ACL contains overlapping or conflicting rules, the matching result and action to take depend on the rule order.

The following ACL match orders are available:

- **config**—Sorts ACL rules in ascending order of rule ID. A rule with a lower ID is matched before a rule with a higher ID. If you use this method, check the rules and their order carefully.
- **auto**—Sorts ACL rules in depth-first order. Depth-first ordering makes sure that any subset of a rule is always matched before the rule. Table 1 lists the sequence of tie breakers that depth-first ordering uses to sort rules for each type of ACL.

Table 1 Sorting ACL rules in depth-first order

ACL category	Sequence of tie breakers
IPv4 basic ACL	**1.** VPN instance **2.** More 0s in the source IP address wildcard (more 0s means a narrower IP address range) **3.** Rule configured earlier
IPv4 advanced ACL	**1.** VPN instance **2.** Specific protocol number **3.** More 0s in the source IP address wildcard mask **4.** More 0s in the destination IP address wildcard **5.** Narrower TCP/UDP service port number range **6.** Rule configured earlier
IPv6 basic ACL	**1.** VPN instance **2.** Longer prefix for the source IP address (a longer prefix means a narrower IP address range) **3.** Rule configured earlier
IPv6 advanced ACL	**1.** VPN instance **2.** Specific protocol number **3.** Longer prefix for the source IPv6 address **4.** Longer prefix for the destination IPv6 address **5.** Narrower TCP/UDP service port number range **6.** Rule configured earlier
Ethernet frame header ACL	**1.** More 1s in the source MAC address mask (more 1s means a smaller MAC address) **2.** More 1s in the destination MAC address mask **3.** Rule configured earlier

A wildcard mask, also called an inverse mask, is a 32-bit binary and represented in dotted decimal notation. In contrast to a network mask, the 0 bits in a wildcard mask represent "do care" bits, and the 1 bits represent "don’t care" bits. If the "do care" bits in an IP address are identical to the "do care" bits in an IP address criterion, the IP address matches the criterion. All "don’t care" bits are ignored. The 0s and 1s in a wildcard mask can be noncontiguous. For example, 0.255.0.255 is a valid wildcard mask.

Rule comments and rule range remarks

Add a comment about an ACL rule to make it easy to understand. The rule comment appears below the rule statement.

In addition, add a rule range remark to indicate the start or end of a range of rules created for the same purpose.

Rule numbering

ACL rules can be manually numbered or automatically numbered. This section describes how automatic ACL rule numbering works.

Rule numbering step

If you do not assign an ID to the rule you are creating, the system automatically assigns it a rule ID. The rule numbering step sets the increment by which the system automatically numbers rules. For example, the default ACL rule numbering step is 5. If you do not assign IDs to rules you are creating, they are automatically numbered 0, 5, 10, 15, and so on. The wider the numbering step, the more rules you can insert between two rules.

By introducing a gap between rules rather than contiguously numbering rules, you have the flexibility of inserting rules in an ACL. This feature is important for a config order ACL, where ACL rules are matched in ascending order of rule ID.

Automatic rule numbering and renumbering

The ID automatically assigned to an ACL rule takes the nearest higher multiple of the numbering step to the current highest rule ID, starting with 0.

For example, if the numbering step is 5 (the default), and there are five ACL rules numbered 0, 5, 9, 10, and 12, the newly defined rule is numbered 15. If the ACL does not contain any rule, the first rule is numbered 0.

Whenever the step changes, the rules are renumbered, starting from 0. For example, if there are five rules numbered 5, 10, 13, 15, and 20, changing the step from 5 to 2 causes the rules to be renumbered 0, 2, 4, 6, and 8.

Implementing time-based ACL rules

You can implement ACL rules based on the time of day by applying a time range to them. A time-based ACL rule only takes effect in any time periods specified by the time range.

The following basic types of time range are available:

- **Periodic time range**—Recurs periodically on a day or days of the week.
- **Absolute time range**—Represents only a period of time and does not recur.

You can specify a time range in ACL rules before or after you create it. However, the rules using the time range take effect only after you define the time range.

IPv4 fragments filtering with ACLs

Traditional packet filtering matches only first fragments of packets, and allows all subsequent non-first fragments to pass through. Attackers can fabricate non-first fragments to attack networks.

To avoid the risks, the ACL implementation of Hewlett Packard Enterprise does the following:

- Filters all fragments by default, including non-first fragments.
- Allows for matching criteria modification, for example, filters non-first fragments only.

Configuration task list

Task	**Remarks**
Configuring a time range	Optional. Applicable to IPv4 and IPv6.
Configuring a basic ACL	Required. Configure at least one task. Basic ACLs and advanced ACLs are applicable to IPv4 and IPv6.
Configuring an advanced ACL	
Configuring an Ethernet frame header ACL	
Copying an ACL	Optional. Applicable to IPv4 and IPv6.

Task	Remarks
Enabling ACL acceleration for an IPv4 basic or IPv4 advanced ACL	Optional.

Configuring a time range

You can create a maximum of 256 time ranges, each having a maximum of 32 periodic statements and 12 absolute statements. If a time range has multiple statements, its active period is calculated as follows:

1. Combining all periodic statements.
2. Combining all absolute statements.
3. Taking the intersection of the two statement sets as the active period of the time range.

To configure a time range:

Step	Command	Remarks
4. Enter system view.	**system-view**	N/A
5. Configure a time range.	**time-range** *time-range-name* { *start-time* **to** *end-time days* [**from** *time1 date1*] [**to** *time2 date2*] | **from** *time1 date1* [**to** *time2 date2*] | **to** *time2 date2* }	By default, no time range exists. Repeat this command with the same time range name to create multiple statements for a time range.

Configuring a basic ACL

Configuring an IPv4 basic ACL

IPv4 basic ACLs match packets based only on source IP addresses.

To configure an IPv4 basic ACL:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an IPv4 basic ACL and enter its view.	**acl number** *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]	By default, no ACL exists. IPv4 basic ACLs are numbered in the range of 2000 to 2999. You can use the **acl name** *acl-name* command to enter the view of a named ACL.
3. Configure a description for the IPv4 basic ACL.	**description** *text*	Optional. By default, an IPv4 basic ACL has no ACL description.
4. Set the rule numbering step.	**step** *step-value*	Optional. The default setting is 5.

Step	Command	Remarks
5. Create or edit a rule.	**rule** [*rule-id*] { **deny** | **permit** } [**counting** | **fragment** | **logging** | **source** { *source-address source-wildcard* | **any** } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *	By default, an IPv4 basic ACL does not contain any rule. The **logging** keyword takes effect only when the module (for example, a firewall) that uses the ACL supports logging.
6. Add or edit a rule comment.	**rule** *rule-id* **comment** *text*	Optional. By default, no rule comments are configured.
7. Add or edit a rule range remark.	**rule** [*rule-id*] **remark** *text*	Optional. By default, no rule range remarks are configured.

Configuring an IPv6 basic ACL

IPv6 basic ACLs match packets based only on source IP addresses.

To configure an IPv6 basic ACL:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an IPv6 basic ACL view and enter its view.	**acl ipv6 number** *acl6-number* [**name** *acl6-name*] [**match-order** { **auto** | **config** }]	By default, no ACL exists. IPv6 basic ACLs are numbered in the range of 2000 to 2999. You can use the **acl ipv6 name** *acl6-name* command to enter the view of a named ACL.
3. Configure a description for the IPv6 basic ACL.	**description** *text*	Optional. By default, an IPv6 basic ACL has no ACL description.
4. Set the rule numbering step.	**step** *step-value*	Optional. The default setting is 5.
5. Create or edit a rule.	**rule** [*rule-id*] { **deny** | **permit** } [**counting** | **fragment** | **logging** | **routing** [**type** *routing-type*] | **source** { *source-address source-prefix* | *source-address/source-prefix* | **any** } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *	By default, an IPv6 basic ACL does not contain any rule. The **logging** keyword takes effect only when the module (for example, a firewall) using the ACL supports logging.
6. Add or edit a rule comment.	**rule** *rule-id* **comment** *text*	Optional. By default, no rule comments are configured.
7. Add or edit a rule range remark.	**rule** [*rule-id*] **remark** *text*	Optional. By default, no rule range remarks are configured.

Configuring an advanced ACL

Configuring an IPv4 advanced ACL

IPv4 advanced ACLs match packets based on source IPv4 addresses, destination IPv4 addresses, packet priorities, protocol numbers, and other protocol header information, such as TCP/UDP source and destination port numbers, TCP flags, ICMP message types, and ICMP message codes.

Compared to IPv4 basic ACLs, IPv4 advanced ACLs allow more flexible and accurate filtering.

To configure an IPv4 advanced ACL:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an IPv4 advanced ACL and enter its view.	**acl number** *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]	By default, no ACL exists. IPv4 advanced ACLs are numbered in the range of 3000 to 3999. You can use the **acl name** *acl-name* command to enter the view of a named ACL.
3. Configure a description for the IPv4 advanced ACL.	**description** *text*	Optional. By default, an IPv4 advanced ACL has no ACL description.
4. Set the rule numbering step.	**step** *step-value*	Optional. The default setting is 5.
5. Create or edit a rule.	**rule** [*rule-id*] { **deny** | **permit** } *protocol* [{ { **ack** *ack-value* | **fin** *fin-value* | **psh** *psh-value* | **rst** *rst-value* | **syn** *syn-value* | **urg** *urg-value* } * | **established** } | **counting** | **destination** { *dest-addr dest-wildcard* | **any** } | **destination-port** *operator port1* [*port2*] | **dscp** *dscp* | **fragment** | **icmp-type** { *icmp-type* [*icmp-code*] | *icmp-message* } | **logging** | **precedence** *precedence* | **reflective** | **source** { *sour-addr sour-wildcard* | **any** } | **source-port** *operator port1* [*port2*] | **time-range** *time-range-name* | **tos** *tos* | **vpn-instance** *vpn-instance-name*] *	By default, an IPv4 advanced ACL does not contain any rule. The **logging** keyword takes effect only when the module (for example, a firewall) using the ACL supports logging.
6. Add or edit a rule comment.	**rule** *rule-id* **comment** *text*	Optional. By default, no rule comments are configured.
7. Add or edit a rule range remark.	**rule** [*rule-id*] **remark** *text*	Optional. By default, no rule range remarks are configured.

Configuring an IPv6 advanced ACL

IPv6 advanced ACLs match packets based on the source IPv6 addresses, destination IPv6 addresses, packet priorities, protocol numbers, and other protocol header fields such as the TCP/UDP source port number, TCP/UDP destination port number, ICMPv6 message type, and ICMPv6 message code.

Compared to IPv6 basic ACLs, IPv6 advanced ACLs allow more flexible and accurate filtering.

To configure an IPv6 advanced ACL:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create an IPv6 advanced ACL and enter its view.	**acl ipv6 number** *acl6-number* [**name** *acl6-name*] [**match-order** { **auto** | **config** }]	By default, no ACL exists. IPv6 advanced ACLs are numbered in the range of 3000 to 3999. You can use the **acl ipv6 name** *acl6-name* command to enter the view of a named ACL.
3. Configure a description for the IPv6 advanced ACL.	**description** *text*	Optional. By default, an IPv6 advanced ACL has no ACL description.
4. Set the rule numbering step.	**step** *step-value*	Optional. The default setting is 5.
5. Create or edit a rule.	**rule** [*rule-id*] { **deny** | **permit** } *protocol* [{ { **ack** *ack-value* | **fin** *fin-value* | **psh** *psh-value* | **rst** *rst-value* | **syn** *syn-value* | **urg** *urg-value* } * | **established** } | **counting** | **destination** { *dest-address dest-prefix* | *dest-address/dest-prefix* | **any** } | **destination-port** *operator port1* [*port2*] | **dscp** *dscp* | **flow-label** *flow-label-value* | **fragment** | **icmp6-type** { *icmp6-type icmp6-code* | *icmp6-message* } | **logging** | **routing** [**type** *routing-type*] | **source** { *source-address source-prefix* | *source-address/source-prefix* **| any** } | **source-port** *operator port1* [*port2*] | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *	By default IPv6 advanced ACL does not contain any rule. The **logging** keyword takes effect only when the module (for example, a firewall) using the ACL supports logging.
6. Add or edit a rule comment.	**rule** *rule-id* **comment** *text*	Optional. By default, no rule comments are configured.
7. Add or edit a rule range remark.	**rule** [*rule-id*] **remark** *text*	Optional. By default, no rule range remarks are configured.

Configuring an Ethernet frame header ACL

Ethernet frame header ACLs, also called "Layer 2 ACLs," match packets based on Layer 2 protocol header fields, such as source MAC address, destination MAC address, 802.1p priority (VLAN priority), and link layer protocol type.

Ethernet frame header ACLs identifies Ethernet packets that are sent to the control plane (such as VTY and local user services), but not those sent to the forwarding plane (such as QoS, firewall, and debug services).

To configure an Ethernet frame header ACL:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Create an Ethernet frame header ACL and enter its view.	**acl number** *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]	By default, no ACL exists. Ethernet frame header ACLs are numbered in the range of 4000 to 4999. You can use the **acl name** *acl-name* command to enter the view of a named Ethernet frame header ACL.
3. Configure a description for the Ethernet frame header ACL.	**description** *text*	Optional. By default, an Ethernet frame header ACL has no ACL description.
4. Set the rule numbering step.	**step** *step-value*	Optional. The default setting is 5.
5. Create or edit a rule.	**rule** [*rule-id*] { **deny** | **permit** } [**cos** *vlan-pri* | **counting** | **dest-mac** *dest-address dest-mask* | { **lsap** *lsap-type lsap-type-mask* | **type** *protocol-type protocol-type-mask* } | **source-mac** *source-address source-mask* | **time-range** *time-range-name*] *	By default, an Ethernet frame header ACL does not contain any rule.
6. Add or edit a rule comment.	**rule** *rule-id* **comment** *text*	Optional. By default, no rule comments are configured.
7. Add or edit a rule range remark.	**rule** [*rule-id*] **remark** *text*	Optional. By default, no rule range remarks are configured.

Copying an ACL

You can create an ACL by copying an existing ACL (source ACL). The new ACL (destination ACL) has the same properties and content as the source ACL, but not the same ACL number and name.

To successfully copy an ACL, make sure that:

- The destination ACL number is from the same category as the source ACL number.
- The source ACL already exists, but the destination ACL does not.

Copying an IPv4 basic, IPv4 advanced, or Ethernet frame header ACL

Step	Command
1. Enter system view.	**system-view**
2. Copy an existing IPv4 basic, IPv4 advanced, or Ethernet frame header ACL to create a new ACL.	**acl copy** { *source-acl-number* | **name** *source-acl-name* } **to** { *dest-acl-number* | **name** *dest-acl-name* }

Copying an IPv6 basic or IPv6 advanced ACL

Step	Command
1. Enter system view.	**system-view**
2. Copy an existing IPv6 basic or IPv6 advanced ACL to create a new ACL.	**acl ipv6 copy** { *source-acl6-number* | **name** *source-acl6-name* } **to** { *dest-acl6-number* | **name** *dest-acl6-name* }

Enabling ACL acceleration for an IPv4 basic or IPv4 advanced ACL

△ **CAUTION:**

- ACL acceleration is not available for ACLs that contain a non-contiguous wildcard mask.
- After you modify an ACL with ACL acceleration enabled, disable and re-enable ACL acceleration to ensure correct rule matching.

ACL acceleration speeds up ACL lookup. The acceleration effect increases with the number of ACL rules. ACL acceleration uses memory. To achieve the best trade-off between memory and ACL processing performance, Hewlett Packard Enterprise recommends enabling ACL acceleration for large ACLs, for example, ACLs containing more than 50 rules.

For example, when you use a large ACL for a session-based service, you can enable ACL acceleration to avoid session timeouts caused by ACL processing delays.

Enable ACL acceleration in an ACL after you have finished editing ACL rules. ACL acceleration always uses ACL criteria that have been set before it is enabled for rule matching. It does not synchronize with any subsequent match criterion changes.

To enable ACL acceleration for an IPv4 basic or IPv4 advanced ACL:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enable ACL acceleration for an IPv4 basic or IPv4 advanced ACL.	**acl accelerate number** *acl-number*	By default, the function is disabled. The ACL must exist. Only IPv4 basic ACLs and advanced ACLs support ACL acceleration.

Displaying and maintaining ACLs

Task	Command	Remarks
Display configuration and match statistics for IPv4 basic, IPv4 advanced, and Ethernet frame header ACLs.	**display acl** { *acl-number* | **all** | **name** *acl-name* } [**slot** *slot-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display information about the ACL acceleration feature.	**display acl accelerate** { *acl-number* | **all** } [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display configuration and match statistics for IPv6 basic and IPv6 advanced ACLs.	**display acl ipv6** { *acl6-number* | **all** | **name** *acl6-name* } [**slot** *slot-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display the usage of ACL rules. (See the following matrix for information about the support for the command.)	**display acl resource** [**slot** *slot-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display the configuration and status of one or all time ranges.	**display time-range** { *time-range-name* | **all** } [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Clear statistics for one or all IPv4 basic, IPv4 advanced, and Ethernet frame header ACLs.	**reset acl counter** { *acl-number* | **all** | **name** *acl-name* }	Available in user view.
Clear statistics for one or all IPv6 basic and advanced ACLs.	**reset acl ipv6 counter** { *acl6-number* | **all** | **name** *acl6-name* }	Available in user view.

The following matrix shows the **display acl resource** command and hardware compatibility:

Hardware	Compatibility
HSR6602	No
6604/6608/6616	• Routers with RPE-X1 or RSE-X1 installed: Yes • Routers with MCP installed: No

ACL configuration examples

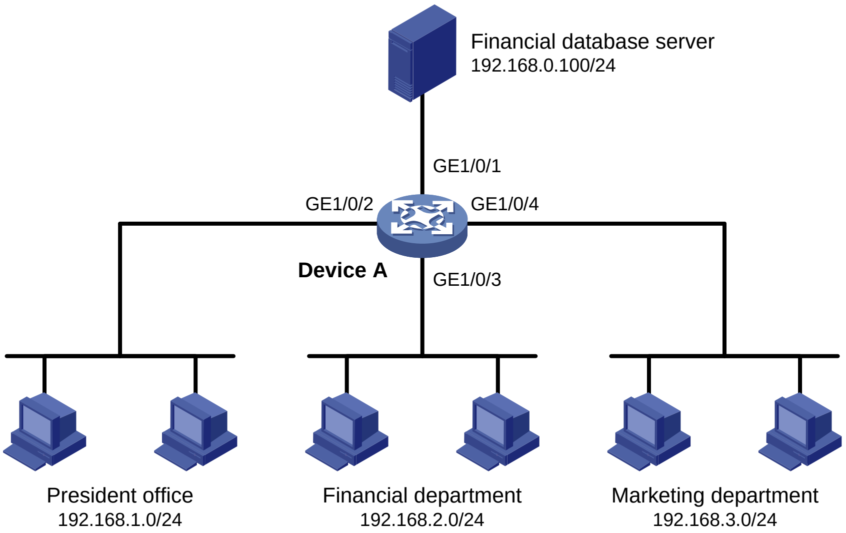
IPv4 advanced ACL configuration examples

Network requirements

A company interconnects its departments through Router A. Configure an ACL to:

- Permit access from the President office at any time to the financial database server.
- Permit access from the Financial department to the database server only during working hours (from 8:00 to 18:00) on working days.
- Deny access from any other department to the database server.

Figure 1 Network diagram



Configuration procedure

# Create a periodic time range from 8:00 to 18:00 on working days.

```
<RouterA> system-view
[RouterA] time-range work 8:0 to 18:0 working-day
```

# Create an IPv4 advanced ACL numbered 3000 and configure three rules in the ACL. One rule permits access from the President office to the financial database server, one rule permits access from the Financial department to the database server during working hours, and one rule denies access from any other department to the database server.

```
[RouterA] acl number 3000
[RouterA-acl-adv-3000] rule permit ip source 192.168.1.0 0.0.0.255 destination
192.168.0.100 0
[RouterA-acl-adv-3000] rule permit ip source 192.168.2.0 0.0.0.255 destination
192.168.0.100 0 time-range work
[RouterA-acl-adv-3000] rule deny ip source any destination 192.168.0.100 0
[RouterA-acl-adv-3000] quit
```

# Enable IPv4 firewall, and apply IPv4 advanced ACL 3000 to filter outgoing packets on interface GigabitEthernet 1/0/1.

```
[RouterA] firewall enable
[RouterA] interface gigabitethernet 1/0/1
[RouterA-GigabitEthernet1/0/1] firewall packet-filter 3000 outbound
[RouterA-GigabitEthernet1/0/1] quit
```

Verifying the configuration

# Ping the database server from a PC in the Financial department during working hours. (All PCs in this example use Windows XP.)

```
C:\> ping 192.168.0.100

Pinging 192.168.0.100 with 32 bytes of data:

Reply from 192.168.0.100: bytes=32 time=1ms TTL=255
Reply from 192.168.0.100: bytes=32 time<1ms TTL=255
Reply from 192.168.0.100: bytes=32 time<1ms TTL=255
Reply from 192.168.0.100: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.0.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

The output shows the database server can be pinged.

# Ping the database server from a PC in the Marketing department during working hours.

```
C:\> ping 192.168.0.100

Pinging 192.168.0.100 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The output shows the database server cannot be pinged.

# Display configuration and match statistics for IPv4 advanced ACL 3000 on Device A during working hours.

```
[RouterA] display acl 3000
Advanced ACL  3000, named -none-, 3 rules,
ACL's step is 5
 rule 0 permit ip source 192.168.1.0 0.0.0.255 destination 192.168.0.100 0
 rule 5 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.0.100 0 time-range work
(4 times matched) (Active)
 rule 10 deny ip destination 192.168.0.100 0 (4 times matched)
```

The output shows rule 5 is active. Rule 5 and rule 10 have been matched four times as the result of the ping operations.

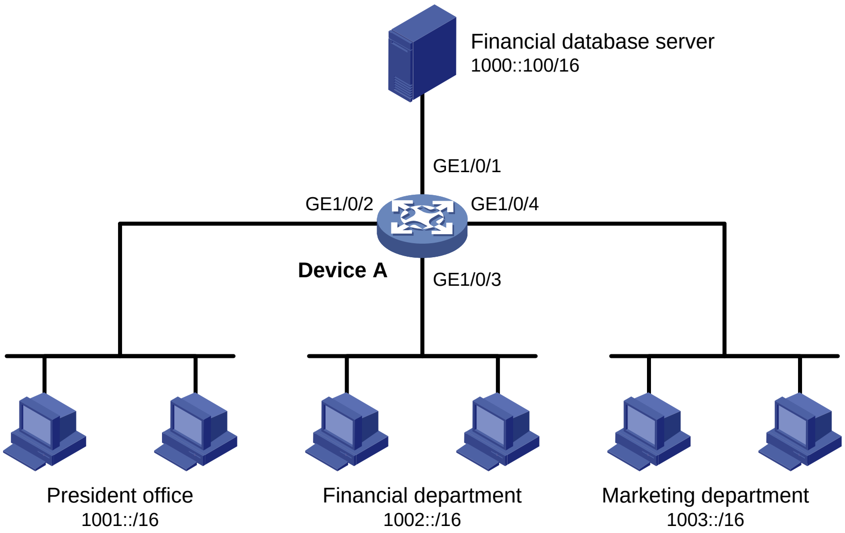
IPv6 advanced ACL configuration example

Network requirements

A company interconnects its departments through Router A. Configure an ACL to do the following:

- Permit access from the President office at any time to the financial database server.
- Permit access from the Financial department to the database server only during working hours (from 8:00 to 18:00) on working days.
- Deny access from any other department to the database server.

Figure 2 Network diagram



Configuration procedure

# Create a periodic time range from 8:00 to 18:00 on working days.

```
<RouterA> system-view
[RouterA] time-range work 8:0 to 18:0 working-day
```

# Create an IPv6 advanced ACL numbered 3000 and configure three rules in the ACL. One rule permits access from the President office to the database server, one rule permits access from the Financial department to the database server during working hours, and one rule denies access from other departments to the database server.

```
[RouterA] acl ipv6 number 3000
[RouterA-acl6-adv-3000] rule permit ipv6 source 1001:: 16 destination 1000::100 128
[RouterA-acl6-adv-3000] rule permit ipv6 source 1002:: 16 destination 1000::100 128
time-range work
[RouterA-acl6-adv-3000] rule deny ipv6 source any destination 1000::100 128
[RouterA-acl6-adv-3000] quit
```

# Enable IPv6 firewall, and apply IPv6 advanced ACL 3000 to filter outgoing IPv6 packets on interface GigabitEthernet 1/0/1.

```
[RouterA] firewall ipv6 enable
[RouterA] interface gigabitethernet 1/0/1
[RouterA-GigabitEthernet1/0/1] firewall packet-filter ipv6 3000 outbound
```

Verifying the configuration

# Ping the database server from a PC in the Financial department during working hours. (All PCs in this example use Windows XP.)

```
C:\> ping 1000::100

Pinging 1000::100 with 32 bytes of data:

Reply from 1000::100: time<1ms
Reply from 1000::100: time<1ms
Reply from 1000::100: time<1ms
Reply from 1000::100: time<1ms
```

```
Ping statistics for 1000::100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The output shows that the database server can be pinged.

# Ping the database server from a PC in the Marketing department during working hours.

```
C:\> ping 1000::100

Pinging 1000::100 with 32 bytes of data:

Destination net unreachable.
Destination net unreachable.
Destination net unreachable.
Destination net unreachable.

Ping statistics for 1000::100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The output shows the database server cannot be pinged.

# Display configuration and match statistics for IPv6 advanced ACL 3000 on Device A during working hours.

```
[RouterA] display acl ipv6 3000
 Advanced IPv6 ACL  3000, named -none-, 3 rules,
 ACL's step is 5
 rule 0 permit ipv6 source 1001::/16 destination 1000::100/128
 rule 5 permit ipv6 source 1002::/16 destination 1000::100/128 time-range work (4 times matched) (Active)
 rule 10 deny ipv6 destination 1000::100/128 (4 times matched)
```

The output shows rule 5 is active. Rule 5 and rule 10 have been matched four times as the result of the ping operations.

QoS overview

In data communications, Quality of Service (QoS) is a network's ability to provide differentiated service guarantees for diversified traffic in terms of bandwidth, delay, jitter, and drop rate.

Network resources are scarce. The contention for resources requires that QoS prioritize important traffic flows over trivial ones. For example, when bandwidth is fixed, more bandwidth for one traffic flow means less bandwidth for the other traffic flows. When making a QoS scheme, consider the characteristics of various applications to balance the interests of diversified users and to utilize network resources.

The following section describes some typical QoS service models and widely used, mature QoS techniques.

QoS service models

Best-effort service model

The best-effort model is a single-service model and is also the simplest service model. In this service model, the network does its best to deliver packets, but does not guarantee delay or reliability.

The best-effort service model is the default model in the Internet and applies to most network applications. It uses the first in first out (FIFO) queuing mechanism.

IntServ model

The integrated service (IntServ) model is a multiple-service model that can accommodate diverse QoS requirements. This service model provides the most granularly differentiated QoS by identifying and guaranteeing definite QoS for each data flow.

In the IntServ model, an application must request service from the network before it sends data. IntServ signals the service request with the Resource Reservation Protocol (RSVP). All nodes receiving the request reserve resources as requested and maintain state information for the application flow.

The IntServ model demands high storage and processing capabilities because it requires all nodes along the transmission path to maintain resource state information for each flow. This model is suitable for small-sized or edge networks, but not large-sized networks, for example, the core layer of the Internet, where billions of flows are present.

For more information about RSVP, see *MPLS Configuration Guide*.

DiffServ model

The differentiated service (DiffServ) model is a multiple-service model that can meet diverse QoS requirements. It is easy to implement and extend. DiffServ does not signal the network to reserve resources before sending data, as IntServ does.

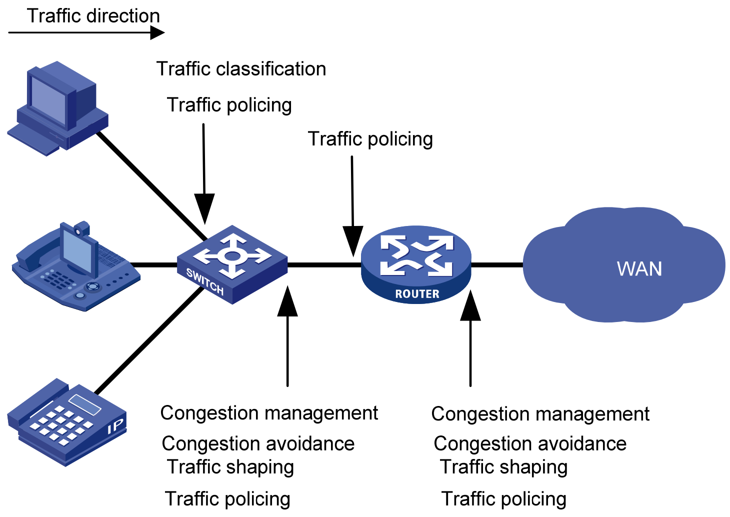
All QoS techniques in this document are based on the DiffServ model.

QoS techniques overview

The QoS techniques include traffic classification, traffic policing, traffic shaping, rate limit, congestion management, and congestion avoidance. The following section briefly introduces these QoS techniques.

Deploying QoS in a network

Figure 3 Position of the QoS techniques in a network



As shown in Figure 3, traffic classification, traffic shaping, traffic policing, congestion management, and congestion avoidance mainly implement the following functions:

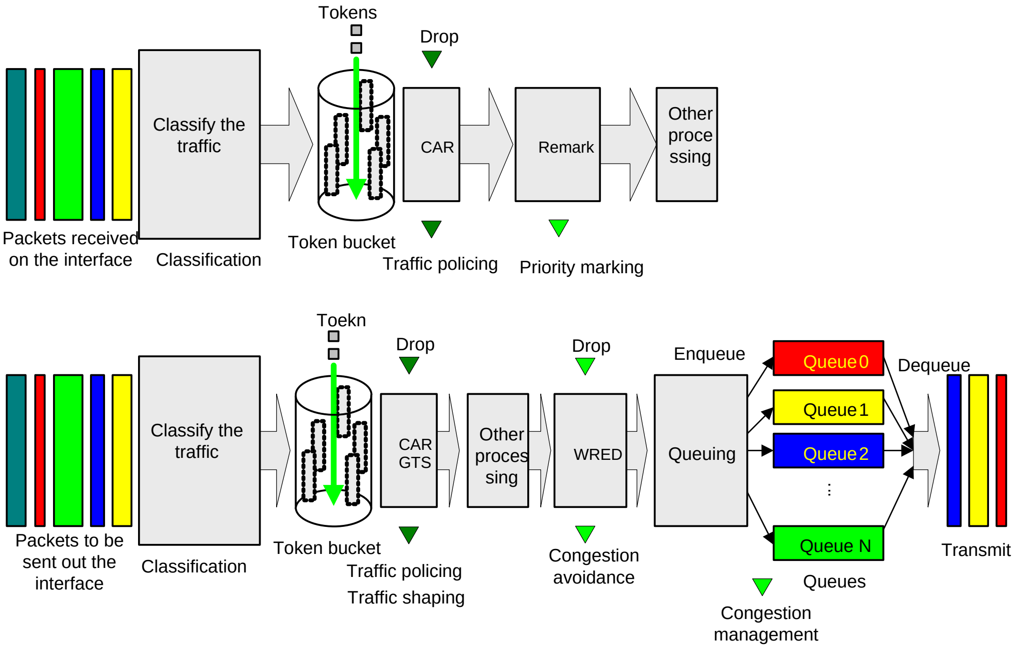
- **Traffic classification**—Uses certain match criteria to assign packets with the same characteristics to a class. Based on classes, you can provide differentiated services.
- **Traffic policing**—Polices flows entering or leaving a device, and imposes penalties on traffic flows that exceed the preset threshold to prevent aggressive use of network resources. You can apply traffic policing to both incoming and outgoing traffic of a port.
- **Traffic shaping**—Proactively adapts the output rate of traffic to the network resources available on the downstream device to eliminate packet drops. Traffic shaping usually applies to the outgoing traffic of a port.
- **Congestion management**—Provides a resource scheduling policy to determine the packet forwarding sequence when congestion occurs. Congestion management usually applies to the outgoing traffic of a port.
- **Congestion avoidance**—Monitors the network resource usage, and is usually applied to the outgoing traffic of a port. When congestion worsens, congestion avoidance reduces the queue length by dropping packets.

QoS processing flow in a device

Figure 4 briefly describes how the QoS module processes traffic:

1. Traffic classifier identifies and classifies traffic for subsequent QoS actions.
2. The QoS module takes various QoS actions on classified traffic as configured, depending on the traffic processing phase and network status. For example, you can configure the QoS module to perform traffic policing for incoming traffic, traffic shaping for outgoing traffic, congestion avoidance before congestion occurs, and congestion management when congestion occurs.

Figure 4 QoS processing flow



QoS configuration approaches

QoS configuration approach overview

You can configure QoS in the following approaches:

- MQC approach
- Non-MQC approach

Some features support both approaches, but some support only one.

MQC approach

In the modular QoS configuration (MQC) approach, you configure QoS service parameters by using QoS policies. A QoS policy defines the shaping, policing, or other QoS actions to take on different classes of traffic. It is a set of class-behavior associations.

A class is a set of match criteria for identifying traffic, and it uses the AND or OR operator:

- If the operator is AND, a packet must match all the criteria to match the class.
- If the operator is OR, a packet matches the class if it matches any of the criteria in the class.

A traffic behavior defines a set of QoS actions to take on packets, such as priority marking and redirect.

By associating a traffic behavior with a class in a QoS policy, you apply the specific set of QoS actions to the class of traffic.

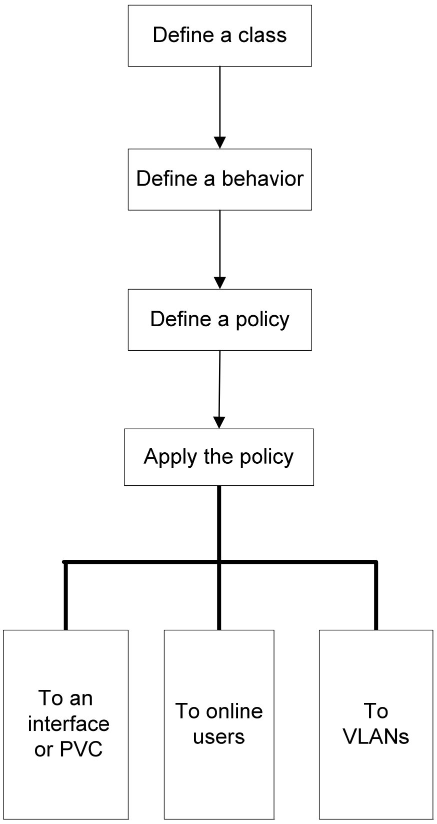
Non-MQC approach

In the non-MQC approach, you configure QoS service parameters without using a QoS policy. For example, you can use the rate limit feature to set a rate limit on an interface without using a QoS policy.

Configuring a QoS policy

Figure 5 shows how to configure a QoS policy.

Figure 5 QoS policy configuration procedure



Defining a class

The system predefines some classes and defines general match criteria for them. A user-defined class cannot be named the same as a system-defined class. You can use these predefined classes when defining a policy. The system-defined classes include:

- The default class

  default-class: Matches the default traffic.
- DSCP-based predefined classes

  ef, af1, af2, af3, af4: Matches IP DSCP value ef, af1, af2, af3, af4, respectively.
- IP precedence-based predefined classes

  ip-prec0, ip-prec1, …ip-prec7: Matches IP precedence value 0, 1, …7, respectively.
- MPLS EXP-based predefined classes

  mpls-exp0, mpls-exp1, …mpls-exp7: Matches MPLS EXP value 0, 1, …7, respectively.

To define a class:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

2. Create a class and enter class mapping view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	By default, the operator of a class is AND. The operator of a class can be AND or OR. • **AND**—A packet is assigned to a class only when the packet matches all the criteria in the class. • **OR**—A packet is assigned to a class if it matches any of the criteria in the class.
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	For more information, see the **if-match** command in *ACL and QoS Command Reference*.

Defining a traffic behavior

A traffic behavior is a set of QoS actions (such as traffic filtering, shaping, policing, and priority marking) to take on a class of traffic.

The system predefines some traffic behaviors and defines general QoS actions for them. A user-defined behavior cannot be named the same as a system-defined behavior. You can use these behaviors when defining a policy. The system-defined behaviors are as follows:

- **ef**—Expedited forwarding.
- **af**—Assured forwarding.
- **be**—Best-effort.
- **be-flow-based**—Uses the weighted random early detection (WRED) drop policy.

For more information about these system-defined behaviors, see "Configuring congestion management."

To define a traffic behavior:

Step	Command
1. Enter system view.	**system-view**
2. Create a traffic behavior and enter traffic behavior view	**traffic behavior** *behavior-name*
3. Configure actions in the traffic behavior.	See the subsequent chapters, depending on the purpose of the traffic behavior: traffic policing, traffic filtering, traffic redirecting, priority marking, traffic accounting, and so on.

Defining a policy

You associate a behavior with a class in a QoS policy to perform the actions defined in the behavior for the class of packets.

You can configure multiple class-behavior associations in a QoS policy, which are matched in the order they are configured.

The system provides a predefined QoS policy named **default**. It includes the associations between predefined classes and predefined traffic behaviors:

- Class **ef** with behavior **ef**.
- Classes **af1** through **af4** with behavior **af**.
- Class **default-class** with behavior **be**.

You cannot name a user-defined QoS policy the same as the system-defined QoS policy.

To associate a class with a behavior in a policy:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
3. Associate a class with a behavior in the policy.	**classifier** *classifier-name* **behavior** *behavior-name*	Repeat this step to create more class-behavior associations.

IMPORTANT:

- If the ACL contains deny rules, the if-match clause is ignored and the matching process continues.
- If a QoS policy uses a Layer 2 ACL for classification, the Layer 2 ACL does not support the **lsap** keyword.

Configuring QoS policy nesting

You can reference a QoS policy in a traffic behavior to re-classify the traffic class associated with the behavior and take action on the re-classified traffic as defined in the policy. The QoS policy referenced in the traffic behavior is called the "child policy." The QoS policy that references the behavior is called the "parent policy."

To nest QoS policies successfully, follow these guidelines:

- If class-based queuing (CBQ) is configured in the child policy, configure generic traffic shaping (GTS) in the parent policy and make sure that the GTS bandwidth configured in the parent policy is equal to or greater than the CBQ bandwidth configured in the child policy.
- If GTS bandwidth in the parent policy is configured in percentage, the CBQ bandwidth in the child policy must be also configured in percentage. If it is configured as an absolute number, the CBQ bandwidth in the child policy can be configured in either percentage or as an absolute number.
- GTS cannot be configured in the child policy.

To nest a child QoS policy in a parent QoS policy:

Step	**Command**	**Remarks**
1. Enter system view.	**system-view**	N/A
2. Create a class for the parent policy and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior for the parent policy and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Nest the child QoS policy.	**traffic-policy** *policy-name*	The QoS policy specified for the *policy-name* argument must already exist.
7. Return to system view.	**quit**	N/A
8. Create the parent policy and enter parent policy view.	**qos policy** *policy-name*	N/A

Step	Command	Remarks
9. Associate the class with the behavior in the parent policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A

Applying the QoS policy

You can apply a QoS policy to the following destinations:

- **An interface or PVC**—The policy takes effect on the traffic sent or received on the interface or PVC.
- **A user profile**—The policy takes effect on the traffic sent or received by the online users of the user profile.
- **A VLAN**—The policy takes effect on the traffic sent or received on all ports in the VLAN. This destination is supported only on SAP modules operating in bridge mode.

Applying the QoS policy to an interface or PVC

A policy can be applied to multiple interfaces or PVCs, but only one policy can be applied in one direction (inbound or outbound) of an interface or PVC.

When you apply the QoS policy to an interface or PVC, follow these guidelines:

- You can apply QoS policies to all physical interfaces but X.25- or LAPB-enabled interfaces.
- The QoS policy applied to the outgoing traffic on an interface or PVC does not regulate local packets, which are critical protocol packets sent by the local system for operation maintenance. The most common local packets include link maintenance, routing (IS-IS, BGP, RIP, and OSPF, for example), LDP, and SSH packets.
- On some cards, QoS policies can be applied but cannot take effect due to limited system resources. In this case, you can adjust related parameters (for example, reducing the number of queues) according to system prompt and then apply a QoS policy again.

To apply the QoS policy to an interface or PVC:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or PVC view.	• Enter interface view: **interface** *interface-type interface-number* • Enter PVC view: **a. interface atm** *interface-number* **b. pvc** *vpi/vci*	Settings in interface view take effect on the current interface. Settings in PVC view take effect on the current PVC.
3. Apply the policy to the interface or PVC.	**qos apply policy** *policy-name* { **inbound** | **outbound** }	N/A

Applying the QoS policy to online users

You can apply a QoS policy to multiple online users. In one direction of each online user, only one policy can be applied. To modify a QoS policy already applied in a certain direction, remove the QoS policy application first.

When you apply the QoS policy to online users, follow these guidelines:

- You can only edit or remove the configurations in a disabled user profile. Disabling a user profile logs out the users that are using the user profile.

- The QoS policy applied to a user profile supports only the **remark**, **car**, and **filter** actions.
- Do not apply a null policy to a user profile. The user profile using a null policy cannot be activated.
- The authentication methods supported for online users include PPPoE, 802.1X, Portal, and MAC authentication.

To apply the QoS policy to online users:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter user profile view.	**user-profile** *profile-name*	The configuration made in user profile view takes effect when the user profile is activated and the users of the user profile are online. For more information about user profiles, see *Security Configuration Guide*.
3. Apply the QoS policy.	**qos apply policy** *policy-name* { **inbound** | **outbound** }	Use the **inbound** keyword to apply the QoS policy to the incoming traffic of the device (traffic sent by the online users). Use the **outbound** keyword to apply the QoS policy to the outgoing traffic (traffic received by the online users).
4. Return to system view.	**quit**	N/A
5. Activate the user profile.	**user-profile** *profile-name* **enable**	By default, user profiles are inactive.

Applying the QoS policy to a VLAN

You can apply a QoS policy to a VLAN to regulate traffic of the VLAN.

When you apply the QoS policy to a VLAN, follow these guidelines:

- QoS policies cannot be applied to dynamic VLANs, such as VLANs created by GVRP.
- When you apply a QoS policy to VLANs, the QoS policy is applied to the specified VLANs on all interface cards. If the hardware resources of an interface card are insufficient, applying a QoS policy to VLANs might fail on the interface card. The system does not automatically roll back the QoS policy configuration already applied to the main processing unit or other interface cards. To ensure consistency, use the **undo qos vlan-policy vlan** command to manually remove the QoS policy configuration applied to them.

To apply the QoS policy to a VLAN:

Step	Command
1. Enter system view.	**system-view**
2. Apply the QoS policy to VLANs.	**qos vlan-policy** *policy-name* **vlan** *vlan-id-list* { **inbound** | **outbound** }

Displaying and maintaining QoS policies

Task	Command	Remarks
Display a specified class-behavior association in a specified policy or all class-behavior associations in a specified policy or in all policies.	**display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Display QoS policy configuration on a specified interface or PVC or all interfaces or PVCs.	**display qos policy interface** [{ *interface-type interface-number* } [**slot** *slot-number*]] [**inbound** | **outbound**] [**pvc** { *pvc-name* [*vpi/vci*] | *vpi/vci* }] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display information about QoS policies applied to VLANs.	**display qos vlan-policy** { **name** *policy-name* | **vlan** *vlan-id* } [**slot** *slot-number*] [**inbound** | **outbound**] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view. This command is supported only on SAP modules operating in bridge mode.
Display traffic behavior configurations.	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display traffic class configurations.	**display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Clear the statistics of the QoS policy applied to a VLAN.	**reset qos vlan-policy** [**vlan** *vlan-id*] [**inbound** | **outbound**]	Available in user view. This command is supported only on SAP modules operating in bridge mode.
Display a specified class-behavior association in a specified policy.	**display qos policy user-defined** [*policy-name* [**classifier** *classifier-name*]] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Configuring priority mapping

This feature is supported only on SAP modules operating in bridge mode.

Overview

When a packet arrives, depending on your configuration, a device assigns a set of QoS priority parameters to the packet based on either a certain priority field carried in the packet or the port priority of the incoming port. This process is called "priority mapping." During this process, the device can modify the priority of the packet depending on device status. The set of QoS priority parameters decides the scheduling priority and forwarding priority of the packet.

Priority mapping is implemented with priority mapping tables and involves priorities such as 802.11e priority, 802.1p priority, DSCP, EXP, IP precedence, local precedence, and drop precedence.

Introduction to priorities

Priorities include the following types: priorities carried in packets, and priorities locally assigned for scheduling only.

The packet-carried priorities include 802.1p priority, DSCP precedence, IP precedence, EXP, and so on. These priorities have global significance and affect the forwarding priority of packets across the network. For more information about these priorities, see "Appendix."

The locally assigned priorities only have local significance. They are assigned by the device for scheduling only. These priorities include the following types:

- **Local precedence**—Local precedence is used for queuing. A local precedence value corresponds to an output queue. A packet with higher local precedence is assigned to a higher priority output queue to be preferentially scheduled.
- **Drop precedence**—Drop precedence is used for making packet drop decisions. Packets with the highest drop precedence are dropped preferentially.

Priority mapping tables

The device provides various types of priority mapping tables, or rather, priority mappings. By looking up a priority mapping table, the device decides which priority value is to assign to a packet for subsequent packet processing.

The default priority mapping tables (as shown in Appendix B Default priority mapping tables) are available for priority mapping. They are adequate in most cases. If a default priority mapping table cannot meet your requirements, you can modify the priority mapping table as required.

Priority mapping configuration tasks

You can configure priority mapping in any of the following approaches:

- Configuring priority trust mode.

  In this approach, you can configure a port to look up a certain priority, 802.1p for example, in incoming packets, in the priority mapping tables. If no packet priority is trusted, the port priority of the incoming port is used.
- Changing port priority.

  By default, all ports are assigned the port priority of zero. By changing the port priority of a port, you change the priority of the incoming packets on the port.

Perform these tasks to configure priority mapping:

Task	Remarks
Configuring a priority mapping table	Optional.
Configuring the trusted packet priority type for an interface or port group	Optional.
Changing the port priority of an interface	Optional.

Configuring a priority mapping table

The router provides the following types of priority mapping table.

Table 2 Priority mapping tables

Priority mapping	Description
dot1p-dp	802.1p-drop mapping table.
dot1p-lp	802.1p-local mapping table.
dscp-dot1p	DSCP-802.1p mapping table.
dscp-dp	DSCP-drop mapping table.
dscp-dscp	DSCP-DSCP mapping table.

To configure a priority mapping table:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter priority mapping table view.	**qos map-table** { **dot1p-dp** | **dot1p-lp** | **dscp-dot1p** | **dscp-dp** | **dscp-dscp** }	For the DSCP-to-drop mapping table, the router does not support mapping DSCP values to drop precedence 1.
3. Configure the priority mapping table.	**import** *import-value-list* **export** *export-value*	Newly configured mappings overwrite the old ones.

Configuring the trusted packet priority type for an interface or port group

You can configure the router to trust a particular priority field carried in packets for priority mapping on a port or port group.

To configure the trusted packet priority type on an interface or port group:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

2. Enter interface view or port group view.	- Enter interface view: **interface** *interface-type interface-number* - Enter port group view: **port-group manual** *port-group-name*	Use one of the commands. Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
3. Configure the trusted packet priority type for the interface.	**qos trust dscp**	By default, no trusted packet priority type is configured.

Changing the port priority of an interface

If an interface does not trust any packet priority, the router uses its port priority to look for the set of priority parameters for the incoming packets. By changing port priority, you can prioritize traffic received on different interfaces.

To change the port priority of an interface:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or port group view.	- Enter interface view: **interface** *interface-type interface-number* - Enter port group view: **port-group manual** *port-group-name*	Use one of the commands. Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
3. Set the port priority of the interface.	**qos priority** *priority-value*	The default is 0.

Displaying and maintaining priority mapping

Task	Command	Remarks
Display priority mapping table configuration.	**display qos map-table** [**dot1p-dp** | **dot1p-lp** | **dscp-dot1p** | **dscp-dp** | **dscp-dscp**] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display the trusted packet priority type on a port.	**display qos trust interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Priority mapping configuration examples

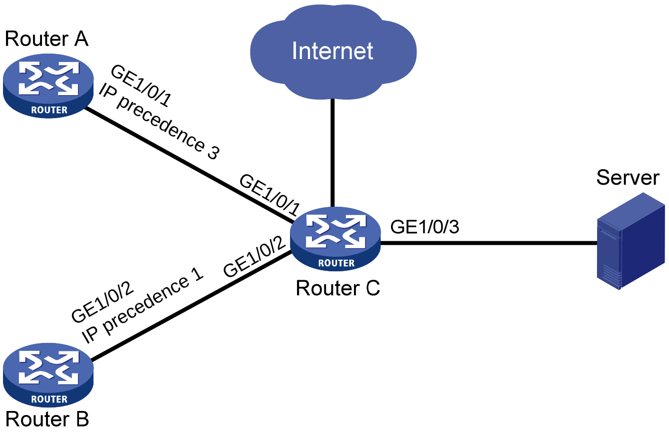
Priority trust mode and port priority configuration example

Network requirements

As shown in Figure 6, the IP precedence of Router A's traffic is 3, and the IP precedence of Router B's traffic is 1.

Configure Router C to preferentially process packets from Router A to Server when GigabitEthernet 1/0/3 of Router C is congested.

Figure 6 Network diagram



Configuration procedure

1. Method 1: Configure Router C to trust DSCP.

   # Configure GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2 to trust DSCP.

```
<RouterC> system-view
[RouterC] interface gigabitethernet 1/0/1
[RouterC-GigabitEthernet1/0/1] qos trust dscp
[RouterC-GigabitEthernet1/0/1] quit
[RouterC] interface gigabitethernet 1/0/2
[RouterC-GigabitEthernet1/0/2] qos trust dscp
[RouterC-GigabitEthernet1/0/2] quit
```

2. Method 2: Configure Router C to trust port priority.

   # Assign port priorities to GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2. Make sure that the priority of GigabitEthernet 1/0/1 is higher than GigabitEthernet 1/0/2, and no trusted packet priority type is configured on GigabitEthernet 1/0/1 or GigabitEthernet 1/0/2.

```
<RouterC> system-view
[RouterC] interface gigabitethernet 1/0/1
[RouterC-GigabitEthernet1/0/1] qos priority 3
[RouterC-GigabitEthernet1/0/1] quit
[RouterC] interface gigabitethernet 1/0/2
[RouterC-GigabitEthernet1/0/2] qos priority 1
[RouterC-GigabitEthernet1/0/2] quit
```

Priority mapping table configuration example

Network requirements

As shown in Figure 7:

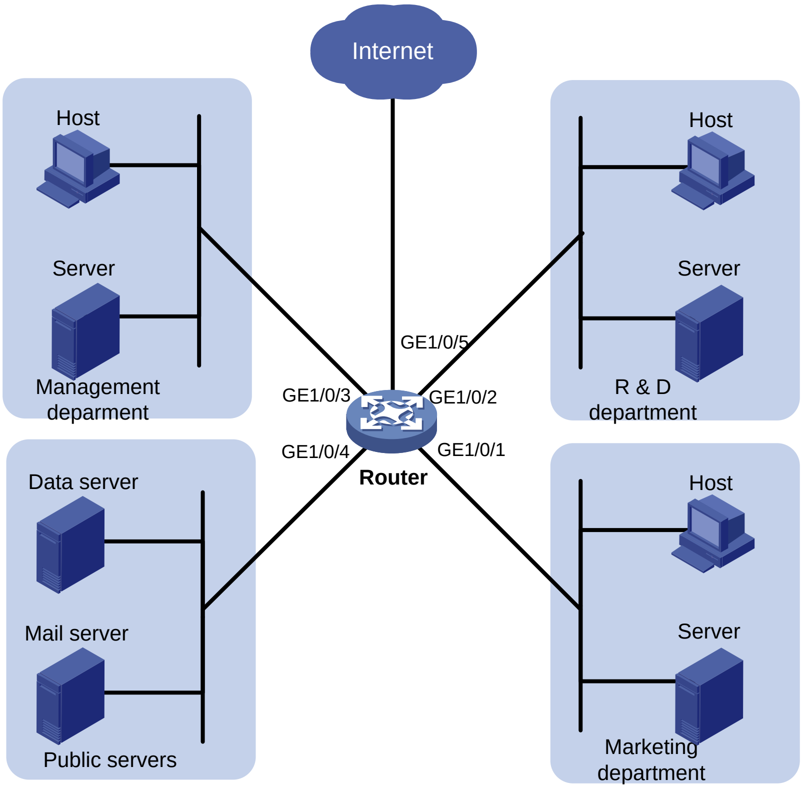
- The marketing department connects to GigabitEthernet 1/0/1 of Router, which sets the 802.1p priority of traffic from the marketing department to 3.
- The R&D department connects to GigabitEthernet 1/0/2 of Router, which sets the 802.1p priority of traffic from the R&D department to 4.
- The management department connects to GigabitEthernet 1/0/3 of Router, which sets the 802.1p priority of traffic from the management department to 5.

Configure port priority, 802.1p-to-local mapping table, and priority marking to implement the plan as described in Table 3.

Table 3 Configuration plan

Traffic destination	Traffic priority order	Queuing plan		
		Traffic source	**Output queue**	**Queue priority**
Public servers	R&D department > management department > marketing department	R&D department	6	High
		Management department	4	High
		Marketing department	2	Low
Internet	Management department > marketing department > R&D department	R&D department	2	Low
		Management department	6	High
		Marketing department	3	Medium

Figure 7 Network diagram



Configuration procedure

1. Configure trusting port priority:

   # Set the port priority of GigabitEthernet 1/0/1 to 3.

```
<Router> system-view
[Router] interface gigabitethernet 1/0/1
[Router-GigabitEthernet1/0/1] qos priority 3
[Router-GigabitEthernet1/0/1] quit
```

   # Set the port priority of GigabitEthernet 1/0/2 to 4.

```
[Router] interface gigabitethernet 1/0/2
[Router-GigabitEthernet1/0/2] qos priority 4
[Router-GigabitEthernet1/0/2] quit
```

   # Set the port priority of GigabitEthernet 1/0/3 to 5.

```
[Router] interface gigabitethernet 1/0/3
[Router-GigabitEthernet1/0/3] qos priority 5
[Router-GigabitEthernet1/0/3] quit
```

2. Configure the priority mapping table:

   # Configure the 802.1p-to-local mapping table to map 802.1p priority values 3, 4, and 5 to local precedence values 2, 6, and 4. This guarantees the R&D department, management department, and marketing department decreased priorities to access the public server.

```
[Router] qos map-table dot1p-lp
[Router-maptbl-dot1p-lp] import 3 export 2
[Router-maptbl-dot1p-lp] import 4 export 6
[Router-maptbl-dot1p-lp] import 5 export 4
[Router-maptbl-dot1p-lp] quit
```

Configuring traffic policing, traffic shaping, and rate limit

Overview

Traffic policing traffic shaping, and rate limit are QoS techniques that help assign network resources, such as bandwidth. They increase network performance and user satisfaction. For example, you can configure a flow to use only the resources committed to it in a certain time range. This avoids network congestion caused by burst traffic.

Traffic policing generic traffic shaping (GTS), and rate limit control limit the traffic rate and resource usage according to traffic specifications. Once a particular flow exceeds its specifications, such as assigned bandwidth, the flow is shaped or policed to make sure that it is under the specifications. You can use token buckets for evaluating traffic specifications.

Traffic evaluation and token buckets

Token bucket features

A token bucket is analogous to a container that holds a certain number of tokens. Each token represents a certain forwarding capacity. The system puts tokens into the bucket at a constant rate. When the token bucket is full, the extra tokens cause the token bucket to overflow.

Evaluating traffic with the token bucket

A token bucket mechanism evaluates traffic by looking at the number of tokens in the bucket. If the number of tokens in the bucket is enough for forwarding the packets, the traffic conforms to the specification, and is called "conforming traffic." Otherwise, the traffic does not conform to the specification, and is called "excess traffic."

A token bucket has the following configurable parameters:

- **Mean rate at which tokens are put into the bucket**—The permitted average rate of traffic. It is usually set to the committed information rate (CIR).
- **Burst size or the capacity of the token bucket**—The maximum traffic size permitted in each burst. It is usually set to the committed burst size (CBS). The set burst size must be greater than the maximum packet size.

Each arriving packet is evaluated. In each evaluation, if the number of tokens in the bucket is enough, the traffic conforms to the specification and the tokens for forwarding the packet are taken away. If the number of tokens in the bucket is not enough, the traffic is excessive.

Complicated evaluation

You can set two token buckets, bucket C and bucket E, to evaluate traffic in a more complicated environment and achieve more policing flexibility. For example, traffic policing uses the following parameters:

- **CIR**—Rate at which tokens are put into bucket C. It sets the average packet transmission or forwarding rate allowed by bucket C.
- **CBS**—Size of bucket C, which specifies the transient burst of traffic that bucket C can forward.
- **Excess burst size (EBS)**—Size of bucket E, which specifies the transient burst of traffic that bucket E can forward.

Figure 8 Two-bucket structure

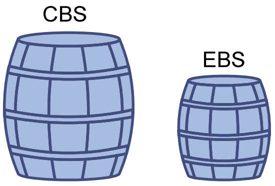


Figure 8 shows the two-bucket structure. CBS is implemented with bucket C, and EBS with bucket E. In each evaluation, packets are measured against the following bucket scenarios:

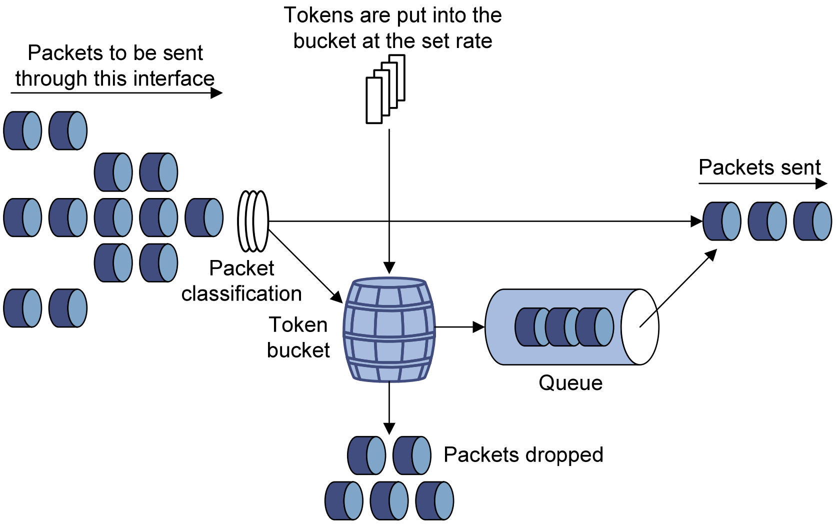
- If bucket C has enough tokens, packets are colored green.
- If bucket C does not have enough tokens but bucket E has enough tokens, packets are colored yellow.
- If neither bucket C nor bucket E has sufficient tokens, packets are colored red.

Traffic policing

Traffic policing supports policing the inbound traffic and the outbound traffic.

A typical application of traffic policing is to supervise the specification of certain traffic entering a network and limit it within a reasonable range, or to "discipline" the extra traffic to prevent aggressive use of network resources by a certain application. For example, you can limit bandwidth for HTTP packets to less than 50% of the total. If the traffic of a certain session exceeds the limit, traffic policing can drop the packets or reset the IP precedence of the packets. Figure 9 shows an example of policing outbound traffic on an interface.

Figure 9 Traffic policing


Tokens are put into the bucket at the set rate
Packets to be sent through this interface
Packets sent
Packet classification
Token bucket
Queue
Packets dropped

Traffic policing is widely used in policing traffic entering the networks of ISPs. It can classify the policed traffic and take predefined policing actions on each packet depending on the evaluation result:

- Forwarding the packet if the evaluation result is "conforming."
- Dropping the packet if the evaluation result is "excess."
- Forwarding the packet with its IP precedence re-marked if the evaluation result is "conforming."
- Delivering the packet to next-level traffic policing with its IP precedence re-marked if the evaluation result is "conforming."

- Entering the next-level policing (you can set multiple traffic policing levels each focused on specific objects).

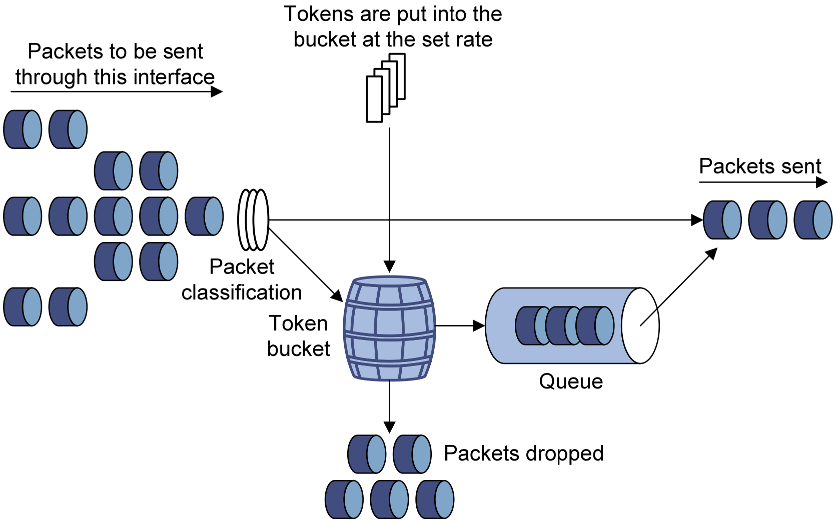
Traffic shaping

Traffic shaping supports shaping the inbound traffic and the outbound traffic.

Traffic shaping limits the outbound traffic rate by buffering exceeding traffic. You can use traffic shaping to adapt the traffic output rate on a device to the input traffic rate of its connected device to avoid packet loss.

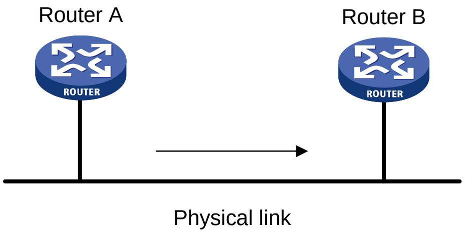
The difference between traffic policing and GTS is that packets to be dropped with traffic policing are retained in a buffer or queue with GTS, as shown in Figure 10. When enough tokens are in the token bucket, the buffered packets are sent at an even rate. Traffic shaping can result in additional delay and traffic policing does not.

Figure 10 GTS


Tokens are put into the bucket at the set rate
Packets to be sent through this interface
Packets sent
Packet classification
Token bucket
Queue
Packets dropped

For example, in Figure 11, Router B performs traffic policing on packets from Router A and drops packets exceeding the limit. To avoid packet loss, you can perform traffic shaping on the outgoing interface of Router A so packets exceeding the limit are cached in Router A. Once resources are released, traffic shaping takes out the cached packets and sends them out.

Figure 11 GTS application



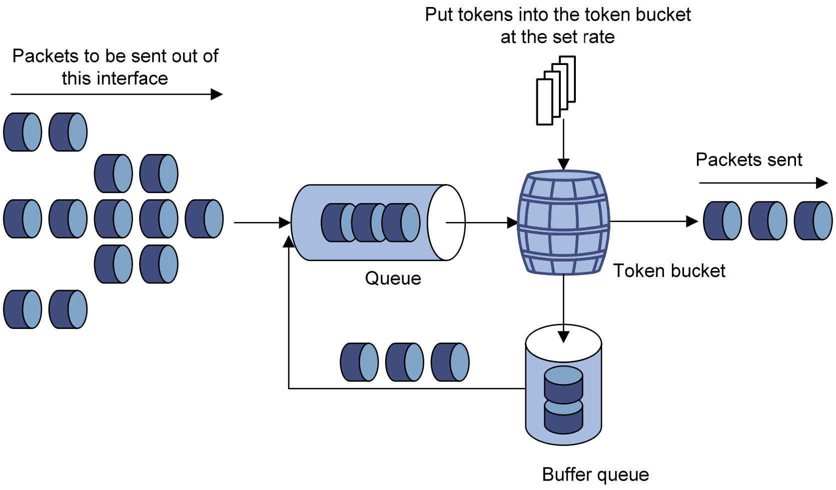
Rate limit

Rate limit supports controlling the rate of inbound traffic and outbound traffic.

The rate limit of a physical interface specifies the maximum rate for forwarding packets (including critical packets).

Rate limit also uses token buckets for traffic control. With rate limit configured on an interface, all packets to be sent through the interface are handled by the token bucket for rate limiting. If enough tokens are in the token bucket, packets can be forwarded. Otherwise, packets are put into QoS queues for congestion management. In this way, the traffic passing the physical interface is controlled.

Figure 12 Rate limit implementation


Put tokens into the token bucket at the set rate
Packets to be sent out of this interface
Packets sent
Queue
Token bucket
Buffer queue

The token bucket mechanism limits traffic rate when accommodating bursts. It allows bursty traffic to be transmitted if enough tokens are available. If tokens are scarce, packets cannot be transmitted until efficient tokens are generated in the token bucket. It restricts the traffic rate to the rate for generating tokens.

Rate limit controls the total rate of all packets on a physical interface. It is easier to use than traffic policing in controlling the total traffic rate on a physical interface.

Configuration task list

Task	Remarks	
Configuring traffic policing	Configuring traffic policing by using the policy approach	
	Configuring traffic policing by using the non-policy approach	Configuring CAR-list-based traffic policing
		Configuring ACL-based traffic policing
		Configuring traffic policing for all traffic
Configuring GTS	Configuring GTS by using the policy approach	
	Configuring GTS by using the non-policy approach	Configuring ACL-based GTS
		Configuring queue-based GTS
		Configuring GTS for all traffic
Configuring the rate limit		

Configuring traffic policing

Configure traffic policing in either policy approach or non-policy approach.

If traffic policing is configured in both the policy approach and non-policy approach, the configuration in policy approach takes effect.

Configuring traffic policing by using the policy approach

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Configure a traffic policing action.	**car cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**pir** *peak-information-rate*] [**green** *action*] [**red** *action*] [**yellow** *action*]	The **pir** *peak-information-rate* and **yellow** *action* options are supported only on SAP modules operating in bridge mode. For a QoS policy implemented in hardware, if you set the CIR or PIR to a value that is not an integral multiple of 64, the system automatically converts the value into the nearest integral multiple of 64 that is greater than the value. For example, if you set 127, 128 takes effect.
7. Return to system view.	**quit**	N/A
8. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
9. Associate the class with the traffic behavior in the QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A
10. Return to system view.	**quit**	N/A
11. Apply the QoS policy.	• Applying the QoS policy to an interface or PVC • Applying the QoS policy to a VLAN • Applying the QoS policy to online users	Choose one of the application destinations as needed.

Configuring traffic policing by using the non-policy approach

Configuring CAR-list-based traffic policing

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure a CAR list.	**qos carl** *carl-index* { **precedence** *precedence-value* | **mac** *mac-address* | **mpls-exp** *mpls-exp-value* | **dscp** *dscp-list* | { **destination-ip-address** | **source-ip-address** } { **subnet** *ip-address mask-length* | **range** *start-ip-address* **to** *end-ip-address* } [**per-address** [**shared-bandwidth**]] }	Configure rules on the CAR list.
3. Enter interface view or port group view.	• Enter interface view: **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*	Use one of the commands. Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
4. Configure a CAR list based CAR policy on the interface or port group.	**qos car** { **inbound** | **outbound** } **carl** *carl-index* **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action*] [**red** *action*]	N/A

Configuring ACL-based traffic policing

Step	Command
1. Enter system view.	**system-view**
2. Configure an ACL.	See "Configuring ACLs."
3. Enter interface view.	**interface** *interface-type interface-number*
4. Configure an ACL-based CAR policy on the interface.	**qos car** { **inbound** | **outbound** } **acl** [**ipv6**] *acl-number* **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**pir** *peak-information-rate*] [**green** *action*] [**red** *action*]

Configuring traffic policing for all traffic

Step	Command
1. Enter system view.	**system-view**
2. Enter interface view or port group view.	• Enter interface view: **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*
3. Configure a CAR action for all traffic on the interface or port group.	**qos car** { **inbound** | **outbound** } **any cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action*] [**red** *action*]

Configuring GTS

GTS for software forwarding does not support IPv6.

Do not configure GTS on a main interface and its subinterfaces at the same time.

Configuring GTS by using the policy approach

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Configure a GTS action.	• In absolute value: **gts cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size* [**queue-length** *queue-length*]]] • In percentage: **gts percent cir** *cir-percent* [**cbs** *cbs-time* [**ebs** *ebs-time*]]	N/A
7. Return to system view.	**quit**	N/A
8. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
9. Associate the class with the traffic behavior in the QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A
10. Return to system view.	**quit**	N/A
11. Apply the QoS policy.	• Applying the QoS policy to an interface or PVC • Applying the QoS policy to a VLAN	Choose one of the application destinations as needed.

Configuring GTS by using the non-policy approach

When you configure GTS in non-policy approach, you can configure the following types of GTS:

- **ACL-based GTS**—Sets GTS parameters for the traffic matching the specific ACL. By specifying multiple ACLs, you can set GTS parameters for different classes of traffic.
- **GTS for all traffic**—Configures GTS parameters for all traffic.

Configuring ACL-based GTS

Step	Command
1. Enter system view.	**system-view**
2. Defining an ACL.	See "Configuring ACLs."
3. Enter interface view.	**interface** *interface-type interface-number*

4. Configure ACL-based GTS on the interface.	**qos gts acl** *acl-number* **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*] [**queue-length** *queue-length*]]

Configuring queue-based GTS

This feature is supported only on SAP modules operating in bridge mode.

To configure queue-based GTS:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or port group view.	• Enter interface view: **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*	Use one of the commands. Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
3. Configure GTS for a queue.	**qos gts queue** *queue-number* **cir** *committed-information-rate* [**cbs** *committed-burst-size*] [**ebs** *excess-burst-size* [**queue-length** *queue-length*]]	N/A

Configuring GTS for all traffic

Step	Command
1. Enter system view.	**system-view**
2. Enter interface view.	**interface** *interface-type interface-number*
3. Configure GTS on the interface.	**qos gts any cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*] [**queue-length** *queue-length*]]

Configuring the rate limit

The rate limit of a physical interface specifies the maximum rate of outgoing packets.

To configure the rate limit:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	Use one of the commands. Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
3. Configure the rate limit for the interface or port group.	**qos lr inbound cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]]	N/A

Configuring packet resequencing

When the network traffic is out of sequence, some systems that cannot resequence packets, such as a video conferencing terminal, might encounter mosaic. The packet resequencing function can alleviate the problem.

To configure packet resequencing:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Enable packet resequencing.	**qos resequencing**	By default, packet resequencing is disabled. After you enable or disable packet resequencing, to make the configuration take effect, you must re-enable the interface by using the **shutdown** and **undo shutdown** command sequence.

Displaying and maintaining traffic policing, GTS, and rate limit

Task	Command	Remarks
Display CAR list information.	**display qos carl** [*carl-index*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display the CAR information on the specified interface.	**display qos car interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display interface GTS configuration information.	**display qos gts interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display interface rate limit configuration information.	**display qos lr interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Traffic policing and GTS configuration examples

Traffic policing and GTS configuration example

Network requirements

As shown in Figure 13:

- Server, Host A, and Host B can access the Internet through Router A and Router B.
- Server, Host A, and GigabitEthernet 1/0/1 of Router A are in the same network segment.
- Host B and GigabitEthernet 1/0/2 of Router A are in the same network segment.

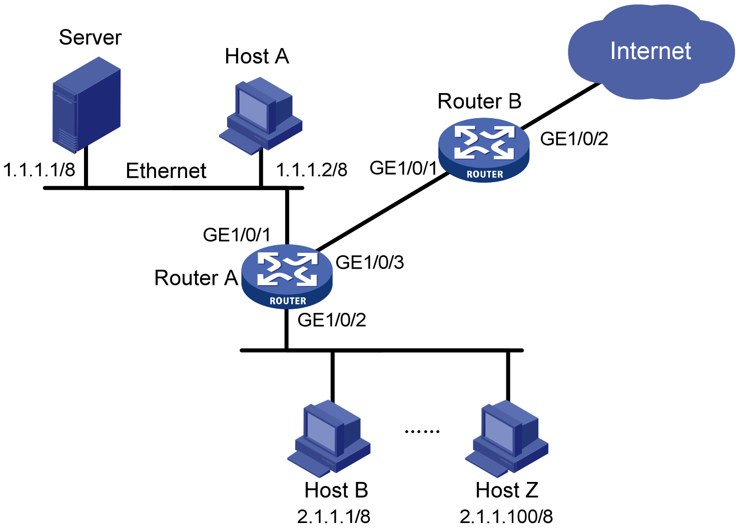
Perform traffic control for packets received on GigabitEthernet 1/0/1 of Router A from Server and Host A, respectively, as follows:

- Limit the rate of packets from Server to 54 kbps. When the traffic rate is below 54 kbps, the traffic is forwarded. When the traffic rate exceeds 54 kbps, the excess packets are marked with IP precedence 0 and then forwarded.
- Limit the rate of packets from Host A to 8 kbps. When the traffic rate is below 8 kbps, the traffic is forwarded. When the traffic rate exceeds 8 kbps, the excess packets are dropped.

Traffic control for packets forwarded by GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2 of Router B is as follows:

- Limit the receiving rate on GigabitEthernet 1/0/1 of Router B to 500 kbps, and the excess packets are dropped.
- Limit the sending rate on GigabitEthernet 1/0/2 of Router B to 1000 kbps, and the excess packets are dropped.

Figure 13 Network diagram



Configuration procedure

1. Configure Router A:

   # Configure GTS on GigabitEthernet 1/0/3, shaping the packets when the sending rate exceeds 500 kbps to decrease the packet loss rate of GigabitEthernet 1/0/1 of Router B.

```
<RouterA> system-view
[RouterA] interface gigabitethernet 1/0/3
[RouterA-GigabitEthernet1/0/3] qos gts any cir 500
[RouterA-GigabitEthernet1/0/3] quit
```

   # Configure ACLs to permit the packets from Server and Host A.

```
[RouterA] acl number 2001
[RouterA-acl-basic-2001] rule permit source 1.1.1.1 0
[RouterA-acl-basic-2001] quit
[RouterA] acl number 2002
[RouterA-acl-basic-2002] rule permit source 1.1.1.2 0
[RouterA-acl-basic-2002] quit
```

   # Configure CAR policies for different flows received on GigabitEthernet 1/0/1.

```
[RouterA] interface gigabitethernet 1/0/1
[RouterA-GigabitEthernet1/0/1] qos car inbound acl 2001 cir 54 cbs 4000 ebs 0 green
pass red remark-prec-pass 0
```

```
[RouterA-GigabitEthernet1/0/1] qos car inbound acl 2002 cir 8 cbs 1875 ebs 0 green
pass red discard
[RouterA-GigabitEthernet1/0/1] quit
```

2. Configure Router B:

   # Configure a CAR policy on GigabitEthernet 1/0/1 to limit the incoming traffic rate to 500 kbps and drop the excess packets.

   ```
   <RouterB> system-view
   [RouterB] interface gigabitethernet 1/0/1
   [RouterB-GigabitEthernet1/0/1] qos car inbound any cir 500 cbs 32000 ebs 0 green pass
   red discard
   [RouterB-GigabitEthernet1/0/1] quit
   ```

   # Configure a CAR policy on GigabitEthernet 1/0/2 to limit the sending rate to 1 Mbps and drop the excess packets.

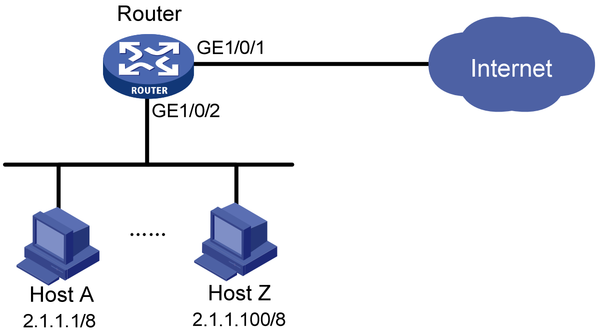
   ```
   [RouterB] interface gigabitethernet 1/0/2
   [RouterB-GigabitEthernet1/0/2] qos car outbound any cir 1000 cbs 65000 ebs 0 green
   pass red discard
   ```

IP rate limiting configuration example

Network requirements

As shown in Figure 14, limit the rate of packets entering GigabitEthernet 1/0/2 of the Router as follows: perform per-IP-address rate limiting for traffic sourced from Host A through Host Z, which are on the network segment 2.1.1.1 through 2.1.1.100, with the per-IP-address rate limit being 500 bps, and make traffic from all IP addresses on the network segment share the remaining bandwidth.

Figure 14 Network diagram



Configuration procedure

# Configure per-IP-address rate limiting on GigabitEthernet 1/0/2 to limit the rate of each PC on the network segment 2.1.1.1 through 2.1.1.100, and make traffic from all IP addresses in the network segment share the remaining bandwidth.

```
<Router> system-view
[Router] qos carl 1 source-ip-address range 2.1.1.1 to 2.1.1.100 per-address
shared-bandwidth
[Router] interface gigabitethernet 1/0/2
[Router-GigabitEthernet1/0/2] qos car inbound carl 1 cir 500 cbs 1875 ebs 0 green pass
red discard
[Router-GigabitEthernet1/0/2] quit
```

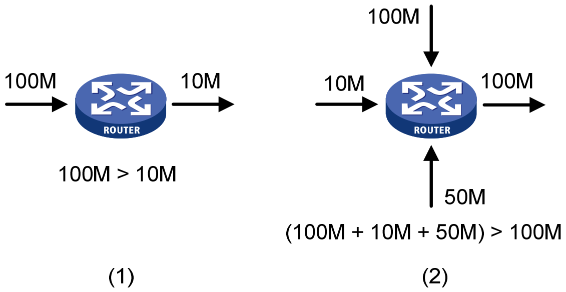
Configuring congestion management

Overview

Causes, impacts, and countermeasures of congestion

Congestion occurs on a link or node when traffic size exceeds the processing capability of the link or node. It is typical of a statistical multiplexing network and can be caused by link failures, insufficient resources, and various other causes. Figure 15 shows some common congestion scenarios.

Figure 15 Traffic congestion causes



Congestion can bring the following negative results:

- Increased delay and jitter during packet transmission.
- Decreased network throughput and resource use efficiency.
- Network resource (memory, in particular) exhaustion and system breakdown.

Congestion is unavoidable in switched networks or multiuser application environments. To improve the service performance of your network, take measures to manage and control it.

One major issue that congestion management deals with is defining a resource dispatching policy to prioritize packets for forwarding when congestion occurs.

Congestion management policies

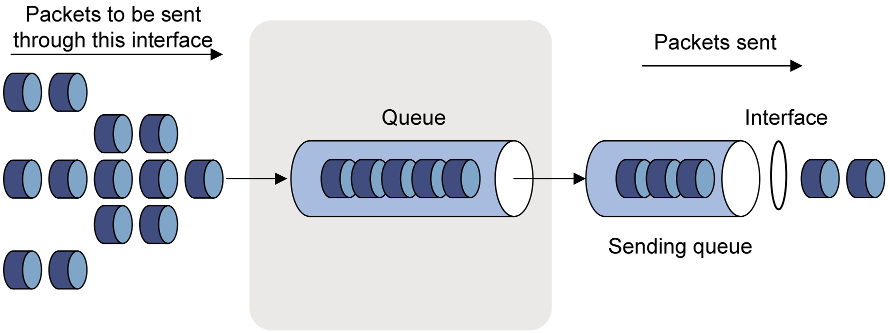
Queuing is a common congestion management technique. It classifies traffic into queues and picks out packets from each queue by using a certain algorithm. Various queuing algorithms are available, and each addresses a particular network traffic problem. Your choice of algorithm significantly affects bandwidth assignment, delay, and jitter.

Queue scheduling treats packets with different priorities differently to transmit high-priority packets preferentially.

This section briefly describes several common queue-scheduling mechanisms.

FIFO

Figure 16 FIFO queuing



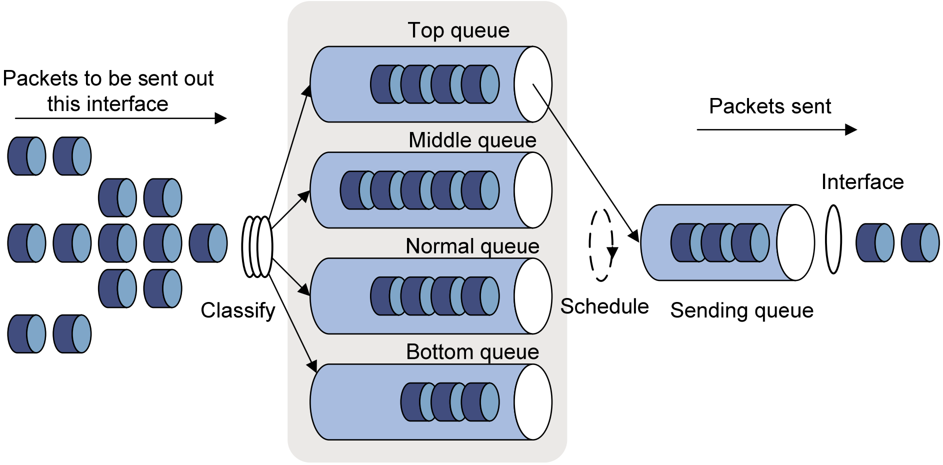
As shown in Figure 16, the first in first out (FIFO) uses a single queue and does not classify traffic or schedule queues. FIFO delivers packets depending on their arrival order, with the one arriving earlier scheduled first. The only concern of FIFO is queue length, which affects delay and packet loss rate. On a device, resources are assigned for packets depending on their arrival order and load status of the device. The best-effort service model uses FIFO queuing.

FIFO does not address congestion problems. If only one FIFO output/input queue exists on a port, you can hardly ensure timely delivery of mission-critical or delay-sensitive traffic or smooth traffic jitter. The situation is worsened if malicious traffic is present to occupy bandwidth aggressively. To control congestion and prioritize forwarding of critical traffic, use other queue scheduling mechanisms, where multiple queues can be configured. Within each queue, however, FIFO is still used.

By default, FIFO queuing is used on interfaces.

PQ

Figure 17 Priority queuing (PQ)



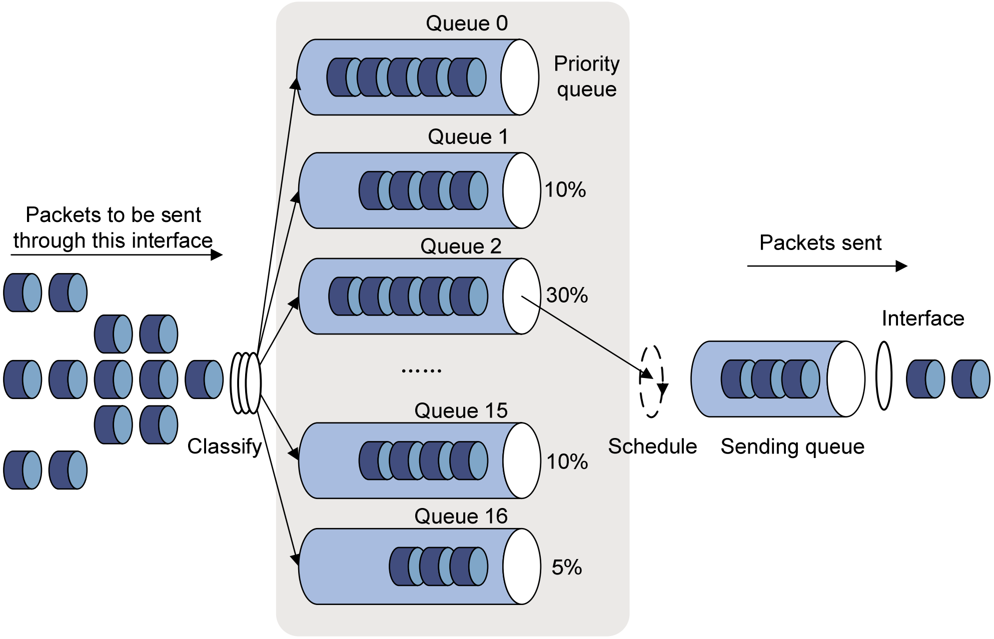
Priority queuing is designed for mission-critical applications. The key feature of mission-critical applications is they require preferential service to reduce the response delay when congestion occurs. Priority queuing can flexibly determine the order of forwarding packets by network protocol (for example, IP and IPX), incoming interface, packet length, source/destination address, and so on. Priority queuing classifies packets into four queues: top, middle, normal, and bottom, in descending priority order. By default, packets are assigned to the normal queue. Each of the four queues is a FIFO queue.

Priority queuing schedules the four queues in the descending order of priority. It sends packets in the queue with the highest priority first. When the queue with the highest priority is empty, it sends packets in the queue with the second highest priority. In this way, you can assign the mission-critical packets to the high priority queue to make sure that they are always served first. The common service packets are assigned to the low priority queues and transmitted when the high priority queues are empty.

The disadvantage of priority queuing is that packets in the lower priority queues cannot be transmitted if packets exist in the higher queues for a long time when congestion occurs.

CQ

Figure 18 Custom queuing (CQ)



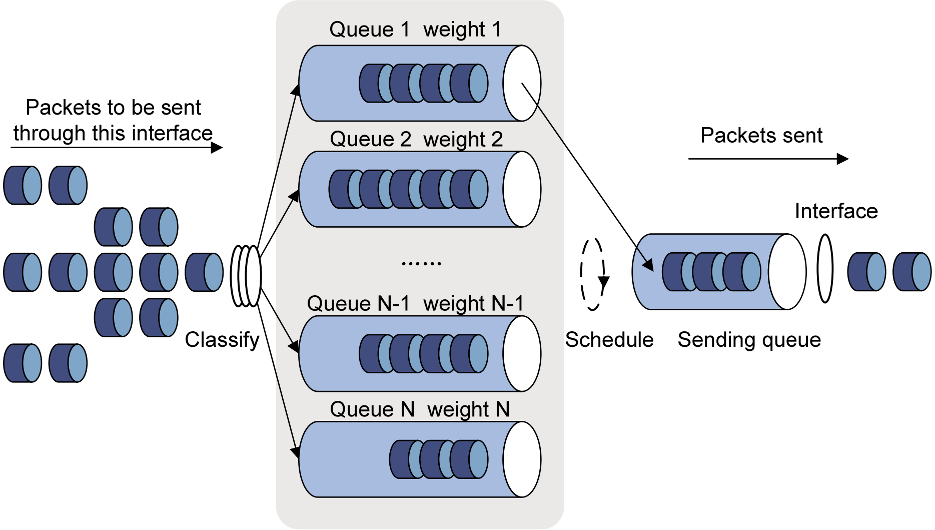
CQ provides 17 queues, numbered from 0 to 16. Queue 0 is a reserved system queue, and queues 1 through 16 are customer queues, as shown in Figure 18. You can define traffic classification rules and assign a percentage of interface/PVC bandwidth for each customer queue. By default, packets are assigned to queue 1.

During a cycle of queue scheduling, CQ first empties the system queue. Then, it schedules the 16 queues in a round robin way: it sends a certain number of packets (based on the percentage of interface bandwidth assigned for each queue) out of each queue in the ascending order of queue 1 to queue 16. CQ guarantees normal packets a certain amount of bandwidth, and ensures that mission-critical packets are assigned more bandwidth.

CQ can assign free bandwidth of idle queues to busy queues. Even though it performs round robin queue scheduling, CQ does no assign fixed time slots for the queues. If a queue is empty, CQ immediately moves to the next queue. When a class does not have packets, the bandwidth for other classes increases.

WFQ

Figure 19 Weighted fair queuing (WFQ)



Before WFQ is introduced, make sure that you have understood fair queuing (FQ). FQ is designed for fairly allocating network resources to reduce delay and jitter of each traffic flow as possible. In an attempt to balance the interests of all parties, FQ follows these principles:

- Different queues have fair dispatching opportunities for delay balancing among streams.
- Short packets and long packets are fairly scheduled: if long packets and short packets exist in queues, statistically the short packets must be scheduled preferentially to reduce the jitter between packets on the whole.

Compared with FQ, WFQ takes weights into account when determining the queue scheduling order. Statistically, WFQ gives high-priority traffic more scheduling opportunities than low-priority traffic. WFQ can automatically classify traffic according to the "session" information of traffic (protocol type, TCP or UDP source/destination port numbers, source/destination IP addresses, IP precedence bits in the ToS field, and so on), and try to provide as many queues as possible so that each traffic flow can be put into these queues to balance the delay of every traffic flow on a whole. When dequeuing packets, WFQ assigns the outgoing interface bandwidth to each traffic flow by precedence. The higher precedence value a traffic flow has, the more bandwidth it gets.

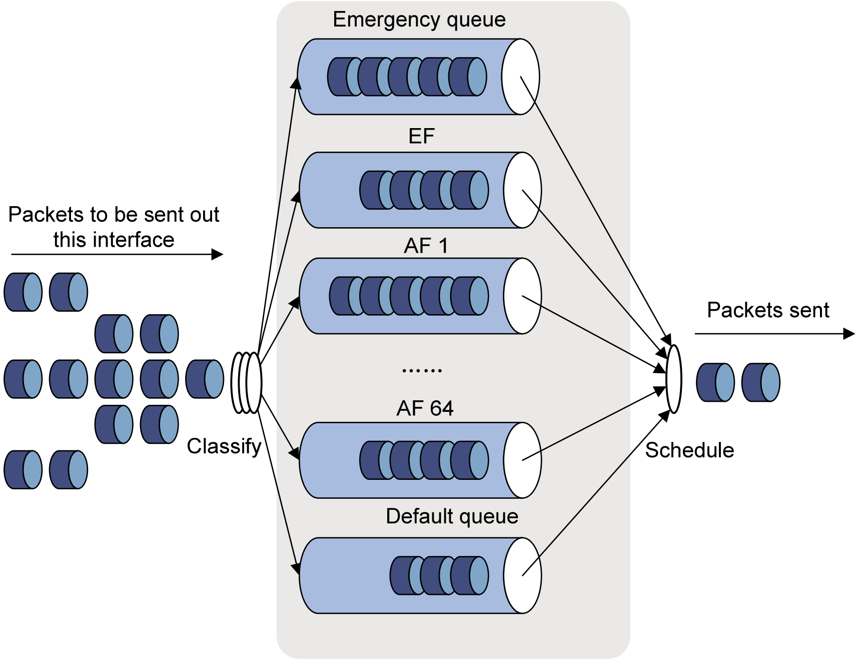
For example, assume five flows exist in the current interface with precedence 0, 1, 2, 3, and 4, respectively. The total bandwidth quota is the sum of all the (precedence value + 1)s, 1 + 2 + 3 + 4 + 5 = 15.

The bandwidth percentage assigned to each flow is (precedence value of the flow + 1)/total bandwidth quota. The bandwidth percentages for flows are 1/15, 2/15, 3/15, 4/15, and 5/15, respectively.

Because WFQ can balance the delay and jitter of each flow when congestion occurs, it is suitable for handling some special occasions. For example, WFQ is used in the assured forwarding (AF) services of the RSVP. In GTS, WFQ schedules buffered packets.

CBQ

Figure 20 CBQ



Class-based queuing (CBQ) extends WFQ by supporting user-defined classes. When network congestion occurs, CBQ uses user-defined traffic match criteria to enqueue packets. Before that, congestion avoidance actions, such as tail drop or WRED and bandwidth restriction check, are performed before packets are enqueued. When being dequeued, packets are scheduled by WFQ.

CBQ provides the following queues:

- **Emergency queue**—Enqueues emergent packets. The emergency queue is a FIFO queue without bandwidth restriction.
- **Low Latency Queuing (LLQ)**—An EF queue. Because packets are fairly treated in CBQ, delay-sensitive flows like video and voice packets might not be transmitted timely. To solve this problem, an EF queue was introduced to preferentially transmit delay-sensitive flows. LLQ combines PQ and CBQ to preferentially transmit delay-sensitive flows like voice packets. When defining traffic classes for LLQ, you can configure a class of packets to be preferentially transmitted. Such a class is called a "priority class." The packets of all priority classes are assigned to the same priority queue. Bandwidth restriction on each class of packets is checked before the packets are enqueued. During the dequeuing operation, packets in the priority queue are transmitted first. Packets in other queues are scheduled by using WFQ. To reduce the delay of the other queues except the priority queue, LLQ assigns the maximum available bandwidth for each priority class. The bandwidth value polices traffic during congestion. When no congestion is present, a priority class can use more than the bandwidth assigned to it. During congestion, the packets of each priority class exceeding the assigned bandwidth are discarded.
- **Bandwidth queuing (BQ)**—An AF queue. The BQ provides strict, exact, guaranteed bandwidth for AF traffic, and schedules the AF classes proportionally. The system supports up to 64 AF queues.
- **Default queue**—A WFQ queue. It transmits the BE traffic by using the remaining interface bandwidth.

The system matches packets with classification rules in the following order:

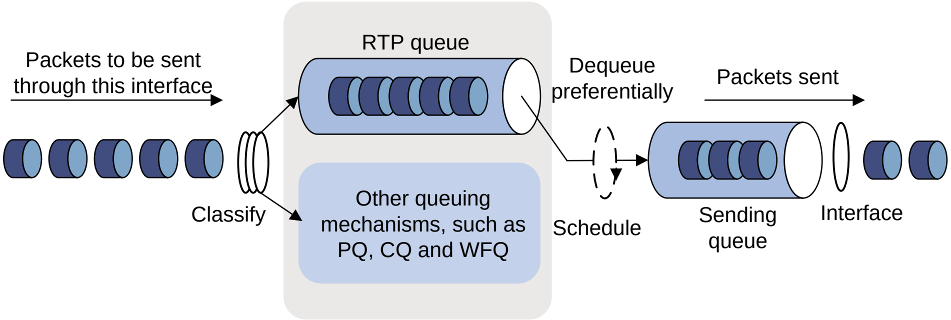
- Match packets with priority classes and then the other classes.
- Match packets with priority classes in the configuration order.
- Match packets with other classes in the configuration order.

- Match packets with classification rules in a class in the configuration order.

RTP priority queuing

Real-time transport protocol (RTP) priority queuing is a simple queuing technique designed to guarantee QoS for real-time services (including voice and video services). It assigns RTP voice or video packets to high-priority queues for preferential sending, minimizing delay and jitter and ensuring QoS for voice or video services sensitive to delay.

Figure 21 RTP queuing


Packets to be sent through this interface
RTP queue
Dequeue preferentially
Packets sent
Classify
Other queuing mechanisms, such as PQ, CQ and WFQ
Schedule
Sending queue
Interface

As shown in Figure 21, RTP priority queuing assigns RTP packets to a high-priority queue. An RTP packet is a UDP packet with an even destination port number in a configurable range. RTP priority queuing can be used in conjunction with any queuing (such as, FIFO, PQ, CQ, WFQ, and CBQ), and it always has the highest priority.

Do not use RTP priority queuing in conjunction with CBQ. LLQ of CBQ can also guarantee real-time service data transmission.

Congestion management technique comparison

Breaking through the single congestion management policy of FIFO for traditional IP devices, the device provides all the congestion management techniques described above to offer powerful QoS capabilities, meeting different QoS requirements of different applications.

Table 4 Congestion management technique comparison

Type	Number of queues	Advantages	Disadvantages
FIFO	1	• No need to configure, easy to use. • Easy to operate, low delay.	• All packets are treated equally. The available bandwidth, delay and drop probability are determined by the arrival order of packets. • No restriction on traffic from connectionless protocols (protocols without any flow control mechanism, UDP, for example), resulting in bandwidth loss for traffic of connection-oriented protocols (TCP, for example). • No delay guarantee for time-sensitive real-time applications, such as VoIP.

Type	Number of queues	Advantages	Disadvantages
PQ	4	Absolute bandwidth and delay guarantees for real-time and mission-critical applications, such as VoIP.	• Need to configure, low processing speed. • If no restriction is imposed on bandwidth assigned to high-priority packets, low-priority packets might fail to get bandwidth.
CQ	16	• Bandwidth assignment in percentages for different applications. • Bandwidth reassignment to increase bandwidth for each class when packets of certain classes are not present.	Need to configure, low processing speed.
WFQ	Configurable	• Easy to configure. • Bandwidth guarantee for packets from cooperative (interactive) sources (such as TCP packets). • Reduced jitter. • Reduced delay for interactive applications with a small amount of data. • Bandwidth assignment based on traffic priority. • Automatic bandwidth reassignment to increase bandwidth for each class when the number of traffic classes decreases.	The processing speed is faster than PQ and CQ but slower than FIFO.

Type	Number of queues	Advantages	Disadvantages
CBQ	Configurable (0 to 64)	• Flexible traffic classification based on various rules and differentiated queue scheduling mechanisms for EF, AF and BE services. • Highly precise bandwidth guarantee and queue scheduling on the basis of AF service weights for various AF services. • Absolutely preferential queue scheduling for the EF service to meet the delay requirement of real-time data. • Overcomes the disadvantage of PQ that some low-priority queues are not serviced by restricting the high-priority traffic. • WFQ scheduling for best-effort traffic (the default class).	The system overheads are large.

If the burst traffic is too heavy, increase the queue length to make queue scheduling more accurate.

Configuring the FIFO queue size

This feature is not supported on SAP modules operating in bridge mode.

FIFO is the default queue scheduling mechanism for an interface or PVC, and the FIFO queue size is configurable.

To configure the FIFO queue size:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or PVC view.	• Enter interface view: **interface** *interface-type interface-number* • Enter PVC view: **a. interface atm** *interface-number* **b. pvc** *vpi/vci*	N/A
3. Configure the FIFO queue size.	**qos fifo queue-length** *queue-length*	By default: • The FIFO queue length is 75 for tunnel interfaces, aggregate interfaces, and HDLC link bundle interfaces. • The FIFO queue length is 1024 for other interfaces.

You must enable the rate limit function for the queuing function to take effect on these interfaces: tunnel interfaces, subinterfaces, Layer 3 aggregate interfaces, HDLC link bundle interfaces, RPR logical interfaces, and VT interfaces configured with PPPoE, PPPoA, or PPPoEoA.

Configuration example

# Set the FIFO queue size to 100.

```
<Sysname> system-view
[Sysname] interface gigabitethernet1/0/1
[Sysname-GigabitEthernet1/0/1] qos fifo queue-length 100
```

Configuring PQ

This feature is not supported on SAP modules operating in bridge mode.

You can define multiple rules for a priority queue list (PQL) and apply the list to an interface. When a packet arrives at the interface or PVC, the system matches the packet with each rule in the order configured. If a match is found, the packet is assigned to the queue and the match procedure is complete. If the packet cannot match any rule, the packet is assigned to the default queue **normal**.

Configuration restrictions and guidelines

- PQ applies to all physical interfaces except interfaces using the X.25 or LAPB protocol at the data link layer.
- You must enable the rate limit function for the queuing function to take effect on these interfaces: tunnel interfaces, subinterfaces, Layer 3 aggregate interfaces, HDLC link bundle interfaces, RPR logical interfaces, and VT interfaces configured with PPPoE, PPPoA, or PPPoEoA.

Configuration procedure

You can configure PQ by applying a PQ list to an interface. For an interface, the latest applied PQ list overwrites the previous one.

To configure PQ:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure a PQ list.	**qos pql** *pql-index* **protocol ip** [*queue-key key-value*] **queue** { **bottom** | **middle** | **normal** | **top** }	N/A
3. Specify the default queue for the PQ list.	**qos pql** *pql-index* **default-queue** { **bottom** | **middle** | **normal** | **top** }	Optional. This command specifies the queue to which unmatched packets are assigned.
4. Set the queue size.	**qos pql** *pql-index* **queue** { **bottom** | **middle** | **normal** | **top** } **queue-length** *queue-length*	Optional.
5. Enter interface view.	**interface** *interface-type interface-number*	N/A
6. Apply the PQ list to the interface.	**qos pq pql** *pql-index*	By default, FIFO applies.

7. Display PQ list configuration information.	**display qos pq interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.
8. Display the contents of the specific PQ list or all the PQ lists.	**display qos pql** [*pql-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

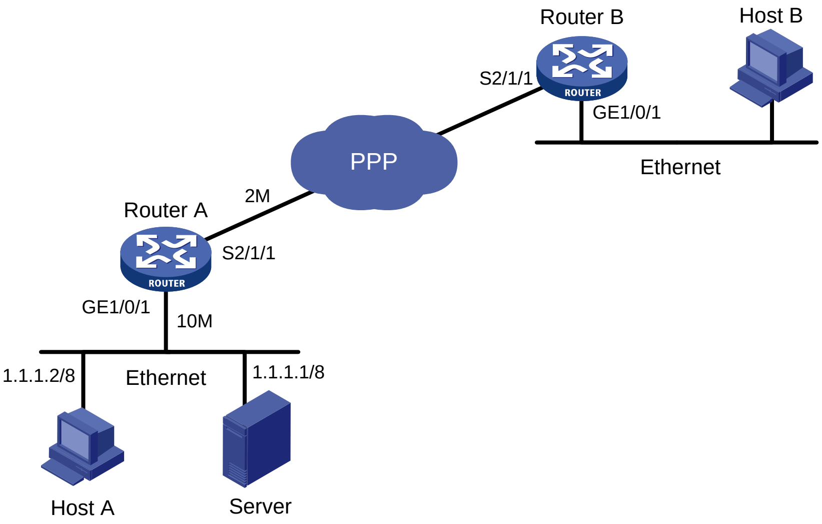
PQ configuration example

Network requirements

As shown in Figure 22, both Server and Host A send data to Host B through Router A. Suppose Server sends critical packets and Host A sends non-critical packets. Congestion might occur on Serial 2/1/1 and result in packet loss because the rate of the incoming interface GigabitEthernet 1/0/1 is greater than that of the outgoing interface Serial 2/1/1 on Router A.

Configure PQ, so that the critical packets from Server are transmitted preferentially when congestion occurs in the network.

Figure 22 Network diagram



Configuration procedure

Configure Router A:

# Configure ACLs to match the packets from Server and Host A, respectively.

```
[RouterA] acl number 2001
[RouterA-acl-basic-2001] rule permit source 1.1.1.1 0.0.0.0
[RouterA] acl number 2002
[RouterA-acl-basic-2002] rule permit source 1.1.1.2 0.0.0.0
```

# Configure a PQ list that assigns the packets from Server to the top queue and those from Host A to the bottom queue when congestion occurs. Set the maximum queue size of the top queue to 50 and that of the bottom queue to 100 in the PQ list.

```
[RouterA] qos pql 1 protocol ip acl 2001 queue top
[RouterA] qos pql 1 protocol ip acl 2002 queue bottom
[RouterA] qos pql 1 queue top queue-length 50
[RouterA] qos pql 1 queue bottom queue-length 100
```

# Apply PQ list 1 to Serial 2/1/1.

```
[RouterA] interface serial 2/1/1
[RouterA-Serial2/1/1] qos pq pql 1
```

Configuring CQ

This feature is not supported on SAP modules operating in bridge mode.

You can configure a CQ list that contains up to 16 queues (1-16), with each queue including the match criteria for packets to enter the queue, the length of the queue, and the bytes sent from the queue during a cycle of round robin queue scheduling. Only one CQ list can be applied to an interface.

You must enable the rate limit function for the queuing function to take effect on these interfaces: tunnel interfaces, subinterfaces, Layer 3 aggregate interfaces, HDLC link bundle interfaces, RPR logical interfaces, and VT interfaces configured with PPPoE, PPPoA, or PPPoEoA.

To configure CQ:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure a CQ list.	**qos cql** *cql-index* **protocol ip** [*queue-key key-value*] **queue** *queue-number*	Optional.
3. Specify the default queue.	**qos cql** *cql-index* **default-queue** *queue-number*	Optional. This command specifies the queue to which unmatched packets are assigned.
4. Set the length of a queue.	**qos cql** *cql-index* **queue** *queue-number* **queue-length** *queue-length*	Optional.
5. Configure the bytes sent from a queue during a cycle of round robin queue scheduling.	**qos cql** *cql-index* **queue** *queue-number* **serving** *byte-count*	Optional.
6. Enter interface view.	**interface** *interface-type interface-number*	N/A
7. Apply the CQ list to the interface.	**qos cq cql** *cql-index*	By default, FIFO applies.
8. Display CQ list configuration on an interface.	**display qos cq interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.
9. Display the configuration of a CQ list.	**display qos cql** [*cql-index*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

CQ configuration example

Network requirements

Configure CQ on interface Serial 2/1/1 to assign packets matching ACL 2000 to queue 1 and specify queue 1 to send 2000 bytes during a cycle of round robin queue scheduling.

Configuration procedure

# Enter system view.

```
<Sysname> system-view
```

# Configure ACL 2000 to match packets sourced from 1.1.1.1 0.0.0.0.

```
[Sysname] acl number 2001
[Sysname-acl-basic-2001] rule permit source 1.1.1.1 0.0.0.0
```

# Configure CQ list 1.

```
[Sysname] qos cql 1 protocol ip acl 2001 queue 1
[Sysname] qos cql 1 queue 1 serving 2000
```

# Apply CQ list 1 to interface Serial 2/1/1.

```
[Sysname] interface serial 2/1/1
[Sysname-Serial 2/1/1] qos cq cql 1
```

Configuring WFQ

You must enable the rate limit function for the queuing function to take effect on these interfaces: tunnel interfaces, subinterfaces, Layer 3 aggregate interfaces, HDLC link bundle interfaces, RPR logical interfaces, and VT interfaces configured with PPPoE, PPPoA, or PPPoEoA.

On an interface without WFQ configured, the **qos wfq** command can enable WFQ and configure WFQ-related parameters. If WFQ is configured for the interface, the **qos wfq** command can modify the WFQ-related parameters.

To configure WFQ:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Configure WFQ.	**qos wfq** [**dscp** | **precedence**] [**queue-length** *max-queue-length* [**queue-number** *total-queue-number*]]	By default, FIFO applies.
4. Display interface WFQ configuration information.	**display qos wfq interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

WFQ configuration example

Network requirements

Configure WFQ on Serial 2/1/1, setting the maximum queue size to 100, and the total number of queues to 512.

Configuration procedure

# Enter system view.

```
<Sysname> system-view
```

# Enter interface view.

```
[Sysname] interface serial 2/1/1
```

# Configure WFQ on Serial 2/1/1, setting the maximum queue size to 100, and the total number of queues to 512.

```
[Sysname-Serial2/1/1] qos wfq queue-length 100 queue-number 512
```

Configuring CBQ

To configured CBQ:

1. Create a class and define a set of traffic match criteria in class view.
2. Create a traffic behavior, and define a group of QoS features in traffic behavior view.
3. Create a policy, and associate a traffic behavior with a class in policy view.
4. Apply the QoS policy in the interface or PVC view.

Predefined classes, traffic behaviors, and policies

The system predefines the following classes, traffic behaviors, and policies:

Predefined classes

The system predefines some classes and defines general rules for them. You can use these predefined classes when defining a policy.

- The default class

  default-class—Matches the default traffic.
- DSCP-based predefined classes

  ef, af1, af2, af3, af4—Matches IP DSCP value ef, af1, af2, af3, af4, respectively.
- IP precedence-based predefined classes

  ip-prec0, ip-prec1, …ip-prec7—Matches IP precedence value 0, 1, …7, respectively.
- MPLS EXP-based predefined classes

  mpls-exp0, mpls-exp1, …mpls-exp7—Matches MPLS EXP value 0, 1, …7, respectively.

Predefined traffic behaviors

The system predefines some traffic behaviors and defines QoS features for them.

- **ef**—Assigns a class of packets to the EF queue and assigns 20% of the available interface/PVC bandwidth to the class of packets.
- **af**—Assigns a class of packets to the AF queue and assigns 20% of the available interface/PVC bandwidth to the class of packets.
- **be**—Defines no features.
- **be-flow-based**—Assigns a class of packets to a WFQ queue and specifies the drop policy as WRED. By default, 256 WFQ queues exist, and WRED is an IP precedence-based drop policy.

Predefined QoS policy

The system predefines a QoS policy, specifies a predefined class for the policy and associates a predefined behavior with the class. The policy is named **default**, with the default CBQ action.

The policy **default** is defined as follows:

- Associates the predefined class **ef** with the predefined traffic behavior **ef**.
- Associates predefined classes **af1** through **af4** with the predefined traffic behavior **af**.
- Associates the predefined class **default-class** with the predefined traffic behavior **be**.

Defining a class

To define a class, create the class first, and then configure match criteria in class view.

To define a class:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	By default, the **and** keyword is used, and the relation between match criteria is logical AND.
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A

Defining a traffic behavior

To define a traffic behavior, create the traffic behavior first and then configure QoS attributes in traffic behavior view.

Configure AF and the minimum guaranteed bandwidth

When you configure AF and the minimum guaranteed bandwidth, follow these guidelines:

- You can apply this traffic behavior only to the outgoing traffic of an interface.
- You cannot configure the **queue ef** command together with the **queue af** command in the same traffic behavior.
- To reference both the **queue ef** command and the **queue af** command in a policy, you must configure them in the same unit (either bandwidth or percentage). If not, your referencing attempts will fail.

To configure AF and the minimum guaranteed bandwidth:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified behavior name cannot be the name of any system-defined behavior.
3. Configure AF and the minimum guaranteed bandwidth.	**queue af bandwidth** { *bandwidth* | **pct** *percentage* }	N/A

Configuring EF and the maximum bandwidth

When you configure EF and the maximum bandwidth, follow these guidelines:

- You cannot configure the **queue ef** command together with the any of the commands **queue af**, **queue-length**, and **wred** for a traffic behavior.
- The default class cannot be associated with a traffic behavior including EF.
- To reference both the **queue ef** command and the **queue af** command in a policy, you must configure them in the same unit (either bandwidth or percentage). If not, your referencing attempts will fail.

To configure EF and the maximum bandwidth:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.

3. Configure EF and the maximum bandwidth.	**queue ef bandwidth** { *bandwidth* [**cbs** *burst*] | **pct** *percentage* [**cbs-ratio** *ratio*] }	N/A

Configuring WFQ

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Configure WFQ.	**queue wfq** [**queue-number** *total-queue-number*]	N/A

You can associate the traffic behavior that contains a WFQ action only with the default class.

Configuring the maximum queue size

Configure the maximum queue size and use tail drop.

When low-priority services preempt the bandwidth for the AF service, you can increase the queue size for the AF service.

To configure the maximum queue size:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Set the maximum queue size.	**queue-length** *queue-length*	N/A

Check that the **queue af** command or the **queue wfq** command has been configured before you configure the **queue-length** command. Executing the **undo queue af** command or the **undo queue wfq** command cancels also the **queue-length** command.

Enabling WRED

When you enable WRED, follow these guidelines:

- Before enabling WRED, configure the **queue af** command or the **queue wfq** command.
- The **wred** command and the **queue-length** command are mutually exclusive.
- When WRED is disabled, other configurations under it are deleted.
- The WRED configuration in QoS policies overrides the WRED configuration directly configured on interfaces.

To enable WRED:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.

3. Enable WRED.	**wred** [**dscp** | **ip-precedence**]	• **dscp**—Uses the DSCP value for calculating the drop probability for a packet. • **ip-precedence**—Uses the IP precedence value for calculating the drop probability for a packet. This keyword is used by default.

Configuring the exponent for WRED to calculate the average queue size

Before configuring the WRED exponent, make sure the **queue af** command or the **queue wfq** command has been configured and the **wred** command has been used to enable WRED.

To configure the exponent for WRED to calculate the average queue size:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Configure the exponent for WRED to calculate the average queue size.	**wred weighting-constant** *exponent*	The default exponent is 9.

Configuring the lower limit, upper limit, and drop probability denominator for each DSCP value in WRED

To perform this configuration, make sure DSCP-based WRED has been enabled with the **wred dscp** command.

Disabling WRED also removes the **wred dscp** command configuration.

Removing the **queue af** or **queue wfq** command configuration also removes the WRED-related parameters.

To configure the lower limit, upper limit, and drop probability denominator for a DSCP value in WRED:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Configure the lower limit, upper limit and drop probability denominator for a DSCP value in WRED.	**wred dscp** *dscp-value* **low-limit** *low-limit* **high-limit** *high-limit* [**discard-probability** *discard-prob*]	*dscp-value*: DSCP value in the range of 0 to 63, which can also be any of the following keywords: **ef**, **af11**, **af12**, **af13**, **af21**, **af22**, **af23**, **af31**, **af32**, **af33**, **af41**, **af42**, **af43**, **cs1**, **cs2**, **cs3**, **cs4**, **cs5**, **cs6**, **cs7**, and **default**.

Configuring the lower limit, upper limit, and drop probability denominator for each IP precedence value in WRED

To perform this configuration, make sure IP precedence-based WRED has been enabled with the **wred ip-precedence** command.

Disabling WRED also removes the **wred ip-precedence** command configuration.

Removing the **queue af** or **queue wfq** command configuration also removes the WRED-related parameters.

To configure the lower limit, upper limit, and drop probability denominator for an IP precedence value in WRED:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Configure the lower limit, upper limit and drop probability denominator for an IP precedence value in WRED.	**wred ip-precedence** *precedence* **low-limit** *low-limit* **high-limit** *high-limit* [**discard-probability** *discard-prob*]	N/A

Defining a QoS policy

You associate a behavior with a class in a QoS policy. Each behavior includes a set of actions, such as queue scheduling (EF, AF, and WFQ), traffic policing, traffic shaping, WRED, and traffic marking.

To associate a traffic behavior with a specific class in policy view:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
3. Associate a traffic behavior with a class in the policy.	**classifier** *classifier-name* **behavior** *behavior-name*	*classifier-name*: Class name. It must be the name of an existing system-defined or user-defined class. *behavior-name*: Name of a behavior. It must be the name of an existing system-defined or user-defined behavior.

Applying the QoS policy

Use the **qos apply policy** command to apply a policy to a physical interface or ATM PVC. You can apply a policy to multiple physical interfaces or ATM PVCs.

Configuration restrictions and guidelines

- You can apply a QoS policy configured with various QoS actions (including remark, car, gts, queue af, queue ef, queue wfq, wred, and so on) to common physical interfaces and the VT interfaces used by MP.
- An inbound QoS policy cannot contain a GTS action or any of these queuing actions: **queue ef**, **queue af**, or **queue wfq**.
- You must enable the rate limit function for the queuing function to take effect on these interfaces: tunnel interfaces, subinterfaces, Layer 3 aggregate interfaces, HDLC link bundle interfaces, RPR logical interfaces, and VT interfaces configured with PPPoE, PPPoA, or PPPoEoA. At the same time, you must configure the **qos max-bandwidth** command to provide base bandwidth for CBQ bandwidth calculation.

- On some cards, QoS policies can be applied but cannot take effect due to limited system resources. In this case, you can adjust related parameters (for example, reducing the number of queues) according to system prompt and then apply a QoS policy again.

Configuration procedure

To apply a policy to an interface or ATM PVC:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or PVC view.	• Enter interface view: **interface** *interface-type interface-number* • Enter PVC view: **a. interface atm** *interface-number* **b. pvc** *vpi/vci*	Settings in interface view take effect on the current interface. Settings in PVC view take effect on the current PVC.
3. Apply a policy to the interface or PVC.	**qos apply policy** *policy-name* { **inbound** | **outbound** }	N/A

Configuring the maximum available interface bandwidth

The maximum available interface bandwidth refers to the maximum interface bandwidth used for bandwidth check when CBQ enqueues packets, rather than the actual bandwidth of the physical interface.

Configuration guidelines

- Hewlett Packard Enterprise recommends that you configure the maximum available interface bandwidth to be smaller than the actual available bandwidth of the physical interface or logical link.
- On a primary channel or template interface (such as VT) configured with the **qos max-bandwidth** command, AF and EF queues perform queue bandwidth check and calculation based on the bandwidth specified with the **qos max-bandwidth** command, so do the AF and EF queues synchronized to the sub-channel interfaces (for example, VA interfaces or B channels). Sub-channel interface bandwidth is ignored. Because the QoS configurations of the primary channel interface and the sub-channel interfaces are the same, prompts are output only for the primary channel interface. If the **qos max-bandwidth** command is not configured, AF and EF queues on the primary channel interface calculate queue bandwidth based on 1 Gbps of bandwidth, and AF and EF queues synchronized to the sub-channel interfaces calculate queue bandwidth based on actual sub-channel interface bandwidth. If queuing on a sub-channel interface fails due to bandwidth change, the prompt will be output for the sub-channel interface.
- On an MP-group interface or MFR interface configured with the **qos max-bandwidth** command, AF and EF perform queue bandwidth check and calculation based on the bandwidth specified with the **qos max-bandwidth** command. On an MP-group interface or MFR interface without the **qos max-bandwidth** command configured, if the sum of sub-channel bandwidth equals to or exceeds the sum of AF bandwidth and EF bandwidth, AF and EF calculate bandwidth based on the actual interface bandwidth. Otherwise, AF and EF calculate bandwidth based on 1 Gbps of bandwidth, and the message indicating insufficient bandwidth is displayed. In the latter case, the queuing function might fail to take effect.
- On tunnel interfaces, subinterfaces, Layer 3 aggregate interfaces, HDLC link bundle interfaces, RPR logical interfaces, or VT interfaces using PPPoE, PPPoA, or PPPoEoA at the data link

layer, you must configure the **qos max-bandwidth** command to provide base bandwidth for CBQ calculation.

Configuration procedure

To configure the maximum interface available bandwidth:

Step	Command
1. Enter system view.	**system-view**
2. Enter interface view.	**interface** *interface-type interface-number*
3. Configure the maximum available bandwidth of the interface.	**qos max-bandwidth** *bandwidth*

If no maximum available bandwidth is configured for an interface, the bandwidth used for CBQ calculation is as follows:

- For a physical interface, the actual baud rate or rate applies.
- For an E1, MFR or any other type of logical serial interface formed by timeslots or multiple links, the total bandwidth of all member channels/links applies.
- For a template interface, a VT interface for example, 1000000 kbps applies.
- For other virtual interfaces, a tunnel, Layer 3 aggregate, or HDLC link bundle interface for example, 0 kbps applies.

Configuration example

Configure the maximum interface available bandwidth.

# Enter system view.

```
<Sysname> system-view
```

# Enter interface view.

```
[Sysname] interface gigabitethernet 1/0/1
```

# Configure the maximum available bandwidth on interface GigabitEthernet 1/0/1.

```
[Sysname-GigabitEthernet1/0/1] qos max-bandwidth 60
```

Setting the maximum reserved bandwidth as a percentage of available bandwidth

The maximum reserved bandwidth is set on a per-interface basis. It decides the maximum bandwidth assignable for the QoS queues on an interface. It is typically set no greater than 80% of available bandwidth, considering the bandwidth for control traffic and Layer 2 frame headers.

Use the default maximum reserved bandwidth setting in normal cases. When tuning the setting, make sure that the Layer 2 frame header plus the data traffic is under the maximum available bandwidth of the interface.

To configure the maximum reserved bandwidth on an interface:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Set the maximum reserved bandwidth as a percentage of available bandwidth.	**qos reserved-bandwidth pct** *percent*	The default setting is 80.

Displaying and maintaining CBQ

Task	Command	Remarks
Display class configuration information.	**display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display traffic behavior configuration information.	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display QoS policy configuration information.	**display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display interface or PVC QoS policy configuration information.	**display qos policy interface** [*interface-type interface-number*] [**slot** *slot-number*] [**pvc** { *pvc-name* [*vpi/vci*] | *vpi/vci* }] [**inbound** | **outbound**] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display interface or PVC CBQ configuration information.	**display qos cbq interface** [*interface-type interface-number*] [**pvc** { *pvc-name* [*vpi/vci*] | *vpi/vci* }] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

CBQ configuration example

Network requirements

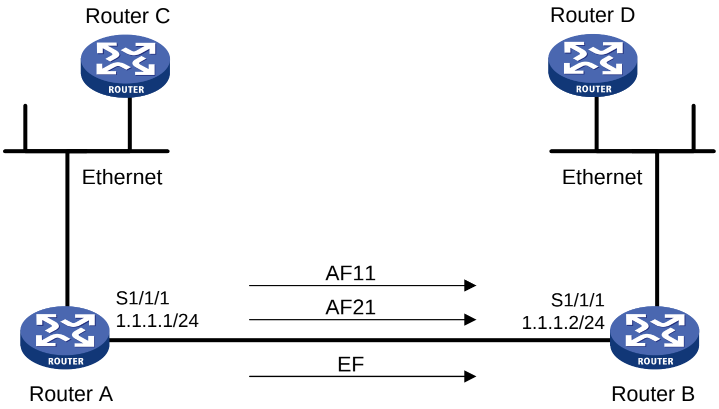
As shown in Figure 23, configure a QoS policy to meet the following requirements:

- Traffic from Router C is classified into three classes based on DSCP values. Perform AF for traffic with the DSCP values being AF11 and AF21, and set a minimum guaranteed bandwidth percentage of 5% for the traffic.
- Perform EF for traffic with the DSCP value being EF and set the maximum bandwidth percentage for the traffic to 30%.

Before performing the configuration, make sure that:

- The route from Router C to Router D through Router A and Router B is reachable.
- The DSCP fields have been set for the traffic before the traffic enters Router A.

Figure 23 Network diagram



Configuration procedure

Configure Router A:

# Define three classes to match the IP packets with the DSCP values AF11, AF21, and EF, respectively.

```
<RouterA> system-view
[RouterA] traffic classifier af11_class
[RouterA-classifier-af11_class] if-match dscp af11
[RouterA-classifier-af11_class] quit
[RouterA]traffic classifier af21_class
[RouterA-classifier-af21_class] if-match dscp af21
[RouterA-classifier-af21_class] quit
[RouterA] traffic classifier ef_class
[RouterA-classifier-ef_class] if-match dscp ef
[RouterA-classifier-ef_class] quit
```

# Define two traffic behaviors, and enable AF and set a minimum guaranteed bandwidth percentage of 5% in each traffic behavior.

```
[RouterA] traffic behavior af11_behav
[RouterA-behavior-af11_behav] queue af bandwidth pct 5
[RouterA-behavior-af11_behav] quit
[RouterA] traffic behavior af21_behav
[RouterA-behavior-af21_behav] queue af bandwidth pct 5
[RouterA-behavior-af21_behav] quit
```

# Define a traffic behavior, and enable EF and set a maximum bandwidth percentage of 30% (both bandwidth and delay are guaranteed for EF traffic) in the traffic behavior.

```
[RouterA] traffic behavior ef_behav
[RouterA-behavior-ef_behav] queue ef bandwidth pct 30
[RouterA-behavior-ef_behav] quit
```

# Define a QoS policy and associate the configured traffic behaviors with classes in the QoS policy.

```
[RouterA] qos policy dscp
[RouterA-qospolicy-dscp] classifier af11_class behavior af11_behav
[RouterA-qospolicy-dscp] classifier af21_class behavior af21_behav
[RouterA-qospolicy-dscp] classifier ef_class behavior ef_behav
[RouterA-qospolicy-dscp] quit
```

# Apply the QoS policy to the outgoing traffic of interface Serial 1/1/1.

```
[RouterA] interface serial 1/1/1
[RouterA-Serial1/1/1] ip address 1.1.1.1 255.255.255.0
RouterA-Serial1/1/1] qos apply policy dscp outbound
```

The configuration enables EF traffic to be forwarded preferentially when congestion occurs.

Configuring RTP priority queuing

Configuration procedure

To configure RTP priority queuing:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Configure RTP priority queuing.	**qos rtpq start-port** *first-rtp-port-number* **end-port** *last-rtp-port-number* **bandwidth** *bandwidth* [**cbs** *burst*]	N/A
4. Display RTP priority queuing configuration information on the interface or all interfaces.	**display qos rtpq interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

You must enable the rate limit function for the queuing function to take effect on these interfaces: tunnel interfaces, subinterfaces, Layer 3 aggregate interfaces, HDLC link bundle interfaces, RPR logical interfaces, and VT interfaces configured with PPPoE, PPPoA, or PPPoEoA.

RTP priority queuing configuration example

Network requirements

Configure RTP priority queuing on interface Serial 1/1/1 as follows:

The start port number is 16384, the end port number is 32767, and 64 kbps bandwidth is reserved for RTP packets. When congestion occurs to the outgoing interface, RTP packets are assigned to the RTP priority queue, whose maximum reserved bandwidth is configured as 70% of the available bandwidth.

Configuration procedure

# Enter system view.

```
<Sysname> system-view
```

# Enter interface view.

```
[Sysname] interface serial 1/1/1
```

# Configure the maximum reserved bandwidth as 70% of the available bandwidth on Serial 1/1/1.

```
[Sysname-Serial1/1/1] qos reserved-bandwidth pct 70
```

# Configure RTP priority queuing on interface Serial 1/1/1: the start port number is 16384, the end port number is 32767, and 64 kbps of bandwidth is reserved for RTP packets. When congestion occurs to the outgoing interface, RTP packets are assigned to the RTP priority queue.

```
[Sysname-Serial1/1/1] qos rtpq start-port 16384 end-port 32767 bandwidth 64
```

Configuring QoS tokens

Because the upper layer protocol TCP provides traffic control, CQ and WFQ might become invalid during FTP transmission. QoS tokens can solve this problem. The token feature of QoS provides a flow control mechanism for underlying-layer queues. This feature can control the number of packets sent to the interface underlying-layer queues based on the number of tokens.

Hewlett Packard Enterprise recommends that you set the token number to 1 on an interface for FTP transmission.

If the upper layer protocol, UDP for example, does not provide flow control, do not use the QoS token function to improve data transmission efficiency.

Configuration procedure

To configure QoS tokens:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	Applicable to only serial interfaces.
3. Specify the number of QoS tokens.	**qos qmtoken** *token-number*	By default, the QoS token feature is disabled.
4. Shut down the interface.	**shutdown**	Re-enable the interface by using the **shutdown** command and then the **undo shutdown** command to make the QoS token feature take effect.
5. Bring up the interface.	**undo shutdown**	N/A

QoS token configuration example

Network requirements

Specify the number of QoS tokens on interface Serial 1/1/1.

Configuration procedure

# Enter system view.

```
<Sysname> system-view
```

# Enter interface view.

```
[Sysname] interface serial 1/1/1
```

# Set the number of QoS tokens to 1, and re-enable the interface to make the configuration take effect.

```
[Sysname-Serial1/1/1] qos qmtoken 1
[Sysname-Serial1/1/1] shutdown
[Sysname-Serial1/1/1] undo shutdown
```

Configuring packet information pre-extraction

On a logical interface, such as a tunnel, RPR logical, Layer 3 aggregate, or HDLC link bundle interface, if the interface has processed the incoming IP packets, for example, if the tunnel interface has used GRE to encapsulate packets, the GRE-encapsulated packets enter the QoS module for processing. As a result, the QoS module cannot get the IP information of the original packets.

To process the original IP packets with QoS on the physical interface for a logical interface, configure packet information pre-extraction on the logical interface.

Configuration procedure

To configure packet information pre-extraction:

Step	**Command**	**Remarks**
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Enter interface view.	• **interface tunnel** *interface-number* • **interface rpr** *interface-number* • **interface route-aggregation** { *interface-number* | *interface-number***.***subnumber* } • **interface hdlc-bundle** *bundle-id*	Use any of the commands. Support for interface types depends on your device model.
3. Enable packet information pre-extraction.	**qos pre-classify**	By default, packet information pre-extraction is disabled.

For more information about tunnel interfaces, see *Layer 3—IP Services Configuration Guide*.

For more information about RPR logical interfaces, see *High Availability Configuration Guide*.

For more information about Layer 3 aggregate interfaces, see *Layer 2—LAN Switching Configuration Guide*.

For more information about HDLC link bundle interfaces, see *Layer 2—WAN Configuration Guide*.

Configuration example

Network requirements

Enable packet information pre-extraction on a tunnel interface.

Configuration procedure

# Enable packet information pre-extraction on tunnel interface Tunnel 0.

```
<Sysname> system-view
[Sysname] interface tunnel 0
[Sysname-Tunnel0] qos pre-classify
```

Configuring hardware congestion management

This feature is supported only on SAP modules operating in bridge mode.

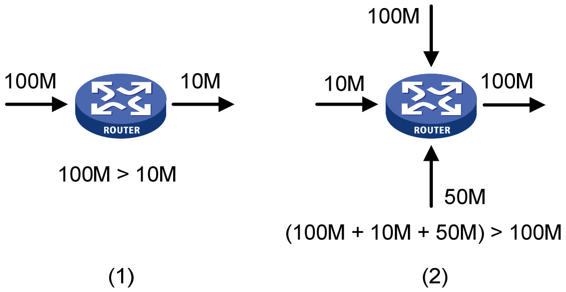
Overview

Causes, impacts, and countermeasures

Network congestion degrades service quality on a traditional network. Congestion is a situation where the forwarding rate decreases due to insufficient resources, resulting in extra delay.

Congestion is more likely to occur in complex packet switching circumstances. Figure 24 shows two common cases.

Figure 24 Traffic congestion causes



Congestion can bring the following negative results:

- Increased delay and jitter during packet transmission
- Decreased network throughput and resource use efficiency
- Network resource (memory, in particular) exhaustion and even system breakdown

Congestion is unavoidable in switched networks and multi-user application environments. To improve the service performance of your network, take proper measures to address the congestion issues.

The key to congestion management is how to define a dispatching policy for resources to decide the order of forwarding packets when congestion occurs.

Congestion management techniques

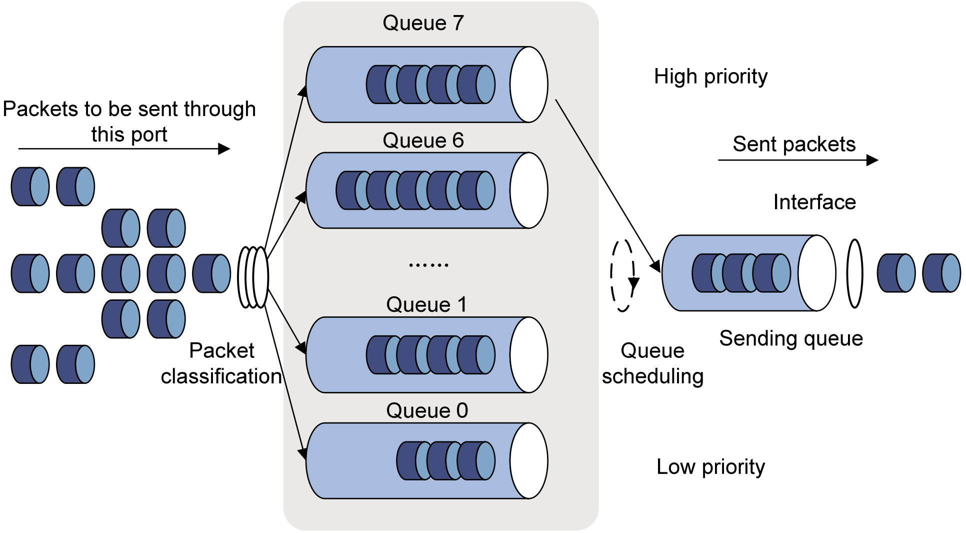
Congestion management uses queuing and scheduling algorithms to classify and sort traffic leaving a port. Each queuing algorithm addresses a particular network traffic problem, and has a different impact on bandwidth resource assignment, delay, and jitter.

Queue scheduling processes packets by their priorities, preferentially forwarding high-priority packets. The following section describes in detail Strict Priority (SP) queuing, Weighted Fair Queuing (WFQ), Weighted Round Robin (WRR) queuing, and Class-Based Queuing (CBQ).

SP queuing

SP queuing is designed for mission-critical applications that require preferential service to reduce the response delay when congestion occurs.

Figure 25 SP queuing



In Figure 25, SP queuing classifies eight queues on a port into eight classes, numbered 7 to 0 in descending priority order.

SP queuing schedules the eight queues in the descending order of priority. SP queuing sends packets in the queue with the highest priority first. When the queue with the highest priority is empty, it sends packets in the queue with the second highest priority, and so on. You can assign mission-critical packets to the high priority queue to make sure that they are always served first and common service packets to the low priority queues and transmitted when the high priority queues are empty.

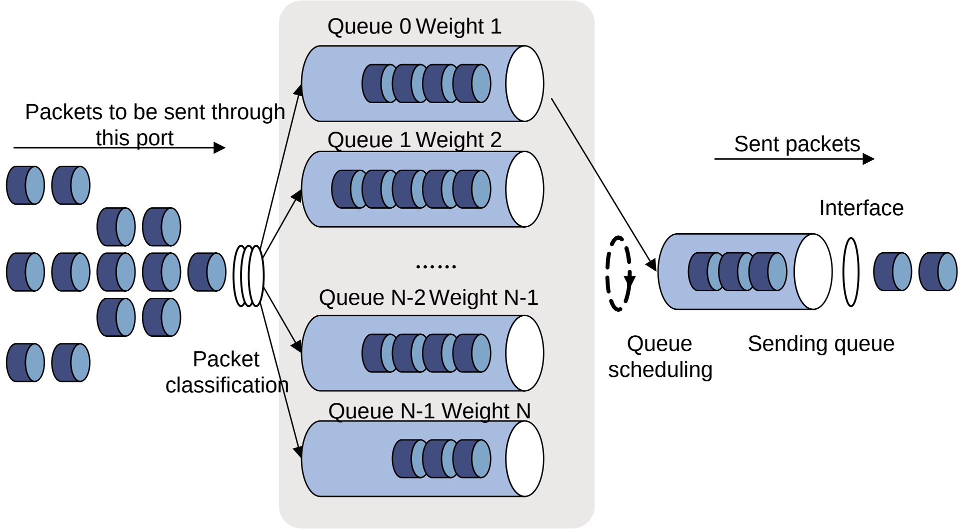
The disadvantage of SP queuing is that packets in the lower priority queues cannot be transmitted if packets exist in the higher priority queues. This might cause lower priority traffic to starve to death.

The router supports basic SP queuing, which contains multiple queues, with each queue corresponding to a different priority. These queues are scheduled in descending order of priority.

WRR queuing

WRR queuing schedules all the queues in turn to ensure every queue is served for a certain time, as shown in Figure 26.

Figure 26 WRR queuing



Assume a port provides eight output queues. WRR assigns each queue a weight value (represented by w7, w6, w5, w4, w3, w2, w1, or w0) to decide the proportion of resources assigned to the queue. On a 100 Mbps port, you can configure the weight values of WRR queuing to 50, 30, 10, 10, 50, 30, 10, and 10 (corresponding to w7, w6, w5, w4, w3, w2, w1, and w0, respectively). In this way, the queue with the lowest priority can get a minimum of 5 Mbps of bandwidth. WRR avoids the disadvantage of SP queuing that packets in low-priority queues might fail to be served for a long time.

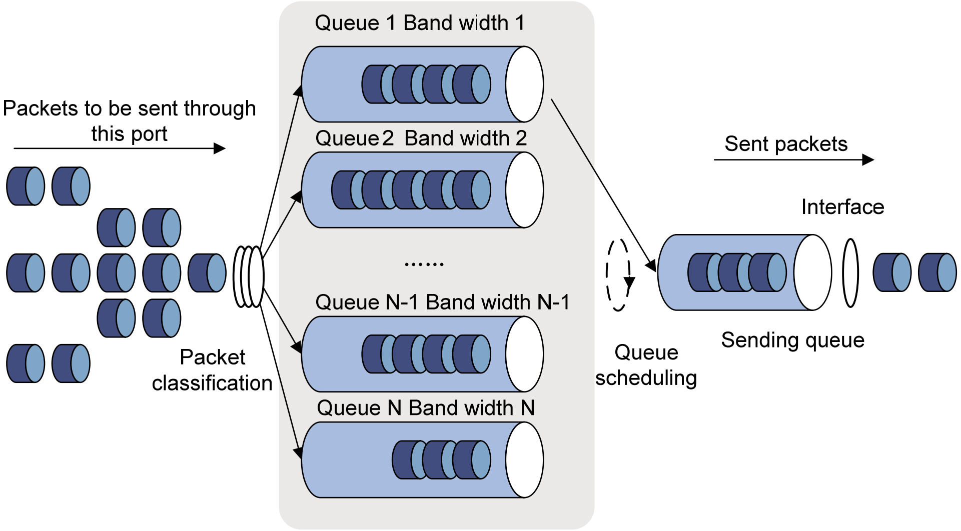
Another advantage of WRR queuing is that when the queues are scheduled in turn, the service time for each queue is not fixed. If a queue is empty, the next queue will be scheduled immediately. This improves bandwidth resource use efficiency.

WRR queuing includes the following types:

- **Basic WRR queuing**—Contains multiple queues. You can configure the weight, percentage (or byte count) for each queue and WRR schedules these queues based on the user-defined parameters in a round robin manner.
- **Group-based WRR queuing**—All the queues are scheduled by WRR. You can divide the output queue to WRR priority queue group 1 and WRR priority queue group 2. Round robin queue scheduling is performed for group 1 first. If group 1 is empty, round robin queue scheduling is performed for group 2.
- **WRR queuing with the maximum delay**—Assures that packets in the highest-priority queue are transmitted within the specified maximum delay, which makes it different from basic WRR queuing.

WFQ queuing

Figure 27 WFQ queuing



WFQ is similar to WRR. They both support scheduling weights in queue length, and can work with SP scheduling together. The difference is that WRR enables you to set the maximum time a packet waits in queue, but WFQ enables you to set guaranteed bandwidth a WFQ queue can get during congestion.

CBQ

CBQ provides one FIFO queue for each user-defined class to buffer traffic of the class. When the network is congested, CBQ classifies packets into user-defined classes, and assigns different classes of packets to different queues after performing congestion avoidance and bandwidth restriction check. When dequeuing packets, CBQ schedules packets from queues in proportion to their weights.

CBQ provides the following queuing types:

- **Low latency queuing (LLQ)**—LLQ queues are EF queues, and ensure strict priority service for real-time traffic. CBQ always schedules traffic in LLQ queues preferentially. To guarantee that other queues can get served when congestion occurs, you can set the maximum bandwidth for each LLQ queue. In normal cases, an LLQ queue can use more bandwidth than allocated. When congestion occurs, the exceeding traffic is dropped. You can also configure a burst size for LLQ queues.
- **Bandwidth queuing (BQ)**—BQ queues are AF queues. BQ provides strict, exact, guaranteed bandwidth for AF traffic, and schedules the AF classes proportionally.
- **WFQ**—One WFQ queue is available for BE traffic, and uses the remaining bandwidth to send the BE traffic.

BQ and WFQ use tail drop by default. You can configure a WRED drop policy to limit traffic.

Hardware congestion management configuration approaches

To manage hardware congestion, you can do the following:

- Configure queue scheduling for each queue in interface view or port group view, as described in Configuring per-queue hardware congestion management.
- Configure queue scheduling in a QoS policy, as described in Configuring CBQ.

Complete the following tasks to achieve hardware congestion management:

Task		**Remarks**
Configuring per-queue hardware congestion management	Configuring SP queuing	Optional.
	Configure group-based WRR queuing	Optional.
	Configuring WFQ queuing	Optional.
Configuring CBQ		Optional.

Configuring per-queue hardware congestion management

Configuring SP queuing

Configuration procedure

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or port group view.	• Enter interface view: **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*	SP queuing is only applicable to Layer 2 interfaces. Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.

3. Configure SP queuing.	**qos sp**	By default, basic SP queuing is used. Only Layer 2 Ethernet interfaces support configuring SP queuing.
4. Display SP queuing configuration.	**display qos sp interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

Configuration example

1. Network requirements

   Configure GigabitEthernet 1/0/1 to use SP queuing.

2. Configuration procedure

   # Enter system view

```
<Sysname> system-view
```

   # Configure GigabitEthernet1/0/1 to use SP queuing.

```
[Sysname] interface gigabitethernet 1/0/1
[Sysname-GigabitEthernet1/0/1] qos sp
```

Configure group-based WRR queuing

When a WRR queue is configured on an interface, WRR queuing is enabled on the interface, and other queues on the interface use the default WRR scheduling value and are assigned to the default WRR priority group.

Configuration procedure

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or port group view.	• Enter interface view: **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*	Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
3. Enable WRR queuing.	**qos wrr**	WRR queuing is only applicable to Layer 2 interfaces. By default, an interface uses SP queuing.
4. Configure a basic WRR queue.	**qos wrr** *queue-id* **weight** *schedule-value*	N/A
5. Display WRR queuing configuration information on interfaces.	**display qos wrr interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

1. Configuring group-based WRR queuing

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

2. Enter interface view or port group view.	• Enter interface view**:** **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*	Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
3. Enable WRR queuing.	**qos wrr**	WRR queuing is only applicable to Layer 2 interfaces. The default queuing algorithm on an interface varies with device models.
4. Assign a queue to a WRR group, and configure scheduling parameters for the queue.	**qos wrr** *queue-id* **group 1 weight** *schedule-value*	N/A
5. Assign a queue to the SP group.	**qos wrr** *queue-id* **group sp**	Optional. Queues in the SP group are scheduled by using the SP mechanism.
6. Display WRR queuing configuration information.	**display qos wrr interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

Configuration example

1. Network requirements
    - Enable WRR queuing on interface GigabitEthernet 1/0/1.
    - Assign queue 0 and queue 1 to the SP group.
    - Assign queue 2, queue 3, and queue 4 to WRR group 1, with the weight of 1, 5, and 10, respectively.
    - Assign queue 5 and queue 6 to WRR group 2, with the weight of 20 and 10, respectively.
2. Configuration procedure

    # Enter system view.

    ```
    <Sysname> system-view
    ```

    # Configure WRR queuing on interface GigabitEthernet 1/0/1.

    ```
    [Sysname] interface gigabitethernet 1/0/1
    [Sysname-GigabitEthernet1/0/1] qos wrr
    [Sysname-GigabitEthernet1/0/1] qos wrr 0 group sp
    [Sysname-GigabitEthernet1/0/1] qos wrr 1 group sp
    [Sysname-GigabitEthernet1/0/1] qos wrr 2 group 1 weight 1
    [Sysname-GigabitEthernet1/0/1] qos wrr 3 group 1 weight 5
    [Sysname-GigabitEthernet1/0/1] qos wrr 4 group 1 weight 10
    ```

Configuring WFQ queuing

With a WFQ queue configured, an interface has WFQ enabled. Other queues on the interface use the default WFQ scheduling value, which varies with device models.

Configuration procedure

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view or port group view.	• Enter interface view**:** **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*	Use one of the commands. Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
3. Enable WFQ queuing.	**qos wfq**	N/A
4. Configure a basic WFQ queue.	**qos wfq** *queue-id* **weight** *schedule-value*	WFQ queuing is only applicable to Layer 2 interfaces.
5. Configure the minimum guaranteed bandwidth for a WFQ queue.	**qos bandwidth queue** *queue-id* **min** *bandwidth-value*	Optional. WFQ queuing is only applicable to Layer 2 interfaces.
6. Display WFQ queuing configuration.	**display qos wfq interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

Configuration example

1. Network requirements

   Configure WFQ queues on an interface and assign the scheduling weight 1, 5, 10, 20, and 10 to queue 1, queue 3, queue 4, queue 5, and queue 6, respectively.

2. Configuration procedure

   # Enter system view.

```
<Sysname> system-view
```

   # Configure WFQ queues on GigabitEthernet 1/0/1.

```
[Sysname] interface gigabitethernet 1/0/1
[Sysname-GigabitEthernet1/0/1] qos wfq
[Sysname-GigabitEthernet1/0/1] qos wfq 1 weight 1
[Sysname-GigabitEthernet1/0/1] qos wfq 3 weight 5
[Sysname-GigabitEthernet1/0/1] qos wfq 4 weight 10
[Sysname-GigabitEthernet1/0/1] qos wfq 5 weight 20
[Sysname-GigabitEthernet1/0/1] qos wfq 6 weight 10
```

Configuring CBQ

CBQ is implemented through QoS policies.

When matching a packet against a QoS policy, the system:

- Matches the packet against class-behavior associations in the order they are configured.
- Matches the packet against the criteria in each class in the order they are configured.

CBQ configuration task list

Complete the following tasks to configure CBQ:

- Defining a class
- Defining a traffic behavior

- Defining a QoS policy
- Applying the QoS policy

Defining a class

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	By default, the **and** keyword is used, and the relation between match criteria is logical AND.
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A

Defining a traffic behavior

Configuring AF and the minimum guaranteed bandwidth

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified behavior name cannot be the name of any system-defined behavior.
3. Configure AF and the minimum guaranteed bandwidth.	**queue af bandwidth** *bandwidth*	N/A

You can apply this traffic behavior only to the outgoing traffic of an interface or ATM PVC.

Configuring EF and the maximum bandwidth

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Configure EF and the maximum bandwidth.	**queue ef bandwidth** *bandwidth* [**cbs** *burst*]	N/A

Configuring WFQ

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Configure WFQ.	**queue wfq**	N/A

Configuring a WRED drop action

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	The specified traffic behavior name cannot be the name of any system-defined behavior.
3. Configure a WRED drop action.	**wred** [**dscp** | **ip-precedence**]	• **dscp**—Uses the DSCP value for calculating the drop probability for a packet. • **ip-precedence**—Uses the IP precedence value for calculating the drop probability for a packet. This keyword is used by default.

You can configure the **wred** [**dscp** | **ip-precedence**] command only after configuring the **queue af** command or the **queue wfq** command.

When a QoS policy including the WRED traffic behavior is applied to an interface, the previous interface-level WRED configuration gets invalid.

Defining a QoS policy

To associate a traffic behavior with a specific class in policy view:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
3. Associate a traffic behavior with a class in the policy.	**classifier** *classifier-name* **behavior** *behavior-name*	• *classifier-name*: Class name. It must be the name of an existing system-defined or user-defined class. • *behavior-name*: Name of a behavior. It must be the name of an existing system-defined or user-defined behavior.

Applying the QoS policy

You can apply a policy to multiple physical interfaces or ATM PVCs.

To apply a CBQ policy to a physical interface or ATM PVC:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

2. Enter interface view, port group view, or PVC view.	- Enter interface view: **interface** *interface-type interface-number* - Enter port group view: **port-group manual** *port-group-name* - Enter PVC view: **a. interface atm** *interface-number* **b. pvc** *vpi/vci*	Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group. Settings in PVC view take effect on the current PVC.
3. Apply a policy to the interface or PVC.	**qos apply policy** *policy-name* { **inbound** | **outbound** }	On some cards, QoS policies can be applied but cannot take effect due to limited system resources. In this case, you can adjust related parameters (for example, reducing the number of queues) according to system prompt and then apply a QoS policy again.

Displaying and maintaining CBQ

Task	Command	Remarks
Display class configuration information.	**display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display traffic behavior configuration information.	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display QoS policy configuration information.	**display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display interface/PVC QoS policy configuration information.	**display qos policy interface** [*interface-type interface-number*] [**slot** *slot-number*] [**pvc** { *pvc-name* [*vpi/vci*] | *vpi/vci* }] [**inbound** | **outbound**] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display interface/PVC CBQ configuration information.	**display qos cbq interface** [*interface-type interface-number*] [**pvc** { *pvc-name* [*vpi/vci*] | *vpi/vci* }] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

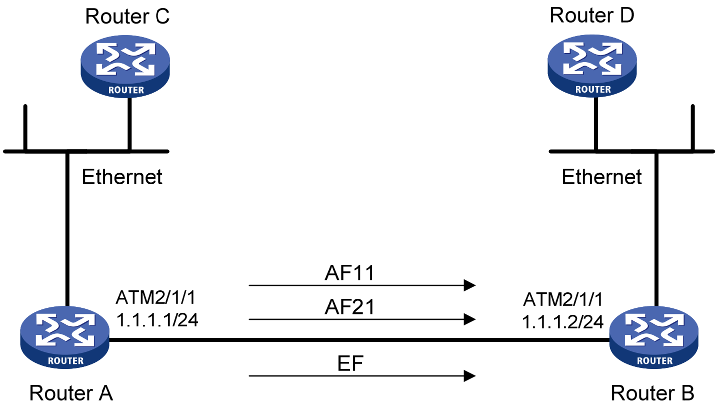
CBQ configuration example

Network requirements

As shown in Figure 28, configure a QoS policy to do the following:

- Classify traffic from Router C into three classes based on DSCP values.
- Perform AF for packets with DSCP AF11 or AF21, and set the minimum guaranteed bandwidth to 500 kbps for the packets.
- Perform EF for packets with DSCP EF, and set the maximum bandwidth to 2000 kbps for the packets.

Figure 28 Network diagram



Configuration procedure

Before performing the configuration, make sure that:

- Router C and Router D can reach each other through Router A and Router B.
- The DSCP field of the traffic has been set before it enters Router A.

Configure Router A:

# Define three classes to match the IP packets with DSCP AF11, AF21 and EF, respectively.

```
<RouterA> system-view
[RouterA] traffic classifier af11_class
[RouterA-classifier-af11_class] if-match dscp af11
[RouterA-classifier-af11_class] quit
[RouterA]traffic classifier af21_class
[RouterA-classifier-af21_class] if-match dscp af21
[RouterA-classifier-af21_class] quit
[RouterA] traffic classifier ef_class
[RouterA-classifier-ef_class] if-match dscp ef
[RouterA-classifier-ef_class] quit
```

# Define two traffic behaviors. Enable AF and set the minimum guaranteed bandwidth to 500 kbps in each traffic behavior.

```
[RouterA] traffic behavior af11_behav
[RouterA-behavior-af11_behav] queue af bandwidth 500
[RouterA-behavior-af11_behav] quit
[RouterA] traffic behavior af21_behav
[RouterA-behavior-af21_behav] queue af bandwidth 500
[RouterA-behavior-af21_behav] quit
```

# Define a traffic behavior. Enable EF and set the maximum bandwidth to 2000 kbps in the traffic behavior.

```
[RouterA] traffic behavior ef_behav
[RouterA-behavior-ef_behav] queue ef bandwidth 2000
[RouterA-behavior-ef_behav] quit
```

# Define a QoS policy, and associate the configured traffic behaviors with classes in the QoS policy.

```
[RouterA] qos policy dscp
[RouterA-qospolicy-dscp] classifier af11_class behavior af11_behav
[RouterA-qospolicy-dscp] classifier af21_class behavior af21_behav
```

```
[RouterA-qospolicy-dscp] classifier ef_class behavior ef_behav
[RouterA-qospolicy-dscp] quit
```

# Apply the QoS policy to the outgoing traffic of ATM PVC ATM 1/0.

```
[RouterA] interface 2/1/1
[RouterA-atm2/1/1] ip address 1.1.1.1 255.255.255.0
[RouterA-atm2/1/1] pvc qostest 0/40
[RouterA-atm-pvc-atm2/1/1-0/40-qostest] qos apply policy dscp outbound
```

When congestion occurs, Router A will forward EF traffic preferentially.

Configuring congestion avoidance

Overview

Avoiding congestion before it occurs is a proactive approach to improving network performance. As a flow control mechanism, congestion avoidance actively monitors network resources (such as queues and memory buffers), and drops packets when congestion is expected to occur or deteriorate.

Compared with end-to-end flow control, this flow control mechanism controls the load of more flows in a device. When dropping packets from a source end, it cooperates with the flow control mechanism (such as TCP flow control) at the source end to regulate the network traffic size. The combination of the local packet drop policy and the source-end flow control mechanism helps maximize throughput and network use efficiency and minimize packet loss and delay.

Tail drop

Congestion management techniques drop all packets arriving at a full queue. This tail drop mechanism results in global TCP synchronization. If packets from multiple TCP connections are dropped, these TCP connections go into the state of congestion avoidance and slow start to reduce traffic, but traffic peak occurs later. Consequently, the network traffic jitters all the time.

RED and WRED

You can use random early detection (RED) or weighted random early detection (WRED) to avoid global TCP synchronization.

Both RED and WRED avoid global TCP synchronization by randomly dropping packets. When the sending rates of some TCP sessions slow down after their packets are dropped, other TCP sessions remain at high sending rates. Link bandwidth is efficiently used because TCP sessions at high sending rates always exist.

The RED or WRED algorithm sets an upper threshold and lower threshold for each queue, and processes the packets in a queue as follows:

- When the queue size is shorter than the lower threshold, no packet is dropped.
- When the queue size reaches the upper threshold, all subsequent packets are dropped.
- When the queue size is between the lower threshold and the upper threshold, the received packets are dropped at random. The drop probability in a queue increases along with the queue size under the maximum drop probability.

WRED uses differentiated drop policies for different IP precedence values. Packets with a lower IP precedence are more likely to be dropped.

If the current queue size is compared with the upper threshold and lower threshold to determine the drop policy, bursty traffic is not fairly treated. To solve this problem, WRED compares the average queue size with the upper threshold and lower threshold to determine the drop probability.

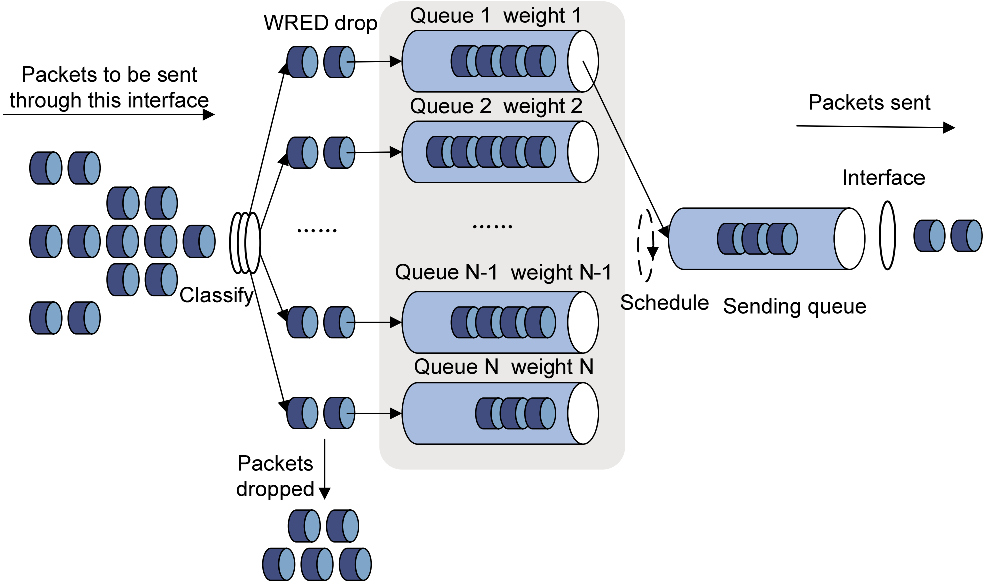
The average queue size reflects the queue size change trend but is not sensitive to bursty queue size changes, and bursty traffic can be fairly treated. The average queue size is calculated using the following formula: average queue size = previous average queue size × $(1-2^{-n})$ + current queue size × 2^{-n}, where n can be configured with the **qos wred weighting-constant** command.

With WFQ queuing used, you can set the exponent for average queue size calculation, upper threshold, lower threshold, and drop probability for packets with different precedence values to provide differentiated drop policies.

With FIFO queuing, PQ, or CQ used, you can set the exponent for average queue size calculation, upper threshold, lower threshold, and drop probability for each queue to provide differentiated drop policies for different classes of packets.

Relationship between WRED and queuing mechanisms

Figure 29 Relationship between WRED and queuing mechanisms



Through combining WRED with WFQ, the flow-based WRED can be realized. Because each flow has its own queue after classification, a flow with a smaller queue size has a lower packet drop probability, when a flow with a larger queue size has a higher packet drop probability. In this way, the benefits of the flow with a smaller queue size are protected.

Introduction to WRED configuration

WRED configuration approaches

You can configure WRED parameters on an interface and enable WRED.

Introduction to WRED parameters

Determine the following parameters before configuring WRED:

- **The upper threshold and lower threshold**—When the average queue size is smaller than the lower threshold, no packet is dropped. When the average queue size is between the lower threshold and the upper threshold, the packets are dropped at random. The longer the queue is, the higher the drop probability is. When the average queue size exceeds the upper threshold, subsequent packets are dropped.
- **The exponent used for average queue size calculation**—The bigger the exponent is, the less sensitive the average queue size is to real-time queue size changes.
- **Denominator for drop probability calculation**—The bigger the denominator is, the smaller the calculated drop probability is.

Configuring WRED on an interface

Configuration procedure

Before configuring the **qos wred enable** command, you must enable WFQ queuing on the interface.

To configure WRED on an interface:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Enable WRED.	**qos wred** [**dscp** | **ip-precedence**] **enable**	N/A
4. Set the WRED exponent for average queue size calculation.	**qos wred weighting-constant** *exponent*	Optional. The default setting is 9.
5. Set the drop-related parameters for the precedence.	**qos wred** { **ip-precedence** *ip-precedence* | **dscp** *dscp-value* } **low-limit** *low-limit* **high-limit** *high-limit* **discard-probability** *discard-prob*	Optional. By default, the *low-limit* argument is 10, the *high-limit* argument is 30, and the *discard-prob* argument is 10.
6. Display the WRED configuration on an interface/PVC or all interfaces/PVCs.	**display qos wred interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Configuration example

Network requirements

- Enable IP precedence-based WRED on interface GigabitEthernet 1/0/1.
- Set the following parameters for packets with IP precedence 3: lower threshold 20, upper threshold 40, and drop probability denominator 15.
- Set the exponential factor for the average queue size calculation to 6.

Configuration procedure

# Enter system view.

```
<Sysname> system-view
```

# Enter interface view.

```
[Sysname] interface gigabitethernet 1/0/1
```

# Enable IP precedence-based WRED.

```
[Sysname-GigabitEthernet1/0/1] qos wred ip-precedence enable
```

# Set the following parameters for packets with IP precedence 3: lower threshold 20, upper threshold 40, and drop probability denominator 15.

```
[Sysname-GigabitEthernet1/0/1] qos wred ip-precedence 3 low-limit 20 high-limit 40
discard-probability 15
```

# Set the exponential factor for the average queue size calculation to 6.

```
[Sysname-GigabitEthernet1/0/1] qos wred weighting-constant 6
```

Applying a WRED table on an interface

This feature is supported only on SAP modules operating in bridge mode.

A WRED table contains global WRED parameters and supports only the queue-based table. With a queue-based table, packets are dropped based on the queue when congestion occurs.

A queue-based WRED table can be applied to multiple interfaces. For a queue-based WRED table already applied to an interface, you can modify the values of the queue-based WRED table, but you cannot remove the queue-based WRED table.

Configuration procedure

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a WRED table and enter its view.	**qos wred queue table** *table-name*	By default, no WRED table exists.
3. Configure the other WRED parameters.	**queue** *queue-value* [**drop-level** *drop-level*] **low-limit** *low-limit* [**discard-probability** *discard-prob*]	Optional.
4. Enter interface view or port group view.	• Enter interface view**:** **interface** *interface-type interface-number* • Enter port group view: **port-group manual** *port-group-name*	Settings in interface view take effect on the current interface. Settings in port group view take effect on all ports in the port group.
5. Apply the WRED table to the interface or port group.	**qos wred apply** *table-name*	A queue-based WRED table is available on only Layer 2 ports.
6. Display the configuration of a WRED table or all WRED tables.	**display qos wred table** [*table-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Configuration example

Apply a queue-based WRED table to Layer 2 port GigabitEthernet 1/0/1:

# Enter system view.

```
<Sysname> system-view
```

# Configure a queue-based WRED table.

```
[Sysname] qos wred queue table queue-table1
[Sysname-wred-table-queue-table1] quit
```

# Enter interface view.

```
[Sysname] interface gigabitethernet 1/0/1
```

# Apply the queue-based WRED table to GigabitEthernet 1/0/1.

```
[Sysname-GigabitEthernet1/0/1] qos wred apply queue-table1
```

Displaying and maintaining WRED

Task	Command	Remarks
Display the WRED configuration on an interface/PVC or all interfaces/PVCs.	**display qos wred interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display the configuration of a WRED table or all WRED tables.	**display qos wred table** [*table-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

WRED configuration example

Network requirements

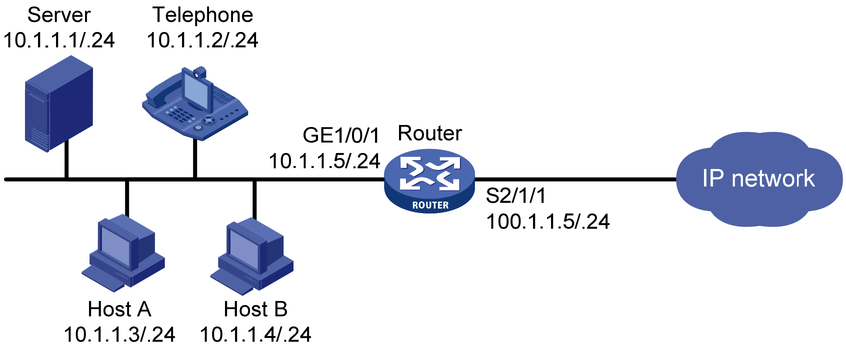
As shown in Figure 30, Server sends critical data traffic, Telephone sends voice traffic, and Host A and Host B send non-critical data traffic. On Router, because the rate of incoming interface GigabitEthernet 1/0/1 is higher than that of outgoing interface Serial 2/1/1, congestion might occur on Serial 2/1/1.

Perform configurations to meet the following requirements:

1. Critical traffic from Server and Telephone is transmitted preferentially when congestion occurs in the network.
2. Certain bandwidth is guaranteed for traffic from Host A and Host B to reduce traffic delay.
3. When congestion deteriorates, packets are dropped based on precedence.

Use WFQ in conjunction with WRED for queue scheduling and packet dropping.

Figure 30 Network diagram



Configuration procedure

# Configure ACLs to match the packets from Server, Telephone, Host A, and Host B, respectively.

```
<Router> system-view
[Router] acl number 2001
[Router-acl-basic-2001] rule 1 permit source 10.1.1.1 0
[Router-acl-basic-2001] quit
[Router] acl number 2002
[Router-acl-basic-2002] rule 2 permit source 10.1.1.2 0
[Router-acl-basic-2002] quit
[Router] acl number 2003
```

```
[Router-acl-basic-2003] rule 3 permit source 10.1.1.3 0
[Router-acl-basic-2003] quit
[Router] acl number 2004
[Router-acl-basic-2004] rule 1 permit source 10.1.1.4 0
[Router-acl-basic-2004] quit
```

# Mark each flow with a priority.

```
[Router] traffic classifier class1
[Router-classifier-class1] if-match acl 2001
[Router-classifier-class1] quit
[Router] traffic classifier class2
[Router-classifier-class2] if-match acl 2002
[Router-classifier-class2] quit
[Router] traffic classifier class3
[Router-classifier-class3] if-match acl 2003
[Router-classifier-class3] quit
[Router] traffic classifier class4
[Router-classifier-class4] if-match acl 2004
[Router-classifier-class4] quit
[Router] traffic behavior behavior1
[Router-behavior-behavior1] remark ip-precedence 5
[Router-behavior-behavior1] quit
[Router]traffic behavior behavior2
[Router-behavior-behavior2] remark ip-precedence 4
[Router-behavior-behavior2] quit
[Router] traffic behavior behavior3
[Router-behavior-behavior3] remark ip-precedence 3
[Router-behavior-behavior3] quit
[Router] traffic behavior behavior4
[Router-behavior-behavior4] remark ip-precedence 2
[Router-behavior-behavior4] quit
[Router] qos policy aa
[Router-qospolicy-aa] classifier class1 behavior behavior1
[Router-qospolicy-aa] classifier class2 behavior behavior2
[Router-qospolicy-aa] classifier class3 behavior behavior3
[Router-qospolicy-aa] classifier class4 behavior behavior4
[Router-qospolicy-aa] quit
[Router] interface gigabitethernet 1/0/1
[Router-GigabitEthernet1/0/1] qos apply policy aa inbound
[Router-GigabitEthernet1/0/1] quit
```

# Configure WFQ to process packets both fairly and based on precedence. Configure WRED to drop packets by precedence when congestion deteriorates.

```
[Router] interface Serial 2/1/1
[Router-Serial2/1/1] qos wfq queue-length 100 queue-number 16
[Router-Serial2/1/1] qos wred enable
[Router-Serial2/1/1] qos wred ip-precedence 5 low-limit 10 high-limit 250
discard-probability 11
[Router-Serial2/1/1] qos wred ip-precedence 4 low-limit 10 high-limit 200
discard-probability 12
```

```
[Router-Serial2/1/1] qos wred ip-precedence 3 low-limit 10 high-limit 180
discard-probability 15
[Router-Serial2/1/1] qos wred ip-precedence 2 low-limit 10 high-limit 180
discard-probability 15
[Router-Serial2/1/1] quit
```

Configuring traffic filtering

You can filter in or filter out a class of traffic by associating the class with a traffic filtering action. For example, you can filter packets sourced from a specific IP address according to network status.

Configuration procedure

To configure traffic filtering:

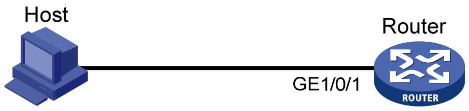
Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Configure the traffic filtering action.	**filter** { **deny** | **permit** }	• **deny**—Drops packets. • **permit**—Permits packets to pass through.
7. Return to system view.	**quit**	N/A
8. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
9. Associate the class with the traffic behavior in the QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A
10. Return to system view.	**quit**	N/A
11. Apply the QoS policy.	• Applying the QoS policy to an interface or PVC • Applying the QoS policy to a VLAN • Applying the QoS policy to online users	Choose one of the application destinations as needed.
12. Display the traffic filtering configuration.	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

Traffic filtering configuration example

Network requirements

As shown in Figure 31, configure traffic filtering to filter the packets with source port not being 21, and received on GigabitEthernet 1/0/1.

Figure 31 Network diagram



Configuration procedure

# Create advanced ACL 3000, and configure a rule to match packets whose source port number is not 21.

```
<Router> system-view
[Router] acl number 3000
[Router-acl-adv-3000] rule 0 permit tcp source-port neq 21
[Router-acl-adv-3000] quit
```

# Create a class named **classifier_1**, and use ACL 3000 as the match criterion in the class.

```
[Router] traffic classifier classifier_1
[Router-classifier-classifier_1] if-match acl 3000
[Router-classifier-classifier_1] quit
```

# Create a behavior named **behavior_1**, and configure the traffic filtering action to drop packets.

```
[Router] traffic behavior behavior_1
[Router-behavior-behavior_1] filter deny
[Router-behavior-behavior_1] quit
```

# Create a policy named **policy**, and associate class **classifier_1** with behavior **behavior_1** in the policy.

```
[Router] qos policy policy
[Router-qospolicy-policy] classifier classifier_1 behavior behavior_1
[Router-qospolicy-policy] quit
```

# Apply the policy named **policy** to the incoming traffic of GigabitEthernet 1/0/1.

```
[Router] interface gigabitethernet 1/0/1
[Router-GigabitEthernet1/0/1] qos apply policy policy inbound
```

Configuring priority marking

Priority marking sets the priority fields or flag bits of packets to modify the priority of traffic. For example, you can use priority marking to set IP precedence or DSCP for a class of IP traffic to change its transmission priority in the network.

To configure priority marking, you can associate a class with a behavior configured with the priority marking action to set the priority fields or flag bits of the class of packets.

Configuration procedure

To configure priority marking:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Set the DSCP value for packets.	**remark dscp** *dscp-value*	Optional.
7. Set the 802.1p priority for packets or configure the inner-to-outer tag priority copying function.	**remark dot1p** *8021p*	Optional.
8. Set the IP precedence for packets.	**remark ip-precedence** *ip-precedence-value*	Optional.
9. Set the local precedence for packets.	**remark local-precedence** *local-precedence*	Optional.
10. Set the QoS-local ID for packets.	**remark qos-local-id** *local-id-value*	Optional.
11. Return to system view.	**quit**	N/A
12. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
13. Associate the class with the traffic behavior in the QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A
14. Return to system view.	**quit**	N/A
15. Apply the QoS policy.	• Applying the QoS policy to an interface or PVC • Applying the QoS policy to a VLAN • Applying the QoS policy to online users	Choose one of the application destinations as needed.
16. Display the priority marking configuration.	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Optional. Available in any view.

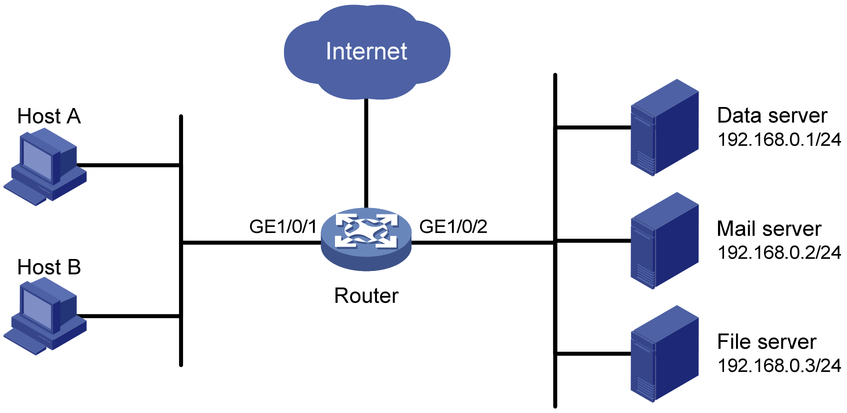
Priority marking configuration example

Network requirements

As shown in Figure 32, configure priority marking on Router to meet the following requirements:

Traffic source	Destination	Processing priority
Host A, B	Data server	High
Host A, B	Mail server	Medium
Host A, B	File server	Low

Figure 32 Network diagram



Configuration procedure

# Create advanced ACL 3000, and configure a rule to match packets with destination IP address 192.168.0.1.

```
<Router> system-view
[Router] acl number 3000
[Router-acl-adv-3000] rule permit ip destination 192.168.0.1 0
[Router-acl-adv-3000] quit
```

# Create advanced ACL 3001, and configure a rule to match packets with destination IP address 192.168.0.2.

```
[Router] acl number 3001
[Router-acl-adv-3001] rule permit ip destination 192.168.0.2 0
[Router-acl-adv-3001] quit
```

# Create advanced ACL 3002, and configure a rule to match packets with destination IP address 192.168.0.3.

```
[Router] acl number 3002
[Router-acl-adv-3002] rule permit ip destination 192.168.0.3 0
[Router-acl-adv-3002] quit
```

# Create a class named **classifier_dbserver**, and use ACL 3000 as the match criterion in the class.

```
[Router] traffic classifier classifier_dbserver
[Router-classifier-classifier_dbserver] if-match acl 3000
```

```
[Router-classifier-classifier_dbserver] quit
```

# Create a class named **classifier_mserver**, and use ACL 3001 as the match criterion in the class.

```
[Router] traffic classifier classifier_mserver
[Router-classifier-classifier_mserver] if-match acl 3001
[Router-classifier-classifier_mserver] quit
```

# Create a class named **classifier_fserver**, and use ACL 3002 as the match criterion in the class.

```
[Router] traffic classifier classifier_fserver
[Router-classifier-classifier_fserver] if-match acl 3002
[Router-classifier-classifier_fserver] quit
```

# Create a behavior named **behavior_dbserver**, and configure the action of setting the DSCP value to 32.

```
[Router] traffic behavior behavior_dbserver
[Router-behavior-behavior_dbserver] remark dscp 32
[Router-behavior-behavior_dbserver] quit
```

# Create a behavior named **behavior_mserver**, and configure the action of setting the DSCP value to 24.

```
[Router] traffic behavior behavior_mserver
[Router-behavior-behavior_mserver] remark dscp 24
[Router-behavior-behavior_mserver] quit
```

# Create a behavior named **behavior_fserver**, and configure the action of setting the DSCP value to 16.

```
[Router] traffic behavior behavior_fserver
[Router-behavior-behavior_fserver] remark dscp 16
[Router-behavior-behavior_fserver] quit
```

# Create a policy named **policy_server**, and associate classes with behaviors in the policy.

```
[Router] qos policy policy_server
[Router-qospolicy-policy_server] classifier classifier_dbserver behavior
behavior_dbserver
[Router-qospolicy-policy_server] classifier classifier_mserver behavior
behavior_mserver
[Router-qospolicy-policy_server] classifier classifier_fserver behavior
behavior_fserver
[Router-qospolicy-policy_server] quit
```

# Apply the policy named **policy_server** to the incoming traffic of GigabitEthernet 1/0/1.

```
[Router] interface gigabitethernet 1/0/1
[Router-GigabitEthernet1/0/1] qos apply policy policy_server inbound
[Router-GigabitEthernet1/0/1] quit
```

Configuring traffic redirecting

This feature is supported only on SAP modules operating in bridge mode.

Traffic redirecting is the action of redirecting the packets matching the specific match criteria to a certain location for processing.

The following redirect actions are supported:

- **Redirecting traffic to the CPU**—Redirects packets that require processing by the CPU to the CPU.
- **Redirecting traffic to an interface**—Redirects packets that require processing by an interface to the interface. Note that this action applies to only Layer 2 packets, and the target interface must be a Layer 2 interface.
- **Redirecting traffic to the next hop**—Redirects packets that require processing by an interface to the interface. This action only applies to Layer 3 packets.

Configuration guidelines

- The actions of redirecting traffic to the CPU and redirecting traffic to an interface are mutually exclusive with each other in the same traffic behavior.
- You can use the **display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**|** { **begin** | **exclude** | **include** } *regular-expression*] command to view the traffic redirecting configuration.

Configuration procedure

To configure traffic redirecting:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Configure a traffic redirecting action.	**redirect** { **cpu** | **interface** *interface-type interface-number* | **next-hop** { *ipv4-add1* [*ipv4-add2*]| *ipv6-add1* [*ipv6-add2*] } }	Optional.
7. Return to system view.	**quit**	N/A
8. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
9. Associate the class with the traffic behavior in the QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A
10. Return to system view.	**quit**	N/A

Step	Command	Remarks
11. Apply the QoS policy.	- Applying the QoS policy to an interface or PVC - Applying the QoS policy to a VLAN	Choose one of the application destinations as needed.

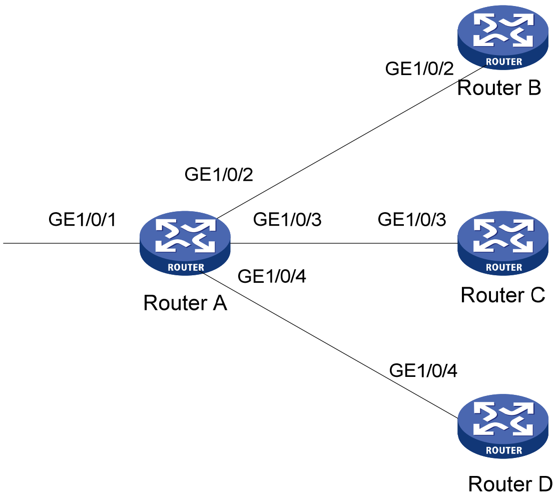
Traffic redirecting configuration example

Network requirements

As shown in Figure 33, configure the actions of redirecting traffic to interfaces:

- Packets with source IP address 2.1.1.1 received on GigabitEthernet 1/0/1 of Router A are forwarded out of GigabitEthernet 1/0/2.
- Packets with source IP address 2.1.1.2 received on GigabitEthernet 1/0/1 of Router A are forwarded out of GigabitEthernet 1/0/3.
- Other packets received on GigabitEthernet 1/0/1 of Router A are forwarded out of GigabitEthernet 1/0/4.

Figure 33 Network diagram



Configuration procedure

# Create basic ACL 2000, and configure a rule to match packets with source IP address 2.1.1.1.

```
<RouterA> system-view
[RouterA] acl number 2000
[RouterA-acl-basic-2000] rule permit source 2.1.1.1 0
[RouterA-acl-basic-2000] quit
```

# Create basic ACL 2001, and configure a rule to match packets with source IP address 2.1.1.2.

```
[RouterA] acl number 2001
[RouterA-acl-basic-2001] rule permit source 2.1.1.2 0
[RouterA-acl-basic-2001] quit
```

# Create a class named **classifier_1**, and use ACL 2000 as the match criterion in the class.

```
[RouterA] traffic classifier classifier_1
[RouterA-classifier-classifier_1] if-match acl 2000
```

```
[RouterA-classifier-classifier_1] quit
```

# Create a class named **classifier_2**, and use ACL 2001 as the match criterion in the class.

```
[RouterA] traffic classifier classifier_2
[RouterA-classifier-classifier_2] if-match acl 2001
[RouterA-classifier-classifier_2] quit
```

# Create a class named **classifier_3** that does not match ACL 2000 or ACL 2001.

```
[RouterA] traffic classifier classifier_3
[RouterA-classifier-classifier_3] if-match not acl 2000
[RouterA-classifier-classifier_3] if-match not acl 2001
[RouterA-classifier-classifier_3] quit
```

# Create a behavior named **behavior_1**, and configure the action of redirecting traffic to interface GigabitEthernet 1/0/2.

```
[RouterA] traffic behavior behavior_1
[RouterA-behavior-behavior_1] redirect interface GigabitEthernet 1/0/2
[RouterA-behavior-behavior_1] quit
```

# Create a behavior named **behavior_2**, and configure the action of redirecting traffic to interface GigabitEthernet 1/0/3.

```
[RouterA] traffic behavior behavior_2
[RouterA-behavior-behavior_2] redirect interface GigabitEthernet 1/0/3
[RouterA-behavior-behavior_2] quit
```

# Create a behavior named **behavior_3**, and configure the action of redirecting traffic to interface GigabitEthernet 1/0/4.

```
[RouterA] traffic behavior behavior_3
[RouterA-behavior-behavior_3] redirect interface gigabitethernent 1/0/4
[RouterA-behavior-behavior_3] quit
```

# Create a policy named **policy**, and associate classes **classifier_1**, **classifier_2**, and **classifier_3** with behaviors **behavior_1**, **behavior_2**, and **behavior_3** in the policy, respectively.

```
[RouterA] qos policy policy
[RouterA-qospolicy-policy] classifier classifier_1 behavior behavior_1
[RouterA-qospolicy-policy] classifier classifier_2 behavior behavior_2
[RouterA-qospolicy-policy] classifier classifier_3 behavior behavior_3
[RouterA-qospolicy-policy] quit
```

# Apply the policy named **policy** to the incoming traffic of GigabitEthernet 1/0/1.

```
[RouterA] interface gigabitethernet 1/0/1
[RouterA-GigabitEthernet1/0/1] qos apply policy policy inbound
```

Configuring DAR

The feature is applicable only to IP packets.

The following matrix shows the feature and router compatibility:

Feature	HSR6602	6604/6608/6616
DAR	No	No

Overview

The Deeper Application Recognition (DAR) feature identifies packets of dynamic protocols like BitTorrent by examining Layer 4 to Layer 7 content other than the IP header. The feature helps service providers and businesses limit aggressive bandwidth use by applications like BitTorrent to ensure fairness and network performance.

BitTorrent is a P2P file sharing communications protocol, which enables personal computers to directly exchange data or services. P2P has been widely used for content (such as audio and video) file sharing. It represents a large amount of bandwidth on the Internet.

DAR can limit, block, or manipulate identified application traffic depending on your configuration, guaranteeing key applications high priority treatment and protecting customer investment.

Configuring DAR for P2P traffic recognition

DAR uses a .mtd P2P signature file for P2P traffic identification. It compares the content of every incoming packet with the signature file. If a match is found, DAR processes the packet as a P2P packet.

Loading the P2P signature file

To identify P2P traffic, first load the P2P signature file. Make sure that the signature file is placed in the root directory. The system can load a signature file only from the root directory.

To load the P2P signature file:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Load the P2P signature file.	**dar p2p signature-file** *filename*	By default, no P2P signature file is loaded.

Configuring a P2P protocol group

You can configure a P2P protocol group to include multiple P2P protocols. By referencing this P2P protocol group in a traffic class, you assign traffic of these P2P protocols to the class and implement the same QoS policy.

To configure a P2P protocol group:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a P2P protocol group and enter protocol group view.	**dar protocol-group** *group-id*	By default, no protocol group exists in the system.
3. Assign a protocol to the protocol group.	**protocol** *protocol-name*	By default, a protocol group contains no protocol.

Enabling P2P traffic recognition

P2P traffic recognition is system resource demanding. It is disabled by default to avoid impacts on other modules.

To enable DAR for traffic recognition:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Layer 3 Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3. Enable P2P traffic recognition.	**dar enable**	By default, P2P traffic recognition is disabled.

Configuring protocol match criteria

To apply QoS policies to data streams to set packet priority or allocate bandwidth for example, use DAR to classify the data streams first.

To configure protocol match criteria:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure the match criterion for a protocol group.	**if-match** [**not**] **protocol** *protocol group-id*	Optional. By default, no match criterion is configured for a protocol group.

Configuring DAR packet accounting

With the packet accounting function of DAR, you can monitor the number of packets and the amount of data traffic of application protocols on each interface. According to the statistics, you can apply a correct QoS policy to the traffic.

To configure DAR packets accounting:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter Ethernet interface view.	**interface** *interface-type interface-number*	N/A
3. Enable DAR packet accounting.	**dar protocol-statistic** [**flow-interval** *time*]	By default, DAR packet accounting is disabled.

Displaying and maintaining DAR

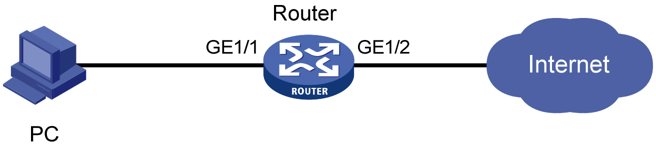
Task	**Command**	**Remarks**
Display DAR protocol packet statistics.	**display dar protocol-statistic** [**protocol** *protocol-name* | **top** *top-number* | **all**] [**interface** *interface-type interface-number*] [**direction** { **in** | **out** }] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Clear DAR protocol packet statistics.	**reset dar protocol-statistic** { { **protocol** *protocol-name* | **interface** *interface-type interface-number* } * | **all** }	Available in user view.

Blocking P2P downloading configuration example

Network requirements

As shown in Figure 34, configure the router to prevent BT clients or eMule/eDonkey clients on the PCs from downloading files from the Internet.

Figure 34 Network diagram



Configuration procedure

# Load the P2P signature file **meta.mtd**.

```
<Router> system-view
[Router] dar p2p signature-file meta.mtd
```

# Configure protocol group 1.

```
[Router] dar protocol-group 1
[Router-protocol-group-1] protocol bittorrent
[Router-protocol-group-1] protocol eMule/eDonkey
[Router-protocol-group-1] quit
```

# Create a class and reference protocol group 1 in it.

```
[Router] traffic classifier p2p
[Router-classifier-p2p] if-match protocol-group 1
```

```
[Router-classifier-p2p] quit
```

# Configure a packet filtering behavior.

```
[Router] traffic behavior deny
[Router-behavior-deny] filter deny
[Router-behavior-deny] quit
```

# Create a QoS policy and associate the traffic behavior with the class in the policy.

```
[Router] qos policy p2p
[Router-qospolicy-p2p] classifier bt behavior deny
[Router-qospolicy-p2p] quit
```

# Enable P2P traffic recognition on GigabitEthernet 1/1, and apply the QoS policy to the incoming traffic of GigabitEthernet 1/0/1.

```
[Router] interface gigabitethernet 1/1
[Router-GigabitEthernet1/1] dar enable
[Router-GigabitEthernet1/1] qos apply policy p2p inbound
```

Run the BT client and the eMule/eDonkey client on a connected PC and start to download files.

Check the interfaces of the clients, and you can see they cannot download files.

Configuring class-based accounting

Class-based accounting collects statistics (in number of packets or bytes) on a per-traffic class basis. For example, you can define the action to collect statistics for traffic sourced from a certain IP address. By analyzing the statistics, you can determine whether anomalies have occurred and what action to take.

Configuration procedure

To configure class-based accounting:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Configure the accounting action.	**accounting**	Optional. The router supports packet-based accounting.
7. Return to system view.	**quit**	N/A
8. Create a policy and enter policy view.	**qos policy** *policy-name*	N/A
9. Associate the class with the traffic behavior in the QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A
10. Return to system view.	**quit**	N/A
11. Apply the QoS policy.	• Applying the QoS policy to an interface or PVC • Applying the QoS policy to a VLAN	Choose one of the application destinations as needed.

Displaying and maintaining class-based accounting

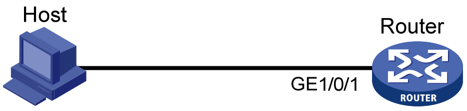
To verify the class-based accounting configuration, use the **display qos policy** command in any view to display the traffic statistics collected after the configuration is complete.

Class-based accounting configuration example

Network requirements

As shown in Figure 35, configure class-based accounting to collect statistics for traffic sourced from 1.1.1.1/24 and received on GigabitEthernet 1/0/1.

Figure 35 Network diagram



Configuration procedure

# Create basic ACL 2000, and configure a rule to match packets with source IP address 1.1.1.1.

```
<Router> system-view
[Router] acl number 2000
[Router-acl-basic-2000] rule permit source 1.1.1.1 0
[Router-acl-basic-2000] quit
```

# Create a class named **classifier_1**, and use ACL 2000 as the match criterion in the class.

```
[Router] traffic classifier classifier_1
[Router-classifier-classifier_1] if-match acl 2000
[Router-classifier-classifier_1] quit
```

# Create a behavior named **behavior_1**, and configure the traffic accounting action.

```
[Router] traffic behavior behavior_1
[Router-behavior-behavior_1] accounting
[Router-behavior-behavior_1] quit
```

# Create a policy named **policy**, and associate class **classifier_1** with behavior **behavior_1** in the policy.

```
[Router] qos policy policy
[Router-qospolicy-policy] classifier classifier_1 behavior behavior_1
[Router-qospolicy-policy] quit
```

# Apply the policy named **policy** to the incoming traffic of GigabitEthernet 1/0/1.

```
[Router] interface gigabitethernet 1/0/1
[Router-GigabitEthernet1/0/1] qos apply policy policy inbound
[Router-GigabitEthernet1/0/1] quit
```

# Display traffic statistics to verify the configuration.

```
[Router] display qos policy interface gigabitethernet 1/0/1

  Interface: GigabitEthernet1/0/1

  Direction: Inbound

  Policy: policy
   Classifier: classifier_1
     Operator: AND
```

```
Rule(s) : If-match acl 2000
Behavior: behavior_1
 Accounting Enable:
   28529 (Packets)
```

Configuring QPPB

Overview

The QoS Policy Propagation Through the Border Gateway Protocol (QPPB) feature enables you to classify IP packets based on BGP community lists, prefix lists, and BGP AS paths.

The idea of QPPB is that the BGP route sender pre-classifies routes before advertising them, and the BGP route receiver sets the IP precedence and QoS-local ID for the routes and takes appropriate QoS actions on the packets that match the routes.

QPPB minimizes the QoS policy configuration and management efforts on the BGP route receiver when the network topology changes. It is suitable for large-scaled complex network that classifies packets based on source or destination IP addresses for QoS.

QPPB applies to IBGP and EBGP. You can use it within an autonomous system or cross multiple autonomous systems.

QPPB fundamentals

QPPB works on the BGP receiver. It depends on the BGP route sender to pre-classify routes.

The BGP route sender uses a routing policy to set route attributes for BGP routes before advertising them.

The BGP receiver uses a routing policy to match routes based on these route attributes, and sets IP precedence and QoS-local ID for the matching routes:

1. Compares the routes with the incoming route policy based on their BGP AS path, prefix, or community attributes.
2. Applies the IP precedence and QoS-local ID to the matching routes.
3. Adds the BGP routes and their associated IP precedence and QoS-local ID to the routing table.
4. Applies the IP precedence and QoS-local ID to the packets sourced from or destined to the IP address in the route.
5. Takes QoS actions on the packets according to the QoS priority settings.

QPPB configuration task list

Complete the following tasks to configure QPPB:

Task		Remarks
Configuring the route sender	Configuring basic BGP functions	Required
	Creating a routing policy	Optional
Configuring the route receiver	Configuring basic BGP functions	Required
	Configuring a routing policy	Required
	Enabling QPPB on the route receiving interface	Required.
	Configuring a QoS policy	Required.
	Applying the QoS policy to an interface	Required.

Configuring the route sender

Configure the BGP route sender to set route attributes for routes before advertising them.

Configuring basic BGP functions

For more information, see *Layer 3—IP Routing Configuration Guide* and *Layer 3—IP Routing Command Reference*.

Creating a routing policy

Configure a routing policy to classify routes and set route attributes for the route classes. For more information, see *Layer 3—IP Routing Configuration Guide* and *Layer 3—IP Routing Command Reference*.

Configuring the route receiver

Configure the BGP route receiver to match the route attributes set by the router sender and set the QPPB-related attributes.

Configuring basic BGP functions

For more information, see *Layer 3—IP Routing Configuration Guide* and *Layer 3—IP Routing Command Reference*.

Configuring a routing policy

Configure a routing policy to match the route attributes set by the route sender and set the IP precedence, QoS-local ID, or both for the matching routes. For more information, see *Layer 3—IP Routing Configuration Guide* and *Layer 3—IP Routing Command Reference*.

Enabling QPPB on the route receiving interface

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	This interface must be the one connected to the route sender.
3. Enable QPPB on the interface.	**bgp-policy** { **destination** | **source** } { **ip-prec-map** | **ip-qos-map** } *	The command applies to only incoming traffic.

Configuring a QoS policy

The classes in the QoS policy use the IP precedence and QoS-local ID set by the routing policy as match criteria.

Applying the QoS policy to an interface

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Apply the specified policy to the interface.	**qos apply policy** *policy-name* { **inbound** | **outbound** }	On some cards, QoS policies can be applied but cannot take effect due to limited system resources. In this case, you can adjust related parameters (for example, reducing the number of queues) according to system prompt and then apply a QoS policy again.

QPPB configuration examples

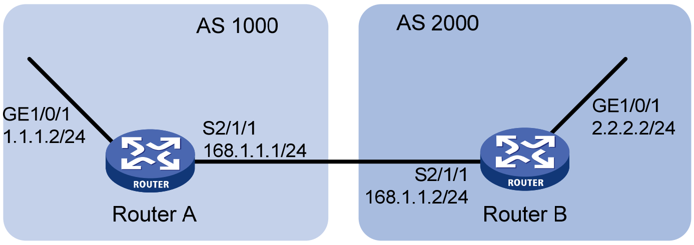
QPPB configuration example in an IPv4 network

Network requirements

As shown in Figure 36, all routers run BGP.

Configure QPPB, so that Router B can receive routes, set IP precedence and QoS-local IDs according to the routing policy, and use the QoS policy to limit the traffic rate to 512 kbps.

Figure 36 Network diagram



Configuration procedure

1. Configure IP addresses for each interface. (Details not shown.)
2. Configure Router A:

   # Configure a BGP connection to Router B, and add the network 1.1.1.0/8 to the BGP routing table.

   ```
   <RouterA> system-view
   [RouterA] bgp 1000
   [RouterA-bgp] peer 168.1.1.2 as-number 2000
   [RouterA-bgp] network 1.1.1.0 255.255.255.0
   [RouterA-bgp] quit
   ```
3. Configure Router B:

   # Configure a BGP connection to Router A, apply the routing policy **qppb** to routes from the peer 168.1.1.1, and add the network 2.2.2.0/8 to the BGP routing table.

   ```
   <RouterB> system-view
   [RouterB] bgp 2000
   ```

```
[RouterB-bgp] peer 168.1.1.1 as-number 1000
[RouterB-bgp] peer 168.1.1.1 route-policy qppb import
[RouterB-bgp] network 2.2.2.0 255.255.255.0
[RouterB-bgp] quit
```

# Configure the routing policy **qppb**.

```
[RouterB] route-policy qppb permit node 0
[RouterB-route-policy] apply ip-precedence 1
[RouterB-route-policy] apply qos-local-id 3
[RouterB-route-policy] quit
```

# Enable QPPB on interface Serial 2/1/1.

```
[RouterB] interface serial 2/1/1
[RouterB-Serial2/1/1] bgp-policy source ip-prec-map ip-qos-map
[RouterB-Serial2/1/1] quit
```

# Configure a QoS policy.

```
[RouterB] traffic classifier qppb
[RouterB-classifier-qppb] if-match ip-precedence 1
[RouterB-classifier-qppb] if-match qos-local-id 3
[RouterB-classifier-qppb] quit
[RouterB] traffic behavior qppb
[RouterB-behavior-qppb] car cir 512 green pass red discard
[RouterB-behavior-qppb] quit
[RouterB] qos policy qppb
[RouterB-qospolicy-qppb] classifier qppb behavior qppb
[RouterB-qospolicy-qppb] quit
```

# Apply QoS policy **qppb** to incoming traffic on interface Serial 2/1./1

```
[RouterB] interface serial 2/1/1
[RouterB-Serial2/1/1] qos apply policy qppb inbound
[RouterB-Serial2/1/1] quit
```

4. Verify the configuration:

# Check whether the related route on Router B takes effect.

```
[RouterB] display ip routing-table 1.1.1.0 24 verbose
Routing Table : Public
Summary Count : 1

  Destination: 1.1.1.0/24
     Protocol: BGP             Process ID: 0
   Preference: 255                   Cost: 0
 IpPrecedence: 1                  QosLcId: 3
      NextHop: 168.1.1.1        Interface: Serial 2/1/1
    BkNextHop: 0.0.0.0        BkInterface:
  RelyNextHop: 0.0.0.0          Neighbor : 168.1.1.1
    Tunnel ID: 0x0                  Label: NULL
        State: Active Adv GotQ        Age: 00h00m45s
      Tag: 0
```

# Display the QoS policy configuration on port Serial 2/1/1 of Router B.

```
[RouterB] display qos policy interface serial 2/1/1
  Interface: Serial2/1/1
  Direction: Inbound
```

```
    Policy: qppb
     Classifier: default-class
       Matched : 0(Packets) 0(Bytes)
       5-minute statistics:
         Forwarded: 0/0 (pps/bps)
         Dropped  : 0/0 (pps/bps)
       Rule(s) : If-match any
       Behavior: be
        -none-
     Classifier: qppb
       Matched : 0(Packets) 0(Bytes)
       5-minute statistics:
         Forwarded: 0/0 (pps/bps)
         Dropped  : 0/0 (pps/bps)
       Operator: AND
       Rule(s) : If-match ip-precedence 1
                 If-match qos-local-id 3
       Behavior: qppb
        Committed Access Rate:
          CIR 512 (kbps), CBS 32000 (byte), EBS 0 (byte)
          Green Action: pass
          Red Action: discard
          Green : 0(Packets) 0(Bytes)
          Red   : 0(Packets) 0(Bytes)
```

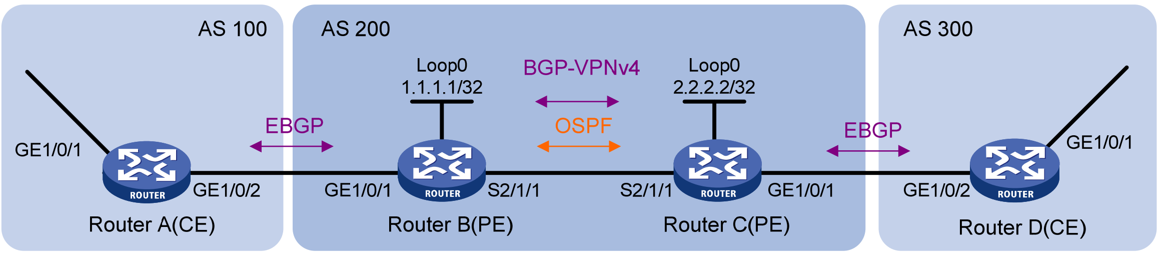
QPPB configuration example in an MPLS L3VPN

Network requirements

As shown in Figure 37, all routers run BGP.

Configure QPPB, so that Router C can receive routes, set the QPPB QoS-local IDs, and use the QoS policy to limit the traffic rate to 2 Mbps in each direction.

Figure 37 Network diagram



Device	Interface	IP address	Device	Interface	IP address
Router A	GE1/0/1	192.168.1.2/24	Router B	GE1/0/1	167.1.1.2/24
	GE1/0/2	167.1.1.1/24		S2/1/1	168.1.1.2/24
Router C	GE1/0/1	169.1.1.2/24	Router D	GE1/0/2	169.1.1.1/24
	S2/1/1	168.1.1.1/24		GE1/0/1	192.168.2.2/24

Configuration procedure

1. Configure IP addresses for each interface. (Details not shown.)

2. Configure Router A:

   # Configure a BGP connection.

```
<RouterA> system-view
[RouterA] bgp 100
[RouterA-bgp] peer 167.1.1.2 as-number 200
[RouterA-bgp] import-route direct
[RouterA-bgp] quit
```

3. Configure Router B:

   # Configure a VPN instance.

```
<RouterB> system-view
[RouterB] ip vpn-instance vpn1
[RouterB-vpn-instance-vpn1] route-distinguisher 200:1
[RouterB-vpn-instance-vpn1] vpn-target 200:1 export-extcommunity
[RouterB-vpn-instance-vpn1] vpn-target 200:1 import-extcommunity
[RouterB-vpn-instance-vpn1] quit
```

   # Configure a BGP connection.

```
[RouterB] router id 1.1.1.1
[RouterB] bgp 200
[RouterB-bgp] peer 2.2.2.2 as-number 200
[RouterB-bgp] peer 2.2.2.2 connect-interface LoopBack0
[RouterB-bgp] ipv4-family vpn-instance vpn1
[RouterB-bgp-vpn1] peer 167.1.1.2 as-number 100
[RouterB-bgp-vpn1] quit
[RouterB-bgp] ipv4-family vpnv4
[RouterB-bgp-af-vpnv4] peer 2.2.2.2 enable
[RouterB-bgp-af-vpnv4] quit
[RouterB-bgp] quit
```

   # Configure MPLS.

```
[RouterB] mpls lsr-id 1.1.1.1
[RouterB] mpls
[RouterB-mpls] quit
[RouterB] mpls ldp
[RouterB-mpls-ldp] quit
```

   # Configure OSPF.

```
[RouterB] ospf
[RouterB-ospf-1] area 0
[RouterB ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[RouterB ospf-1-area-0.0.0.0] network 168.1.1.0 0.0.0.255
[RouterB ospf-1-area-0.0.0.0] quit
[RouterB-ospf-1] quit
```

   # Bind interface GigabitEthernet 1/0/1 to VPN instance vpn1.

```
[RouterB] interface gigabitethernet 1/0/1
[RouterB-GigabitEthernet1/0/1] ip binding vpn-instance vpn1
[RouterB-GigabitEthernet1/0/1] ip address 167.1.1.2 24
[RouterB-GigabitEthernet1/0/1] quit
```

   # Enable MPLS on interface Serial 2/1/1.

```
[RouterB] interface serial 2/1/1
[RouterB-Serial2/1/1] mpls
```

```
[RouterB-Serial2/1/1] mpls ldp
[RouterB-Serial2/1/1] quit
```

4. Configure Router C:

# Configure a VPN instance.

```
<RouterC> system-view
[RouterC] ip vpn-instance vpn1
[RouterC-vpn-instance-vpn1] route-distinguisher 200:1
[RouterC-vpn-instance-vpn1] vpn-target 200:1 export-extcommunity
[RouterC-vpn-instance-vpn1] vpn-target 200:1 import-extcommunity
[RouterC-vpn-instance-vpn1] quit
```

# Configure a BGP connection.

```
[RouterC] router id 2.2.2.2
[RouterC] bgp 200
[RouterC-bgp] peer 1.1.1.1 as-number 200
[RouterC-bgp] peer 1.1.1.1 connect-interface LoopBack0
[RouterC-bgp] ipv4-family vpn-instance vpn1
[RouterC-bgp-vpn1] peer 169.1.1.1 as-number 300
[RouterC-bgp-vpn1] peer 169.1.1.1 route-policy qppb import
[RouterC-bgp-vpn1] quit
[RouterC-bgp] ipv4-family vpnv4
[RouterC-bgp-af-vpnv4] peer 1.1.1.1 enable
[RouterC-bgp-af-vpnv4] peer 1.1.1.1 route-policy qppb import
[RouterC-bgp-af-vpnv4] quit
[RouterC-bgp] quit
```

# Configure a routing policy.

```
[RouterC] route-policy qppb permit node 0
[RouterC-route-policy] apply qos-local-id 1023
[RouterC-route-policy] quit
```

# Configure MPLS.

```
[RouterC] mpls lsr-id 2.2.2.2
[RouterC] mpls
[RouterC-mpls] quit
[RouterC] mpls ldp
[RouterC-mpls-ldp] quit
```

# Configure OSPF.

```
[RouterC] ospf
[RouterC-ospf-1] area 0
[RouterC ospf-1-area-0.0.0.0] network 2.2.2.2 0.0.0.0
[RouterC ospf-1-area-0.0.0.0] network 168.1.1.0 0.0.0.255
[RouterC ospf-1-area-0.0.0.0] quit
[RouterC-ospf-1] quit
```

# Configure a QoS policy.

```
[RouterC] traffic classifier qppb
[RouterC-classifier-qppb] if-match qos-local-id 1023
[RouterC-classifier-qppb] quit
[RouterC] traffic behavior qppb
[RouterC-behavior-qppb] car cir 2000 green pass red discard
[RouterC-behavior-qppb] quit
```

```
[RouterC] qos policy qppb
[RouterC-qospolicy-qppb] classifier qppb behavior qppb
[RouterC-qospolicy-qppb] quit
```

# Enable MPLS on interface Serial 2/1/1.

```
[RouterC] interface serial 2/1/1
[RouterC-Serial2/1/1] mpls
[RouterC-Serial2/1/1] mpls ldp
```

# Enable QPPB on interfaces Serial 2/1/1 and GigabitEthernet 1/0/1.

```
[RouterC-Serial2/1/1] bgp-policy destination ip-qos-map
[RouterC-Serial2/1/1] quit
[RouterC] interface gigabitethernet 1/0/1
[RouterC-GigabitEthernet1/0/1] bgp-policy destination ip-qos-map
[RouterC-GigabitEthernet1/0/1] quit
```

# Bind interface GigabitEthernet 1/0/1 to VPN instance vpn1.

```
[RouterC] interface gigabitethernet 1/0/1
[RouterC-GigabitEthernet1/0/1] ip binding vpn-instance vpn1
[RouterC-GigabitEthernet1/0/1] ip address 169.1.1.2 24
```

# Apply QoS policy **qppb** to both the incoming and outgoing traffic of interface GigabitEthernet 1/0/1.

```
[RouterC-GigabitEthernet1/0/1] qos apply policy qppb inbound
[RouterC-GigabitEthernet1/0/1] qos apply policy qppb outbound
```

5. Configure Router D:

# Configure a BGP connection.

```
<RouterD> system-view
[RouterD] bgp 300
[RouterD-bgp] peer 169.1.1.2 as-number 200
[RouterD-bgp] import direct
[RouterD-bgp] quit
```

6. Verify the configuration:

# Check whether the related routes on Router A take effect.

```
[RouterA] display ip routing-table
Routing Tables: Public
         Destinations : 7        Routes : 7
Destination/Mask    Proto  Pre  Cost         NextHop         Interface
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
167.1.1.0/24        Direct 0    0            167.1.1.1       GE1/0/2
167.1.1.1/32        Direct 0    0            127.0.0.1       InLoop0
192.168.1.0/24      Direct 0    0            192.168.1.2     GE1/0/1
192.168.1.2/32      Direct 0    0            127.0.0.1       InLoop0
192.168.2.0/24      BGP    255  0            167.1.1.2       GE1/0/2
```

# Check whether the related routes on Router B take effect.

```
[RouterB] display ip routing-table
Routing Tables: Public
         Destinations : 6        Routes : 6
Destination/Mask    Proto  Pre  Cost         NextHop         Interface
1.1.1.1/32          Direct 0    0            127.0.0.1       InLoop0
2.2.2.2/32          OSPF   10   1            168.1.1.1       S2/1/1
```

```
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
168.1.1.0/24        Direct 0    0            168.1.1.2       S2/1/1
168.1.1.2/32        Direct 0    0            127.0.0.1       InLoop0
[RouterB] display ip routing-table vpn-instance vpn1
Routing Tables: vpn1
         Destinations : 6        Routes : 6
Destination/Mask    Proto  Pre  Cost         NextHop         Interface
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
167.1.1.0/24        Direct 0    0            167.1.1.2       GE1/0/1
167.1.1.2/32        Direct 0    0            127.0.0.1       InLoop0
192.168.1.0/24      BGP    255  0            167.1.1.1       GE1/0/1
192.168.2.0/24      BGP    255  0            2.2.2.2         NULL0
```

# Check whether the related routes on Router C take effect.

```
[RouterC] display ip routing-table
Routing Tables: Public
         Destinations : 6        Routes : 6
Destination/Mask    Proto  Pre  Cost         NextHop         Interface
1.1.1.1/32          OSPF   10   1            168.1.1.2       S2/1/1
2.2.2.2/32          Direct 0    0            127.0.0.1       InLoop0
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
168.1.1.0/24        Direct 0    0            168.1.1.1       S2/1/1
168.1.1.1/32        Direct 0    0            127.0.0.1       InLoop0
[RouterC] display ip routing-table vpn-instance vpn1
Routing Tables: vpn1
         Destinations : 6        Routes : 6
Destination/Mask    Proto  Pre  Cost         NextHop         Interface
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
169.1.1.0/24        Direct 0    0            169.1.1.2       GE1/0/1
169.1.1.2/32        Direct 0    0            127.0.0.1       InLoop0
192.168.1.0/24      BGP    255  0            1.1.1.1         NULL0
192.168.2.0/24      BGP    255  0            169.1.1.1       GE1/0/1
```

# Check whether the related routes on Router D take effect.

```
[RouterD] display ip routing-table
Routing Tables: Public
         Destinations : 7        Routes : 7
Destination/Mask    Proto  Pre  Cost         NextHop         Interface
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
169.1.1.0/24        Direct 0    0            169.1.1.1       GE1/0/2
169.1.1.1/32        Direct 0    0            127.0.0.1       InLoop0
192.168.1.0/24      BGP    255  0            169.1.1.2       GE1/0/2
192.168.2.0/24      Direct 0    0            192.168.2.2     GE1/0/1
192.168.2.2/32      Direct 0    0            127.0.0.1       InLoop0
```

# Display the QoS policy configuration on interface GigabitEthernet 1/0/1 of Router C.

```
[RouterC] display qos policy interface gigabitethernet 1/0/1
  Interface: GigabitEthernet1/0/1
  Direction: Inbound
  Policy: qppb
   Classifier: default-class
     Matched : 0(Packets) 0(Bytes)
     5-minute statistics:
       Forwarded: 0/0 (pps/bps)
       Dropped  : 0/0 (pps/bps)
     Rule(s) : If-match any
     Behavior: be
      -none-
   Classifier: qppb
     Matched : 0(Packets) 0(Bytes)
     5-minute statistics:
       Forwarded: 0/0 (pps/bps)
       Dropped  : 0/0 (pps/bps)
     Operator: AND
     Rule(s) : If-match qos-local-id 1023
     Behavior: qppb
      Committed Access Rate:
        CIR 2000 (kbps), CBS 125000 (byte), EBS 0 (byte)
        Green Action: pass
        Red Action: discard
        Green : 0(Packets) 0(Bytes)
        Red   : 0(Packets) 0(Bytes)
  Direction: Outbound
  Policy: qppb
   Classifier: default-class
     Matched : 0(Packets) 0(Bytes)
     5-minute statistics:
       Forwarded: 0/0 (pps/bps)
       Dropped  : 0/0 (pps/bps)
     Rule(s) : If-match any
     Behavior: be
      -none-
   Classifier:
     Matched : 0(Packets) 0(Bytes)
     5-minute statistics:
       Forwarded: 0/0 (pps/bps)
       Dropped  : 0/0 (pps/bps)
     Operator: AND
     Rule(s) : If-match qos-local-id 1023
     Behavior: qppb-l3vpn
      Committed Access Rate:
        CIR 2000 (kbps), CBS 125000 (byte), EBS 0 (byte)
        Green Action: pass
        Red Action: discard
        Green : 0(Packets) 0(Bytes)
        Red   : 0(Packets) 0(Bytes)
```

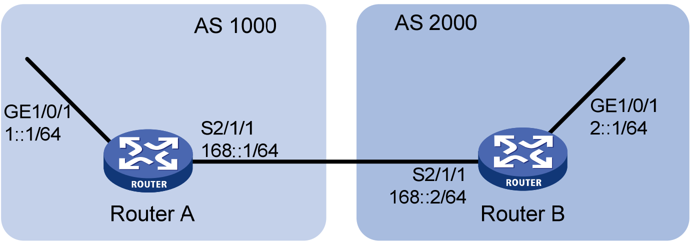
QPPB configuration example in an IPv6 network

Network requirements

As shown in Figure 38, all routers run BGP.

Configure QPPB, so that Router B can receive routes and set the QPPB IP precedence. Configure a QoS policy to limit the rate of traffic with the set IP precedence to 512 kbps.

Figure 38 Network diagram



Configuration procedure

1. Enable IPv6 globally, and configure IP addresses for each interface. (Details not shown.)
2. Configure Router A:

   # Configure BGP.

   ```
   <RouterA> system-view
   [RouterA] bgp 1000
   [RouterA-bgp] ipv6-family
   [RouterA-bgp-af-ipv6] peer 168::2 as-number 2000
   [RouterA-bgp-af-ipv6] network 1:: 64
   [RouterA-bgp-af-ipv6] quit
   [RouterA-bgp] quit
   ```
3. Configure Router B:

   # Configure BGP.

   ```
   <RouterB> system-view
   [RouterB] bgp 2000
   [RouterB-bgp] ipv6-family
   [RouterB-bgp-af-ipv6] peer 168::1 as-number 1000
   [RouterB-bgp-af-ipv6] peer 168::1 route-policy qppb import
   [RouterB-bgp-af-ipv6] network 2:: 64
   [RouterB-bgp-af-ipv6] quit
   [RouterB-bgp] quit
   ```

   # Configure a routing policy.

   ```
   [RouterB] route-policy qppb permit node 0
   [RouterB-route-policy] apply ip-precedence 4
   [RouterB-route-policy] quit
   ```

   # Enable QPPB on interface GigabitEthernet 1/0/1.

   ```
   [RouterB] interface gigabitethernet 1/0/1
   [RouterB-GigabitEthernet1/0/1] bgp-policy destination ip-prec-map
   ```

   # Configure a QoS policy.

```
[RouterB] traffic classifier qppb
[RouterB-classifier-qppb] if-match ip-precedence 4
[RouterB-classifier-qppb] quit
[RouterB] traffic behavior qppb
[RouterB-behavior-qppb] car cir 512 red discard
[RouterB-behavior-qppb] quit
[RouterB] qos policy qppb
[RouterB-qospolicy-qppb] classifier qppb behavior qppb mode qppb-manipulation
[RouterB-qospolicy-qppb] quit
```

# Apply the QoS policy to the incoming traffic of GigabitEthernet 1/0/1.

```
[RouterB] interface gigabitethernet 1/0/1
[RouterB-GigabitEthernet1/0/1] qos apply policy qppb inbound
[RouterB-GigabitEthernet1/0/1] quit
```

4. Verify the configuration:

# Check whether the related routes on Router A take effect.

```
[RouterA] display ipv6 routing-table
Routing Table :
         Destinations : 7        Routes : 7

Destination: ::1/128                                     Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 1::/64                                      Protocol  : Direct
NextHop    : 1::1                                        Preference: 0
Interface  : GE1/0/1                                      Cost      : 0

Destination: 1::1/128                                    Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 2::/64                                      Protocol  : BGP4+
NextHop    : 168::2                                      Preference: 255
Interface  : S2/1/1                                        Cost      : 0

Destination: 168::/64                                    Protocol  : Direct
NextHop    : 168::1                                      Preference: 0
Interface  : S2/1/1                                        Cost      : 0

Destination: 168::1/128                                  Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: FE80::/10                                   Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : NULL0                                       Cost      : 0
```

# Check whether the related routes on Router B take effect.

```
[RouterB] display ipv6 routing-table
```

```
Routing Table :
         Destinations : 7        Routes : 7

Destination: ::1/128                                     Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 1::/64                                      Protocol  : BGP4+
NextHop    : 168::1                                      Preference: 255
Interface  : S2/1/1                                        Cost      : 0

Destination: 2::/64                                      Protocol  : Direct
NextHop    : 2::1                                        Preference: 0
Interface  : GE1/0/1                                      Cost      : 0

Destination: 2::1/128                                    Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: 168::/64                                    Protocol  : Direct
NextHop    : 168::2                                      Preference: 0
Interface  : S2/1/1                                        Cost      : 0

Destination: 168::2/128                                  Protocol  : Direct
NextHop    : ::1                                         Preference: 0
Interface  : InLoop0                                     Cost      : 0

Destination: FE80::/10                                   Protocol  : Direct
NextHop    : ::                                          Preference: 0
Interface  : NULL0                                       Cost      : 0
```

# Display the QoS policy configuration information of GigabitEthernet 1/0/1 on Router B.

```
[RouterB] display qos policy interface gigabitethernet 1/0/1

  Interface: GigabitEthernet1/0/1

  Direction: Inbound

  Policy: qppb
   Classifier: default-class
     Matched : 0(Packets) 0(Bytes)
     5-minute statistics:
       Forwarded: 0/0 (pps/bps)
       Dropped  : 0/0 (pps/bps)
     Rule(s) : If-match any
     Behavior: be
      -none-
   Classifier: qppb
     Matched : 0(Packets) 0(Bytes)
```

```
5-minute statistics:
  Forwarded: 0/0 (pps/bps)
  Dropped  : 0/0 (pps/bps)
Operator: AND
Rule(s) : If-match ip-precedence 4
Behavior: qppb
 Committed Access Rate:
   CIR 512 (kbps), CBS 125000 (byte), EBS 0 (byte)
   Green Action: pass
   Red Action: discard
   Green : 0(Packets) 0(Bytes)
   Red   : 0(Packets) 0(Bytes)
```

Appendix

Appendix A Acronyms

Table 5 Acronyms

Acronym	Full spelling
AF	Assured Forwarding
BE	Best Effort
BQ	Bandwidth Queuing
CAR	Committed Access Rate
CBS	Committed Burst Size
CBQ	Class Based Queuing
CBWFQ	Class Based Weighted Fair Queuing
CE	Customer Edge
CIR	Committed Information Rate
CQ	Custom Queuing
DAR	Deeper Application Recognition
DCBX	Data Center Bridging Exchange Protocol
DiffServ	Differentiated Service
DoS	Denial of Service
DSCP	Differentiated Services Code Point
EBS	Excess Burst Size
EF	Expedited Forwarding
FIFO	First in First out
FQ	Fair Queuing
GTS	Generic Traffic Shaping
IntServ	Integrated Service
ISP	Internet Service Provider
LLQ	Low Latency Queuing
LSP	Label Switched Path
P2P	Peer-to-Peer
MPLS	Multiprotocol Label Switching
PE	Provider Edge
PIR	Peak Information Rate
PQ	Priority Queuing
QoS	Quality of Service
QPPB	QoS Policy Propagation Through the Border Gateway Protocol

Acronym	Full spelling
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RTP	Real-Time Transport Protocol
SP	Strict Priority
TE	Traffic Engineering
ToS	Type of Service
VoIP	Voice over IP
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin

Appendix B Default priority mapping tables

For the default **dscp-dscp** priority mapping table, an input value yields a target value equal to it.

Table 6 Default dot1p-lp and dot1p-dp priority mapping tables

Input priority value	dot1p-lp mapping	dot1p-dp mapping
802.1p priority (dot1p)	**Local precedence (lp)**	**Drop precedence (dp)**
0	2	0
1	0	0
2	1	0
3	3	0
4	4	0
5	5	0
6	6	0
7	7	0

Table 7 Default dscp-dp and dscp-dot1p priority mapping tables

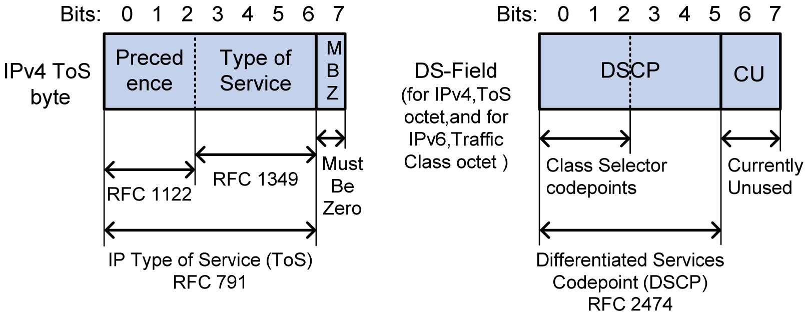
Input priority value	dscp-dp mapping	dscp-dot1p mapping
DSCP	**Drop precedence (dp)**	**802.1p priority (dot1p)**
0 to 7	0	0
8 to 15	0	1
16 to 23	0	2
24 to 31	0	3
32 to 39	0	4
40 to 47	0	5
48 to 55	0	6

Input priority value	dscp-dp mapping	dscp-dot1p mapping
56 to 63	0	7

Appendix C Introduction to packet precedences

IP precedence and DSCP values

Figure 39 ToS and DS fields


Bits: 0 1 2 3 4 5 6 7
IPv4 ToS byte
Preced ence
Type of Service
M B Z
RFC 1122
RFC 1349
Must Be Zero
IP Type of Service (ToS) RFC 791
Bits: 0 1 2 3 4 5 6 7
DS-Field (for IPv4,ToS octet,and for IPv6,Traffic Class octet)
DSCP
CU
Class Selector codepoints
Currently Unused
Differentiated Services Codepoint (DSCP) RFC 2474

As shown in Figure 39, the ToS field in the IPv4 header contains 8 bits, where the first 3 bits (0 to 2) represent IP precedence from 0 to 7. The Traffic Classes field in the IPv6 header contains 8 bits, where the first 3 bits (0 to 2) represent IP precedence from 0 to 7. According to RFC 2474, the ToS field in the IPv4 header or the Traffic Classes field in the IPv6 header is redefined as the DS field, where a DSCP value is represented by the first 6 bits (0 to 5) and is in the range of 0 to 63. The remaining 2 bits (6 and 7) are reserved.

Table 8 IP precedence

IP precedence (decimal)	IP precedence (binary)	Description
0	000	Routine
1	001	priority
2	010	immediate
3	011	flash
4	100	flash-override
5	101	critical
6	110	internet
7	111	network

Table 9 DSCP values

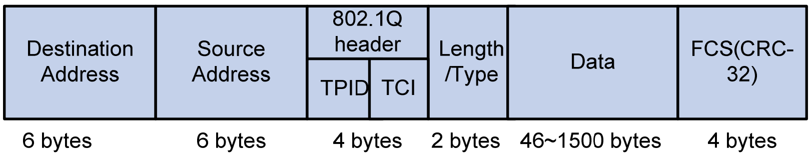
DSCP value (decimal)	DSCP value (binary)	Description
46	101110	ef
10	001010	af11
12	001100	af12
14	001110	af13
18	010010	af21

DSCP value (decimal)	DSCP value (binary)	Description
20	010100	af22
22	010110	af23
26	011010	af31
28	011100	af32
30	011110	af33
34	100010	af41
36	100100	af42
38	100110	af43
8	001000	cs1
16	010000	cs2
24	011000	cs3
32	100000	cs4
40	101000	cs5
48	110000	cs6
56	111000	cs7
0	000000	be (default)

802.1p priority

802.1p priority lies in the Layer 2 header and applies to occasions where Layer 3 header analysis is not needed and QoS must be assured at Layer 2.

Figure 40 An Ethernet frame with an 802.1Q tag header



As shown in Figure 40, the 4-byte 802.1Q tag header consists of the TPID (2 bytes in length), whose value is 0x8100, and the TCI (2 bytes in length). Figure 41 shows the format of the 802.1Q tag header. The Priority field in the 802.1Q tag header is called the "802.1p priority," because its use is defined in IEEE 802.1p. Table 10 shows the values for 802.1p priority.

Figure 41 802.1Q tag header

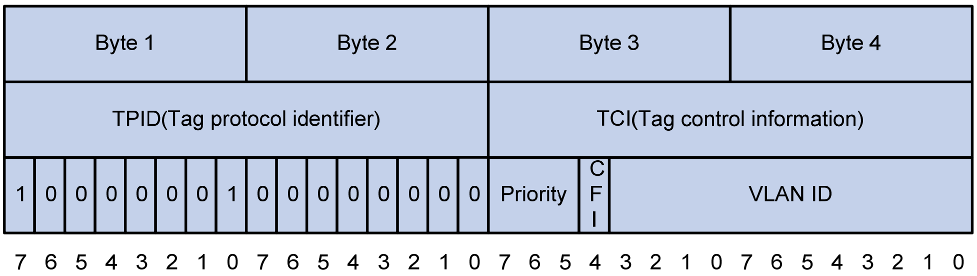


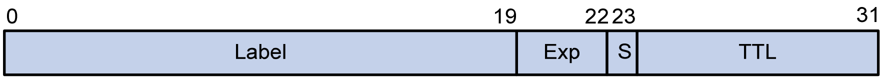
Table 10 Description on 802.1p priority

802.1p priority (decimal)	802.1p priority (binary)	Description
0	000	best-effort
1	001	background
2	010	spare
3	011	excellent-effort
4	100	controlled-load
5	101	video
6	110	voice
7	111	network-management

EXP values

The EXP field is in MPLS labels for MPLS QoS purposes.

Figure 42 MPLS label structure



As shown in Figure 42, the EXP field is 3 bits long and is in the range of 0 to 7.

Configuring MPLS QoS

The MPLS-related knowledge is necessary for understanding MPLS QoS. For more information about MPLS, see *MPLS Configuration Guide*. For more information about EXP precedence, see "Configuring priority mapping." For more information about traffic policing, see "Configuring traffic policing, traffic shaping, and line rate." For more information about priority marking, see "Configuring priority marking." For more information about congestion management, see "Configuring congestion management."

Overview

In the area of QoS, to provide the support for DiffServ as IP does, MPLS uses 3 bits analogous to IP precedence, called "EXP bits," to carry class-of-service information. With the EXP bits, MPLS QoS is achieved to identify different traffic flows and implement differentiated services, guaranteeing low delay and low packet loss ratio for voice and video traffic.

MPLS QoS supports Committed Access Rate (CAR), priority marking, and congestion management. MPLS QoS provides the following functions:

- Classify traffic on the PE to apply differentiated QoS strategies for different traffic classes. For example, MPLS QoS can organize packets with EXP value 1 into a class and packets with EXP value 2 into another class, and then perform traffic policing and priority marking for each class of packets.
- When a PE labels a packet, it maps the IP precedence to the EXP field of the label. In this way, the class information carried in the IP header is carried in the label.
- Differentiated dispatching (such as PQ, WFQ, or CBQ) is performed between a P device and a PE according to the EXP field to provide differentiated QoS for labeled traffic on an LSP.

The EXP field in an MPLS label is processed following these rules:

- Any QoS-capable device can reset the EXP field of the topmost label.
- During label encapsulation, the ToS field of the IP packet is directly changed into the EXP field of the MPLS label.
- The EXP field remains unchanged when label swapping is performed.
- During a label push operation, the EXP field of the newly pushed outer label inherits the EXP field of the inner label.
- After a label pop operation, if the packet is still an MPLS packet, the EXP field of the popped label is not copied to the inner label; if the packet is an IP packet, the EXP field of the popped label is not copied to the ToS field of the IP packet.

Configuring MPLS CAR

By configuring CAR for traffic entering an MPLS network, you can limit the transmission rate to avoid network congestion, and in addition, mark priority for the traffic.

Before you configure MPLS CAR, complete basic MPLS configurations. For more information about basic MPLS configurations, see *MPLS Configuration Guide*.

To configure MPLS CAR:

Step	Command
1. Enter system view.	**system-view**

Step	Command
2. Enter interface view.	**interface** *interface-type interface-number*
3. Configure an MPLS CAR policy for the interface or port group.	**qos car** { **inbound** | **outbound** } { **any | acl** *acl-numbe*r | **carl** *carl-index* } **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action*] [**red** *action*]

The *action* argument for MPLS can be as follows:

- **remark-mpls-exp-continue** *new-exp*—Sets the EXP value to *new-exp* and continues to process the packet using the next CAR policy. The value range for *new-exp* argument is 0 to 7.
- **remark-mpls-exp-pass** *new-exp*—Sets the EXP value to *new-exp* and permits the packet to pass through. The value range for *new-exp* is 0 to 7.

Configuring MPLS priority marking

In an MPLS network, you can adjust the priority of an MPLS traffic flow by re-marking its EXP value.

Before you configure MPLS priority marking, complete basic MPLS configurations. For more information about basic MPLS configurations, see *MPLS Configuration Guide*.

To configure MPLS priority marking:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic class and enter traffic class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	The *classifier-name* argument cannot be the name of any system-defined traffic class. The default operator for the match criteria configured in a traffic class is **and**.
3. Configure a match criterion for the traffic class.	**if-match** [**not**] **mpls-exp** *exp-value-list*	The match criterion applies only to MPLS packets.
4. Return to system view.	**quit**	N/A
5. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	N/A
6. Configure an EXP re-marking action in the behavior.	**remark mpls-exp** *exp-value*	N/A
7. Return to system view.	**quit**	N/A
8. Create a QoS policy and enter QoS policy view.	**qos policy** *policy-name*	N/A
9. Associate the traffic class with the traffic behavior in the QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A

Step	Command	Remarks
10. Return to system view.	**quit**	N/A
11. Enter interface view or port group view.	**interface** *interface-type interface-number*	N/A
12. Apply the QoS policy to the interface or port group.	**qos apply policy** *policy-name* { **inbound** | **outbound** }	N/A

Configuring MPLS congestion management

By configuring MPLS congestion management, you can assign packets exceeding the bandwidth to the queues by priority, and then send these packets according to a certain queue scheduling mechanism, avoiding dropping packets directly.

You can configure PQ and CQ for MPLS.

Configuration prerequisites

Complete basic MPLS configurations. For more information about basic MPLS configurations, see *MPLS Configuration Guide*.

Configure MPLS PQ

Step	Command
1. Enter system view.	**system-view**
2. Configure a PQ list.	**qos pql** *pql-index* **protocol mpls exp** *exp-value-list* **queue** { **bottom** | **middle** | **normal** | **top** }
3. Enter interface view.	**interface** *interface-type interface-number*
4. Apply the PQ list to the interface.	**qos pq pql** *pql-index*

Configure MPLS CQ

Step	Command
1. Enter system view.	**system-view**
2. Configure an EXP-based CQ list.	**qos cql** *cql-index* **protocol mpls exp** *exp-value-list* **queue** *queue-number*
3. Enter interface view.	**interface** *interface-type interface-number*
4. Apply the CQ list to the interface.	**qos cq cql** *cql-index*

MPLS QoS configuration example

Network requirements

As shown in Figure 43:

- Both CE 1 and CE 2 belong to VPN 1.
- The bandwidth of the link between PE 1 and P is 2 M.
- The bandwidth of the link between PE 2 and P is 2 M.

Provide differentiated QoS services for flows with different precedence values in VPN 1.

The configuration in this example involves the following parts:

First, configure MPLS VPN on CE 1, PE 1, P, PE 2, and CE 2 as follows:

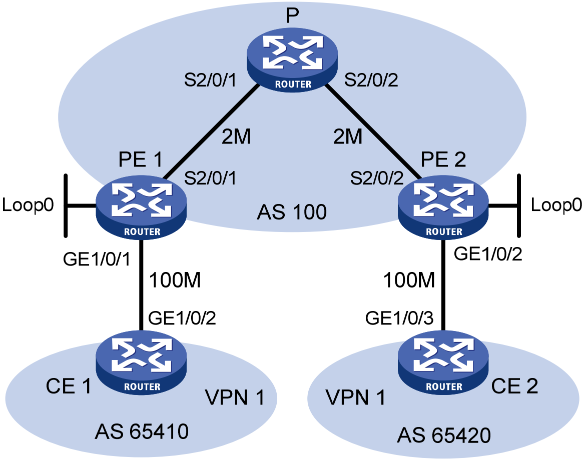
- Run OSPF between PE 1 and P, and between PE 2 and P.
- Form a MP-EBGP neighborship between PE and CE.
- Form a MP-IBGP neighborship between PE and PE.

Second, configure MPLS QoS on PE 1 and P as follows:

- Configure a QoS policy on the incoming interface GigabitEthernet 1/0/1 on PE 1 and set the EXP field value for an MPLS packet according to the DSCP attribute of the MPLS packets.
- On the device P, classify traffic on the basis of the EXP field and configure flow-based CBQ: guarantee 10% of the bandwidth for traffic with an EXP value of 1, guarantee 20% of the bandwidth for traffic with an EXP value of 2, guarantee 30% of the bandwidth for traffic with an EXP value of 3, and guarantee a low delay and 40% of the bandwidth for traffic with an EXP value of 4.

For the MPLS VPN configuration, see *MPLS Configuration Guide*. This section introduces only the MPLS QoS configuration.

Figure 43 Network diagram



Device	Interface	IP address	Device	Interface	IP address
CE 1	GE 1/0/2	10.1.1.2/24	CE 2	GE 1/0/3	10.2.1.2/24
PE 1	GE 1/0/1	10.1.1.1/24	PE 2	GE 1/0/2	10.2.1.1/24
	S 2/0/1	12.1.1.1/24		S 2/0/2	12.2.1.1/24
	Loop0	1.1.1.1/32		Loop0	1.1.1.2/32
P	S 2/0/1	12.1.1.2/24			
	S 2/0/2	12.2.1.2/24			

Configuration procedure

1. Configure device PE 1:

   # Configure four classes to match the DSCP values AF11, AF21, AF31, and EF of the MPLS packets in the same VPN.

   ```
   <PE1> system-view
   [PE1] traffic classifier af11
   [PE1-classifier-af11] if-match dscp af11
   [PE1-classifier-af11] traffic classifier af21
   [PE1-classifier-af21] if-match dscp af21
   [PE1-classifier-af21] traffic classifier af31
   [PE1-classifier-af31] if-match dscp af31
   [PE1-classifier-af31] traffic classifier efclass
   [PE1-classifier-efclass] if-match dscp ef
   [PE1-classifier-efclass] quit
   ```

   # Configure four traffic behaviors to set the EXP value to 1, 2, 3, and 4, respectively, for MPLS packets.

   ```
   [PE1] traffic behavior exp1
   [PE1-behavior-exp1] remark mpls-exp 1
   [PE1-behavior-exp1] traffic behavior exp2
   [PE1-behavior-exp2] remark mpls-exp 2
   [PE1-behavior-exp2] traffic behavior exp3
   [PE1-behavior-exp3] remark mpls-exp 3
   [PE1-behavior-exp3] traffic behavior exp4
   [PE1-behavior-exp4] remark mpls-exp 4
   [PE1-behavior-exp4] quit
   ```

   # Create QoS policy **REMARK**, and associate the behaviors with the classes in the QoS policy to mark different classes of packets with different EXP values.

   ```
   [PE1] qos policy REMARK
   [PE1-qospolicy-REMARK] classifier af11 behavior exp1
   [PE1-qospolicy-REMARK] classifier af21 behavior exp2
   [PE1-qospolicy-REMARK] classifier af31 behavior exp3
   [PE1-qospolicy-REMARK] classifier efclass behavior exp4
   [PE1-qospolicy-REMARK] quit
   ```

   # Apply QoS policy **REMARK** to the incoming traffic of interface GigabitEthernet 1/0/1 of PE1 in the MPLS network.

   ```
   [PE1] interface gigabitethernet 1/0/1
   [PE1-GigabitEthernet1/0/1] qos apply policy REMARK inbound
   [PE1-GigabitEthernet1/0/1] quit
   ```

2. Configure device P:

   # Configure four classes to match EXP values 1, 2, 3 and 4 of the MPLS packets, respectively.

   ```
   <P> system-view
   [P] traffic classifier EXP1
   [P-classifier-EXP1] if-match mpls-exp 1
   [P-classifier-EXP1] traffic classifier EXP2
   [P-classifier-EXP2] if-match mpls-exp 2
   [P-classifier-EXP2] traffic classifier EXP3
   [P-classifier-EXP3] if-match mpls-exp 3
   [P-classifier-EXP3] traffic classifier EXP4
   ```

```
[P-classifier-EXP4] if-match mpls-exp 4
[P-classifier-EXP4] quit
```

# Create four traffic behaviors and configure AF or EF actions for them.

```
[P] traffic behavior AF11
[P-behavior-AF11] queue af bandwidth pct 10
[P-behavior-AF11] traffic behavior AF21
[P-behavior-AF21] queue af bandwidth pct 20
[P-behavior-AF21] traffic behavior AF31
[P-behavior-AF31] queue af bandwidth pct 30
[P-behavior-AF31] traffic behavior EF
[P-behavior-EF] queue ef bandwidth pct 40
[P-behavior-EF] quit
```

# Create QoS policy **QUEUE**, and associate the behaviors with the classes to meet the following requirements: guarantee 10% of the bandwidth for traffic with an EXP value of 1, guarantee 20% of the bandwidth for traffic with an EXP value of 2, guarantee 30% of the bandwidth for traffic with an EXP value of 3, and guarantee a low delay and 40% of the bandwidth for traffic with an EXP value of 4.

```
[P] qos policy QUEUE
[P-qospolicy-QUEUE] classifier EXP1 behavior AF11
[P-qospolicy-QUEUE] classifier EXP2 behavior AF21
[P-qospolicy-QUEUE] classifier EXP3 behavior AF31
[P-qospolicy-QUEUE] classifier EXP4 behavior EF
[P-qospolicy-QUEUE] quit
```

# Apply QoS policy **QUEUE** to the outgoing traffic of Serial 2/0/2 on device P.

```
[P] interface serial 2/0/2
[P-Serial2/0/2] qos apply policy QUEUE outbound
```

After the configuration, when congestion occurs in VPN 1, the bandwidth proportion between flows with the DSCP value being af11, af21, af31, and ef is 1:2:3:4, and the delay for the flow with the DSCP value being ef is smaller than the other traffic flows.

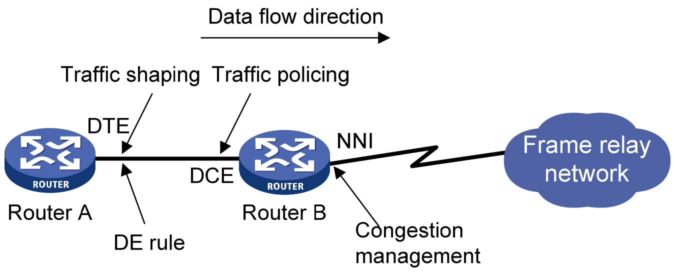
Configuring FR QoS

Overview

On a FR interface, you can use generic QoS services to perform traffic policing, traffic shaping, congestion management, and congestion avoidance. You can also use FR-specific QoS mechanisms, including FR traffic shaping, FR traffic policing, FR congestion management, FR discard eligibility (DE) rule list, and FR queuing management.

FR QoS is more flexible than generic QoS. It works on a per PVC basis, and generic QoS works on a per interface basis. For more information about Frame Relay, see *Layer 2—WAN Configuration Guide*.

Figure 44 FR QoS implementation



FR QoS uses these parameters:

- **CIR ALLOW**—Average transmission rate guaranteed on a VC. When no congestion occurs, CIR ALLOW is guaranteed for data transmission.
- **CIR**—Minimum transmitting rate that an FR VC provides. CIR is guaranteed for data transmission even if congestion occurs in the network.
- **CBS**—Traffic that an FR VC is committed to transmit within the interval of Tc. When congestion occurs in the network, transmitting traffic of the CBS size is guaranteed by the FR network.
- **EBS**—Maximum traffic that can exceed CBS on an FR VC within the interval of Tc. When congestion occurs in the network, the traffic of EBS is dropped first. Transmitting traffic of the EBS size is not guaranteed by the FR network.

FRTS

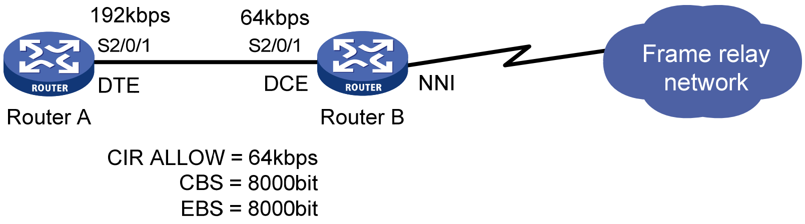
The functionality of FRTS

Frame relay traffic shaping (FRTS) limits the outgoing traffic rate and smoothes bursts for PVCs so they can transmit traffic at a nearly constant rate.

FRTS applies to the outgoing interface of a switch. You can use FRTS to remove the bottleneck created when the input rate of a device is slower than the output rate of the sending device.

As shown in Figure 45, Router B transmits packets to Router A at 128 kbps, whereas the maximum interface rate of Router A is only 64 kbps. This traffic rate disparity creates a bottleneck at the interface that connects Router A to the FR network. To avoid packet loss, you can use FRTS at the outgoing interface Serial 2/0/1 of Router B so the interface can transmit packets constantly at 64 kbps when no congestion is present. Even if congestion occurs in the network, Router B can still transmit packets at the rate of 32 kbps.

Figure 45 FRTS implementation

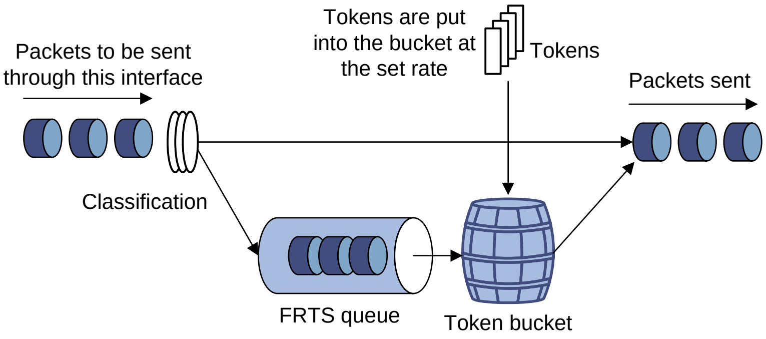


FRTS uses the parameters CIR ALLOW, CIR, CBS, and EBS for traffic shaping. FR PVCs can transmit packets at the rate of CIR ALLOW. In case of bursty packets, FRTS allows an FR PVC to transmit packets at a rate exceeding CIR ALLOW.

How FRTS works

FRTS is implemented using token buckets. The meanings of the related parameters in the protocol are modified as required by the actual algorithm and principles. See Figure 46 for how a token bucket works.

Figure 46 How a token bucket works


Tokens are put into the bucket at the set rate
Tokens
Packets to be sent through this interface
Packets sent
Classification
FRTS queue
Token bucket

In the token bucket approach, packets requiring traffic control are put into the token bucket for processing before transmission. If enough tokens are available in the token bucket for sending these packets, the packets are allowed to pass. If the number of tokens in the token bucket is not enough for sending these packets, these packets are put into the FR class queue (the FRTS queue in FRTS implementation). Once enough tokens are available in the token bucket, the packets are taken out of the FR class queue for transmission. In this way, you can control the traffic of a certain class of packets. Tokens are in the unit of bits.

The FR protocol-provisioned related parameters correspond to the FRTS parameters as follows:

- The sum of CBS and EBS equals the token bucket size.
- CIR ALLOW defines the number of tokens put into the token bucket per second.

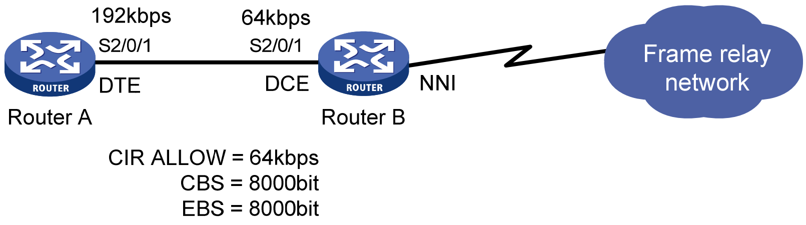
For efficiency, the FRTS introduces the concept of dynamic Tc. Tc (Tc=size of packet/CIR ALLOW) allows for dynamic adjustment depending on the transmitted packet size. The device allocates the required tokens to the current packets waiting for transmission within the latest Tc regardless of the packet size (which is smaller than 1500 bytes).

Take sending an 800-byte packet for example. Given the CIR ALLOW of 64000 kbps, it takes Tc=6400/64000=0.1s (100ms) to put the required tokens into the token bucket. The packet is transmitted successfully after 6400 bits of tokens are put into the token bucket within 100 ms.

FR traffic policing

FR traffic policing monitors the traffic entering the network from each PVC and restricts the traffic within a permitted range. If the traffic on a PVC exceeds the user-defined threshold, the device takes some measures, like packet drop, to protect the network resources.

Figure 47 FR traffic policing implementation



As shown in Figure 47, Router A at the user side transmits packets at the rate of 192 kbps to Router B at the switching side. However, Router B only wants to provide the bandwidth of 64 kbps for Router A. Therefore, you must configure FR traffic policing at the DCE side of Router B.

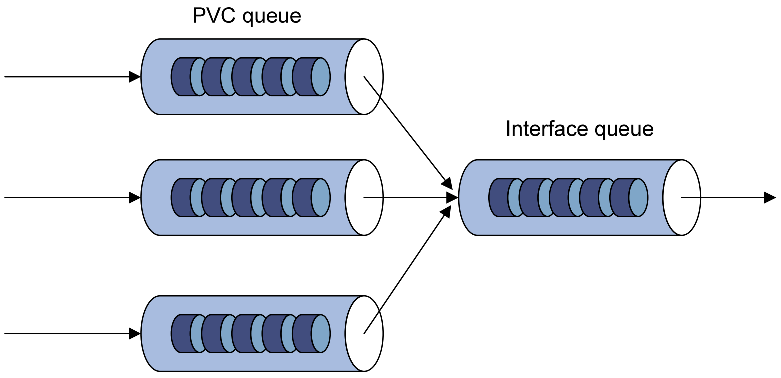
FR traffic policing can only be applied to the DCE interface of a device. FR traffic policing can monitor the traffic transmitted from the DTE side. When the traffic is smaller than CBS, the packets can be transmitted, and the device does not process the packets. When the traffic is larger than CBS and smaller than EBS + CBS, the packets can be transmitted. However, as for those packets of the traffic exceeding CBS, the device sets the DE flag bits in the FR packet headers to 1. When the traffic is larger than CBS + EBS, the device transmits the traffic of CBS + EBS and drops the traffic exceeding CBS + EBS. As for the traffic exceeding CBS, the device sets the DE flag bits in the FR packet headers to 1.

FR queuing

Besides FR PVC queues, FR interfaces also have interface queues. With FRTS disabled, only FR interface queues take effect, and the predefined FR PVC queues take effect only in the case that FRTS is enabled.

The relationship between PVC queues and interface queues is shown in Figure 48.

Figure 48 FR queuing



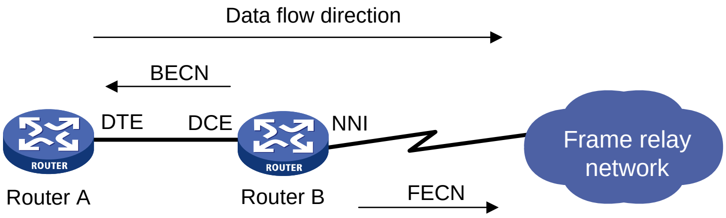
For 6600/HSR6600 routers, only FIFO queuing is available on FR interfaces.

FR congestion management

FR congestion management can process FR packets when congestion occurs in the network. It drops the packets with the DE flag bits set to 1 and notifies other devices on the network about the congestion.

FR congestion management is applied on the outgoing interface of an FR switching device. If no congestion occurs, the FR switching device forwards the FR packets without any processing. If congestion occurs, packets with the FE flag bits set to 1 are dropped. As for forward packets to be forwarded, the FECN flag bits in the FR packet headers are set to 1. As for backward packets on the same PVC, the BECN flag bits in the FR packet headers are set to 1. If no backward packets are transmitted within a period, the device automatically transmits the packets with the BECN flag bits set to 1 to the calling DTE.

Figure 49 FR congestion management implementation



FR DE rule list

In an FR network, packets with the DE flag bits set to 1 are dropped first when congestion occurs in the network. DE rule lists are applied on the FR PVCs of a device, with each DE rule list containing multiple DE rules. If a packet transmitted over the PVC matches the rules in the DE rule list, its DE flag bit is set to 1. The packet is dropped first when congestion occurs in the network.

FR QoS configuration task list

Task	Remarks
Creating and configuring an FR class	Required.
Configuring FRTS	Optional.
Configuring FR traffic policing	Optional.
Configuring FR congestion management	Optional.
Configuring FR DE rule list	Optional.
Configuring FR fragmentation	Optional.

Creating and configuring an FR class

An FR class specifies a set of QoS parameters for traffic control and regulation on FR PVCs. To provide QoS on a PVC, create an FR class, set QoS parameters such as FRTS settings in the class, and map the class to the PVC.

You can map an FR class to PVCs by mapping it to FR interfaces or DLCIs.

- The FR class mapped to an FR interface takes effect on all PVCs on the interface.

- The FR class mapped to a DLCI takes effect only on the PVC identified by the DLCI.

An QoS-capable FR PVC selects an FR class in the following order:

- The FR class mapped to the DLCI
- The FR class mapped to the FR interface

To configure and create an FR class:

Step		Command	Remarks
1. Enter system view.		**system-view**	N/A
2. Create an FR class and enter FR class view.		**fr class** *class-name*	By default, no FR class is created.
3. Return to system view.		**quit**	N/A
4. Map the FR class to an FR interface or DLCI.	(Method 1) Map the FR class to an FR interface.	**a.** Enter FR interface view: **interface** *interface-type interface-number* **b.** Map the FR class to the FR interface: **fr-class** *class-name*	Use either method or both methods. By default, no FR class is mapped to any FR interface or DLCI.
	(Method 2) Map the FR class to an DLCI.	**c.** Enter FR interface view: **interface** *interface-type interface-number* **d.** Enter FR PVC view: **fr dlci** *dlci-number* **e.** Map the FR class to the DLCI: **fr-class** *class-name*	

In FR class view, you can configure QoS parameters for QoS services such as FRTS, FR traffic policing, FR congestion management, and FR queuing. For more information about the parameter configurations, see the subsequent sections.

Configuring FRTS

Configuration restrictions and guidelines

- FRTS applies to the outgoing FR packets. It is typically applied on DTE.
- You can use the **cbs**, **ebs**, and **cir allow** commands to set both inbound and outbound parameters for FR PVCs. However, only outbound parameters take effect for FRTS.
- To ensure large-size packets can pass through, set CBS less than CIR ALLOW.

Configuration procedure

To configure FRTS:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter FR interface view.	**interface** *interface-type interface-number*	N/A

Step	Command	Remarks
3. Enable FRTS.	**fr traffic-shaping**	By default, FRTS is disabled.
4. Return to system view.	**quit**	N/A
5. Enter FR class view.	**fr class** *class-name*	N/A
6. Set CBS for FR PVCs.	**cbs** [**outbound**] *committed-burst-size*	Optional. The default setting is 56000 bps.
7. Set EBS for FR PVCs.	**ebs** [**outbound**] *excess-burst-size*	Optional. The default setting is 0 bit.
8. Set CIR ALLOW for FR PVCs.	**cir allow** [**outbound**] *committed-information-rate*	Optional. The default setting is 56000 bps.
9. Set CIR for FR PVCs.	**cir** *committed-information-rate*	Optional. The default setting is 56000 bps.
10. Enable FRTS adaptation.	**traffic-shaping adaptation** { **becn** *percentage* | **interface-congestion** *number* }	Optional. By default, the command is enabled with the *percentage* argument being 25 for traffic with the BECN flag.

Configuring FR traffic policing

Configuration restrictions and guidelines

- FR traffic policing is applied to the interfaces receiving FR packets and can only be applied to the DCE of an FR network.
- You can use the **cbs**, **ebs**, and **cir allow** commands to set both inbound and outbound parameters for FR PVCs. However, only inbound parameters take effect for FR traffic policing.

Configuration procedure

To configure FR traffic policing:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter FR interface view.	**interface** *interface-type interface-number*	N/A
3. Enable FR traffic policing.	**fr traffic-policing**	By default, FR traffic policing is disabled.
4. Return to system view.	**quit**	N/A
5. Enter FR class view.	**fr class** *class-name*	N/A
6. Set CBS for FR PVCs.	**cbs** [**inbound**] *committed-burst-size*	Optional. The default setting is 56000 bps.
7. Set EBS for FR PVCs.	**ebs** [**inbound**] *excess-burst-size*	Optional. The default setting is 0 bit.

Step	Command	Remarks
8. Set CIR ALLOW for FR PVCs.	**cir allow** [**inbound**] *committed-information-rate*	Optional. The default setting is 56000 bps.

Configuring FR congestion management

FR congestion management includes congestion management on the FR interface and congestion management on the FR PVC. You can set the congestion thresholds in FR PVC view or FR interface view for a specific FR class.

Configuring FR congestion management for an FR interface

The device determines whether congestion occurs based on the percentage of the current FR interface queue length to the total interface queue length. If the percentage exceeds the set congestion threshold, the device considers congestion has occurred and takes action on packets (for example, drops packets), to alleviate the condition.

To configure FR congestion management for an FR interface:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter FR interface view.	**interface** *interface-type interface-number*	N/A
3. Enable FR congestion management on the FR interface.	**fr congestion-threshold** { **de** | **ecn** } *queue-percentage*	By default, FR congestion management is disabled for an FR interface.

Configuring FR congestion management for an FR PVC

The device determines whether congestion has occurred based on the percentage of the current FR PVC queue length to the total interface queue length. If the percentage exceeds the set congestion threshold, the device considers that congestion has occurred and takes action on packets (for example, drops packets), to alleviate the condition.

When you configure FR congestion management for an FR PVC, follow these guidelines:

- With FR congestion management enabled, an FR interface performs only FIFO queuing or PVC PQ.
- With FR congestion management enabled, an FR PVC performs only FIFO queuing.
- To use congestion management on an FR PVC, make sure that FRTS has been enabled on the main interface of the FR PVC.

To configure FR congestion management for an FR PVC:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter FR class view.	**fr class** *class-name*	N/A
3. Enable FR congestion management for FR PVCs.	**congestion-threshold** { **de** | **ecn** } *queue-percentage*	By default, FR congestion management is disabled for FR PVCs.

Configuring FR DE rule list

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure a DE rule list.	• Configure an interface-based DE rule list: **fr del** *list-number* **inbound-interface** *interface-type interface-number* • Configure an IP-based DE rule list: **fr del** *list-number* **protocol ip** [**acl** *acl-number* | **fragments** | **greater-than** *bytes* | **less-than** *bytes* | **tcp** *ports* | **udp** *ports*]	Use one of the commands. By default, no DE rule list is created.
3. Enter FR interface view.	**interface** *interface-type interface-number*	N/A
4. Apply the DE rule list to the specified FR PVC.	**fr de del** *list-number* **dlci** *dlci-number*	By default, no DE rule list is applied to an FR PVC. Up to 10 DE rule lists can be applied to a device, and a DE rule list can be configured with up to 100 DE rules.

Configuring FR PVC queuing

With FRTS enabled on an FR interface, each FR PVC of the interface is configured with an independent queuing mechanism.

Configuration restrictions and guidelines

- By default, FR PVCs use FIFO queuing.
- With congestion management enabled, an FR interface supports only FIFO queuing.
- With congestion management enabled, an FR PVC supports only FIFO queuing.

Configuration procedure

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter FR class view.	**fr class** *class-name*	N/A
3. Configure FIFO queue length for the FR PVC.	**fifo queue-length** *queue-length*	Optional. The default setting is 40.

Configuring FR fragmentation

The devices support end-to-end FRF.12 fragmentation.

On low-speed FR links, large data packets cause excessive delay. FR fragmentation can fragment large FR packets into several small packets which can be transmitted on low-speed links with low delay.

When voice packets and data packets are transmitted simultaneously, large data packets occupy the bandwidth for a long time. As a result, voice packets are delayed or even dropped, affecting voice quality. The purpose of FR fragmentation configuration is to reduce delay for voice packets and guarantee the real-time transmission of voice packets. With FR fragmentation configured, large data packets are fragmented into small data fragments. Voice packets and the data fragments are sent alternatively, so that voice packets can be timely and evenly processed and the delay for voice packets is reduced.

Configuration restrictions and guidelines

- The configured FR fragmentation function takes effect after you associate the FR PVCs requiring FR fragmentation with the FR class and enable FRTS on the FR PVCs.
- MFR interfaces do not support FRF.12 fragmentation. If the interfaces at both ends of a link are MFR interfaces with FRF.12 fragmentation enabled, FRF.12 fragmentation does not take effect. Packets are sent out from the local end without being fragmented and can be received by the remote end. When pinging the remote end on the local end, you can get response from the remote end. If the local MFR interface is connected to a normal FR interface (a serial interface with FR encapsulation enabled), FRF.12 fragmentation does not work at the local end and packets are sent out from the local end without being fragmented, however, FRF.12 fragmentation takes effect on the remote end.

Configuration procedure

To configure FR fragmentation:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter FR class view.	**fr class** *class-name*	N/A
3. Enable FR fragmentation.	**fragment** [*fragment-size*]	By default, FR fragmentation is disabled.

Displaying and maintaining FR QoS

Task	Command	Remarks
Display the mapping relationship between FR classes and interfaces (including the DLCIs of an interface, subinterfaces of an interface, and the DLCIs of subinterfaces).	**display fr class-map** { **fr-class** *class-name* | **interface** *interface-type interface-number* } [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display the configuration and statistics information about FR QoS.	**display fr pvc-info** [**interface** *interface-type interface-number*] [*dlci-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display information about all the configured FR switching PVCs.	**display fr switch-table** { **all** | **name** *switch-name* | **interface** *interface-type interface-number* } [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Task	Command	Remarks
Display information about CBQ applied to an interface.	**display qos policy interface** [*interface-type interface-number* [**dlci** *dlci-number* | **inbound** | **outbound**]] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display information about FR fragmentation.	**display fr fragment-info** [**interface** *interface-type interface-number*] [*dlci-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.
Display the statistics information about data transmitted and received through FR.	**display fr statistics** [**interface** *interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

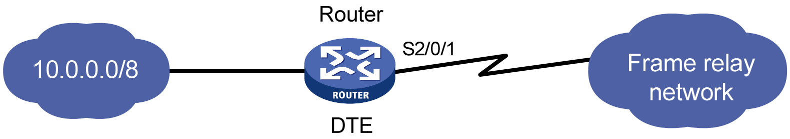
FR QoS configuration examples

FRTS configuration example

Network requirements

As shown in Figure 50, the router connects to the FR network through Serial 2/0/1. Its average transmit rate is 96 kbps, maximum transmit rate is 128 kbps, and minimum transmit rate is 32 kbps. Configure FRTS on the router to adjust 20% of the BECN-flagged traffic every time.

Figure 50 Network diagram



Configuration procedure

# Create FR class **96k** and configure its FRTS parameters.

```
[Router] fr class 96k
[Router-fr-class-96k] cir allow 96000
[Router-fr-class-96k] cir 32000
[Router-fr-class-96k] cbs 96000
[Router-fr-class-96k] ebs 32000
[Router-fr-class-96k] traffic-shaping adaptation becn 20
[Router-fr-class-96k] quit
```

# Enable FR encapsulation and FRTS on interface Serial 2/0/1.

```
[Router] interface serial 2/0/1
[Router-Serial2/0/1] link-protocol fr
[Router-Serial2/0/1] fr traffic-shaping
```

# Create an FR PVC and apply FR class **96k** to the FR PVC.

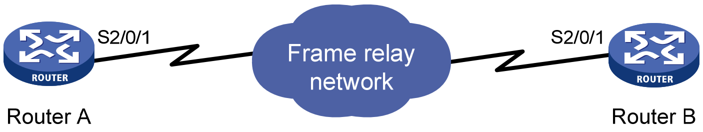
```
[Router-Serial2/0/1] fr dlci 16
[Router-fr-dlci-Serial2/0/1-16] fr-class 96k
```

FR fragmentation configuration example

Network requirements

As shown in Figure 51, Router A connects to Router B through an FR network. Because many large-sized data packets pass through the FR network, the transmission delay is very high. To reduce transmission delays, enable FR fragmentation (FRF.12) on the two devices to fragment large-sized data packets into small data packets.

Figure 51 Network diagram



Configuration procedure

1. Configure Router A:

   # Create FR class **test1**, enable FR fragmentation, and set the fragment size to 128 bytes.

   ```
   <RouterA> system-view
   [RouterA] fr class test1
   [RouterA-fr-class-test1] fragment 128
   [RouterA-fr-class-test1] quit
   ```

   # Enable FR encapsulation and FRTS on interface Serial 2/0/1.

   ```
   [RouterA] interface serial 2/0/1
   [RouterA-Serial2/0/1] link-protocol fr
   [RouterA-Serial2/0/1] ip address 10.1.1.2 255.0.0.0
   [RouterA-Serial2/0/1] fr traffic-shaping
   ```

   # Create DLCI 16, and associate the FR class **test1** with DLCI 16.

   ```
   [RouterA-Serial2/0/1] fr dlci 16
   ```

   # Apply the FR class **test1** to DLCI 16.

   ```
   [RouterA-fr-dlci-Serial2/0/1-16] fr-class test1
   ```

2. Configure Router B:

   # Create FR class **test1**, enable FR fragmentation, and set the fragment size to 128 bytes.

   ```
   <RouterB> system-view
   [RouterB] fr class test1
   [RouterB-fr-class-test1] fragment 128
   [RouterB-fr-class-test1] quit
   ```

   # Enable FR encapsulation and FRTS on interface Serial 2/0/1.

   ```
   [RouterB] interface serial 2/0/1
   [RouterB-Serial2/0/1] link-protocol fr
   [RouterB-Serial2/0/1] ip address 10.1.1.1 255.0.0.0
   [RouterB-Serial2/0/1] fr traffic-shaping
   ```

   # Create DLCI 16 and apply FR class **test1** to DLCI 16.

   ```
   [RouterB-Serial2/0/1] fr dlci 16
   [RouterB-fr-dlci-Serial2/0/1-16] fr-class test1
   ```

Configuring HQoS

HQoS overview

Hierarchical Quality of Service (QoS) uniformly manages traffic and hierarchically schedules traffic by user, network service, and application. It provides more granular traffic control and quality assurance services than traditional QoS.

HQoS-capable devices can hierarchically classify and schedule traffic, for example, by both user and application. HQoS guarantees QoS for advanced users and saves the overall networking costs.

HQoS implementation methods

On 6600/HSR6600 routers, you can implement HQoS by nesting QoS policies or applying hierarchical CAR policies to an interface.

Implementing HQoS through nesting QoS policies

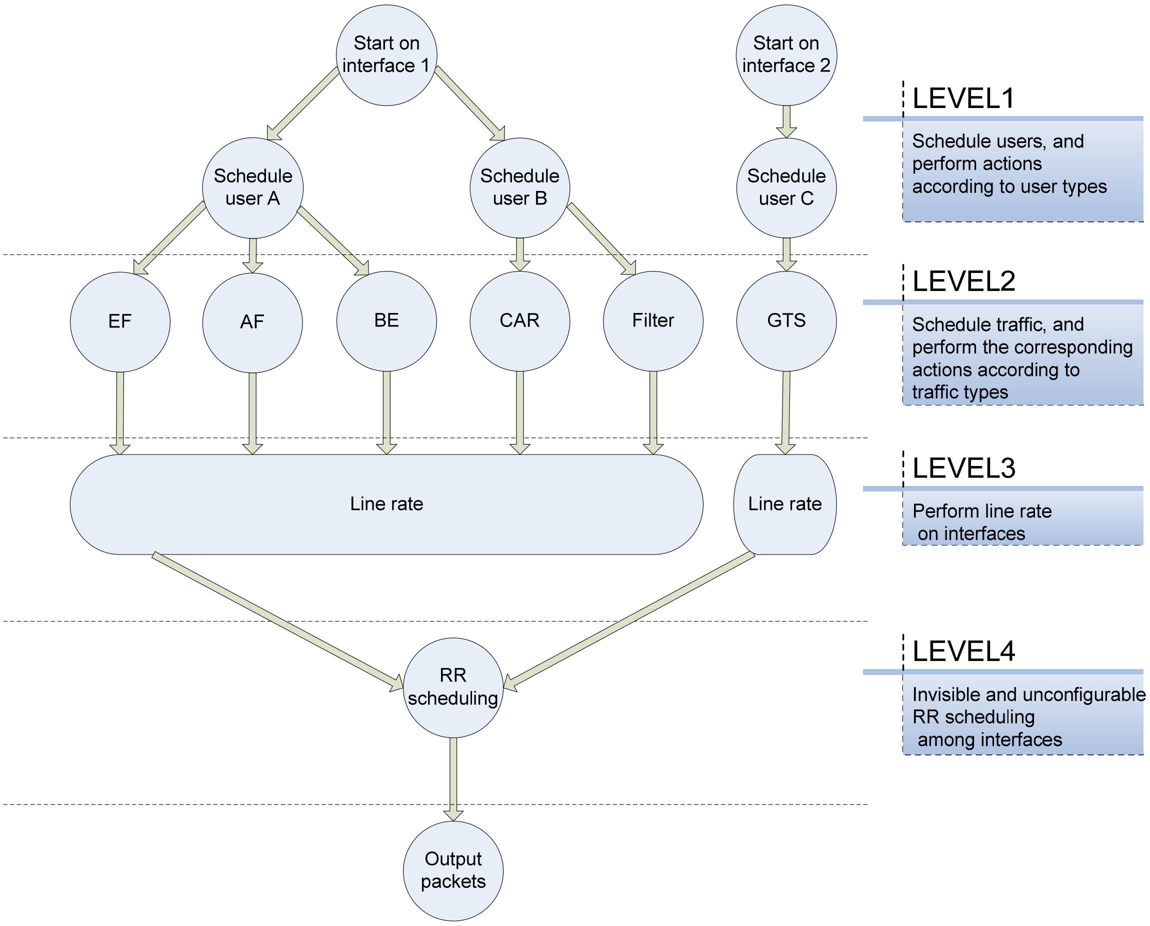
Implement HQoS though nesting QoS policies as follows:

- Configure traffic classes to identify different types of packets.
- Configure traffic behaviors with the corresponding actions to be performed for different classes of packets.
- Associate classes with behaviors in a QoS policy, and apply the QoS policy, typically, to an interface.

When packets pass an interface with the QoS policy applied, packets are classified into multiple classes according to the match criteria, and the actions in the corresponding behaviors are performed for these classes of packets.

You can reference a QoS policy in a traffic behavior to re-classify the traffic class associated with the behavior and take action on the re-classified traffic as defined in the policy. The QoS policy referenced in the traffic behavior is called the ”child QoS policy"; the QoS policy that references the behavior is called the "parent QoS policy".

Figure 52 Implementing 4-level HQoS scheduling through nesting QoS polices


Start on interface 1
Start on interface 2
Schedule user A
Schedule user B
Schedule user C
EF
AF
BE
CAR
Filter
GTS
Line rate
Line rate
RR scheduling
Output packets
LEVEL1
Schedule users, and perform actions according to user types
LEVEL2
Schedule traffic, and perform the corresponding actions according to traffic types
LEVEL3
Perform line rate on interfaces
LEVEL4
Invisible and unconfigurable RR scheduling among interfaces

As shown in Figure 52, start the HQoS scheduling through nesting QoS policies on the interfaces. The HQoS scheduling operates in the following workflow:

1. First, the classes in the parent QoS policy is used to differentiate users, and the corresponding actions are performed for the users.
2. Then, the classes of the child QoS policy are used to differentiate services, and actions such as expedited forwarding (EF), assured forwarding (AF), best-effort (BE), committed access rate (CAR), traffic filtering, and generic traffic shaping (GTS) are performed for the corresponding classes, respectively.
3. After the parent QoS policy and child QoS policy process the packet, line rate is performed for packets, and round robin (RR) scheduling is performed among interfaces to send packets out different interfaces.

Implementing HQoS through interface-level hierarchical CAR

CAR rate-limits the specific traffic flows. CAR polices the rate of traffic entering the network, and uses the token bucket to color the packets processed by CAR. The token bucket size is committed burst size (CBS) + excess burst size (EBS). The system puts tokens into the token bucket at the rate of committed information rate (CIR). When the system forwards packets, if the token bucket has enough tokens for forwarding the packets and the traffic rate is lower than the CIR, the system colors the packets green and performs the action for green packets (including marking priority, forwarding, and proceeding with the next CAR action); if the token bucket does not have enough tokens for forwarding the packets and the traffic rate is higher than the CIR, the system colors the packets red and performs the action for red packets (including marking priority, forwarding, proceeding with the next CAR action, and dropping).

CAR is widely used in networks because it is easy to configure and provides obvious rate-limiting effects. However, traditional CAR provides a fixed upper rate limit, and cannot enable bandwidth sharing and prioritize the specific traffic. The routers supports interface-level hierarchical CAR, which can meet the requirements mentioned above.

To use the interface-level hierarchical CAR to implement HQoS, configure multiple CAR policies of different levels in the inbound or outbound direction of an interface.

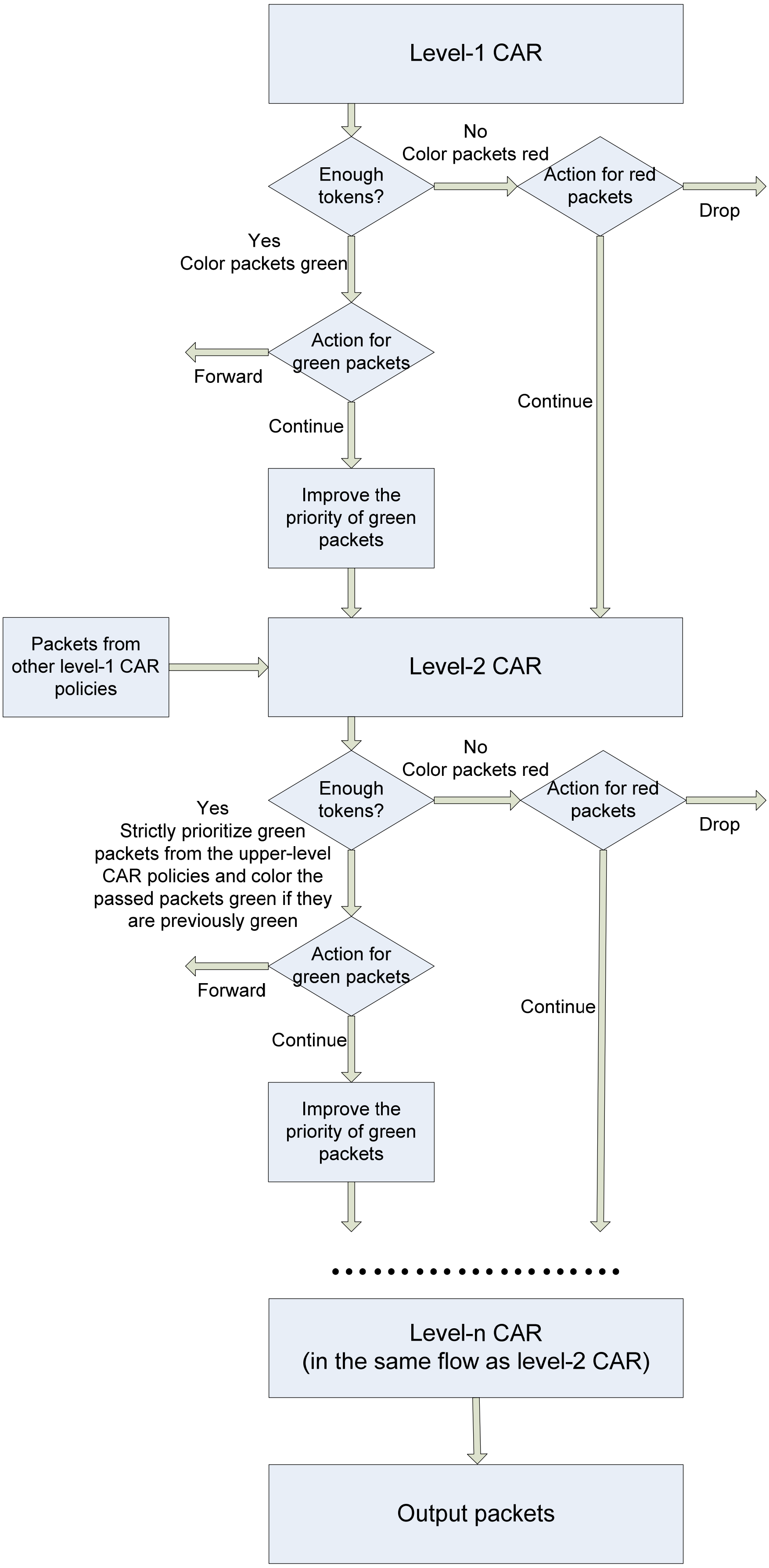
In the level-1 CAR policy, color the packets to be prioritized and within the guaranteed bandwidth green to improve the packet priority, and color the packets exceeding the guaranteed bandwidth red. The action for green and red packets is using the next CAR policy, so the red packets continue to be processed by using the level-2 CAR policy.

In the level-2 CAR policy, red packets are forwarded until all green packets are forwarded. In this way, the specific packets are prioritized. Additionally, compared with the level-1 CAR policy, the level-2 CAR policy sets higher bandwidth for low-priority traffic to implement bandwidth sharing.

HQoS implemented by using interface-level hierarchical CAR is easy to configure, and can cooperate with CAR actions and policy-based routing to implement different QoS functions. At the same time, HQoS implemented by using interface-level hierarchical CAR can better utilize the bandwidth than HQoS implemented through nesting QoS policies.

Packets that have not been processed by CAR are green by default. Packets colored red by CAR cannot be colored green.

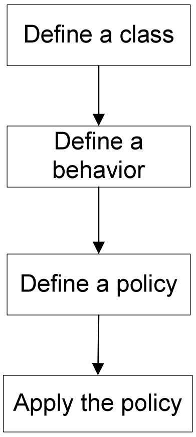
Figure 53 Implementing HQoS through interface-level hierarchical CAR



Level-1 CAR
No
Color packets red
Enough tokens?
Action for red packets
Drop
Yes
Color packets green
Action for green packets
Forward
Continue
Continue
Improve the priority of green packets
Packets from other level-1 CAR policies
Level-2 CAR
No
Color packets red
Enough tokens?
Action for red packets
Drop
Yes
Strictly prioritize green packets from the upper-level CAR policies and color the passed packets green if they are previously green
Action for green packets
Forward
Continue
Continue
Improve the priority of green packets
Level-n CAR
(in the same flow as level-2 CAR)
Output packets

Implementing HQoS through nesting QoS policies

Figure 54 QoS policy configuration procedure



Defining a traffic class

The system pre-defines some traffic classes and defines general match criteria for them. A user-defined traffic class cannot be named the same as a system-defined traffic class. You can use these pre-defined traffic classes when defining a policy. The system-defined traffic classes include:

- The default traffic class

  default-class—Matches the default traffic.
- DSCP-based pre-defined traffic classes

  ef, af1, af2, af3, af4—Matches IP DSCP value ef, af1, af2, af3, af4 respectively.
- IP precedence-based pre-defined traffic classes

  ip-prec0, ip-prec1, …ip-prec7—Matches IP precedence value 0, 1, …7 respectively.
- MPLS EXP-based pre-defined traffic classes

  mpls-exp0, mpls-exp1, …mpls-exp7—Matches MPLS EXP value 0, 1, …7 respectively.

To define a traffic class:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic class and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	By default, the operator of a class is AND. The operator of a class can be AND or OR. • **AND**—A packet is assigned to a class only when the packet matches all the criteria in the class. • **OR**—A packet is assigned to a class if it matches any of the criteria in the class.
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	For more information, see the **if-match** command in *ACL and QoS Command Reference*.

Defining a traffic behavior

A traffic behavior is a set of QoS actions (such as traffic filtering, shaping, policing, and priority marking) to take on a class of traffic.

The system pre-defines some traffic behaviors and defines general QoS actions for them. A user-defined behavior cannot be named the same as a system-defined behavior. You can use these behaviors when defining a policy. The system-defined behaviors are as follows:

- **ef**—Expedited forwarding.
- **af**—Assured forwarding.
- **be**—Best-effort.
- **be-flow-based**—Uses the weighted random early detection (WRED) drop policy.

To define a traffic behavior:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a traffic behavior and enter traffic behavior view.	**traffic behavior** *behavior-name*	N/A
3. Configure actions in the traffic behavior.	See the following parts in QoS configuration: traffic policing, traffic filtering, traffic redirecting, priority marking, traffic accounting, and so on.	

Defining a policy

Configuring parent QoS policy

You associate a behavior with a class in a QoS policy to perform the actions defined in the behavior for the class of packets.

The system provides a pre-defined QoS policy named **default**. It includes the associations between predefined classes and predefined traffic behaviors:

- Class **ef** with behavior **ef**.
- Classes **af1** through **af4** with behavior **af**.
- Class **default-class** with behavior **be**.

You cannot name a user-defined QoS policy the same as the system-defined QoS policy.

To associate a class with a behavior in a policy:

Step	Command
1. Enter system view.	**system-view**
2. Create a policy and enter policy view.	**qos policy** *policy-name*
3. Associate a class with a behavior in the policy.	**classifier** *classifier-name* **behavior** *behavior-name*

NOTE:

On some devices, if the ACL contains deny rules, the if-match clause is ignored and the matching process continues.

Configuring QoS policy nesting

You can reference a QoS policy in a traffic behavior to re-classify the traffic class associated with the behavior and take action on the re-classified traffic as defined in the policy. The QoS policy referenced in the traffic behavior is called the "child QoS policy"; the QoS policy that references the behavior is called the "parent QoS policy".

To nest a child QoS policy in a parent QoS policy:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Create a class for the parent QoS policy and enter class view.	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	N/A
3. Configure match criteria.	**if-match** [**not**] *match-criteria*	N/A
4. Return to system view.	**quit**	N/A
5. Create a behavior for the parent QoS policy and enter behavior view.	**traffic behavior** *behavior-name*	N/A
6. Nest the child QoS policy.	**traffic-policy** *policy-name*	The QoS policy specified for the *policy-name* argument must already exist.
7. Return to system view.	**quit**	N/A
8. Create the parent QoS policy and enter parent QoS policy view.	**qos policy** *policy-name*	N/A
9. Associate the class with the behavior in the parent QoS policy.	**classifier** *classifier-name* **behavior** *behavior-name*	N/A

NOTE:

To nest QoS policies successfully, follow these guidelines:

- The support for QoS policy nesting depends on your device model.
- If class-based queuing (CBQ) is configured in the child QoS policy, configure generic traffic shaping (GTS) in the parent QoS policy and make sure that the GTS bandwidth configured in the parent QoS policy is equal to or greater than the CBQ bandwidth configured in the child QoS policy.
- If GTS bandwidth in the parent QoS policy is configured in percentage, the CBQ bandwidth in the child QoS policy must be also configured in percentage; if it is configured as an absolute number, the CBQ bandwidth in the child QoS policy can be configured in either percentage or as an absolute number.
- GTS cannot be configured in the child QoS policy.

Applying the QoS policy

You can apply a QoS policy to an interface or permanent virtual circuit (PVC). The policy takes effect on the traffic sent or received on the interface or PVC.

A policy can be applied to multiple interfaces or PVCs, but only one policy can be applied in one direction (inbound or outbound) of an interface or PVC.

To apply the QoS policy to an interface or PVC:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A

Step	Command	Remarks
2. Enter interface view or PVC view.	- Enter interface view: **interface** interface-type interface-number - Enter PVC view: **a. interface atm** *interface-number* **b. pvc** *vpi/vci*	Settings in interface view take effect on the current interface. Settings in PVC view take effect on the current PVC.
3. Apply the policy to the interface, port group, or PVC.	**qos apply policy** *policy-name* { **inbound** | **outbound** }	N/A

NOTE:

The QoS policy applied to the outgoing traffic on an interface or PVC does not regulate local packets, which are critical protocol packets sent by the local system for operation maintenance. The most common local packets include link maintenance, routing (IS-IS, BGP, and OSPF for example), RIP, LDP, and SSH packets.

Implementing HQoS through interface-level hierarchical CAR

Configuring CAR-list-based traffic policing

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure a committed access rate (CAR) list.	**qos carl** *carl-index* { **precedence** *precedence-value* | **mac** *mac-address* | **mpls-exp** *mpls-exp-value* | **dscp** *dscp-list* | { **destination-ip-address** | **source-ip-address** } { **subnet** *ip-address mask-length* | **range** *start-ip-address* **to** *end-ip-address* } [**per-address** [**shared-bandwidth**]] }	Configure rules on the CAR list.
3. Enter interface view.	**interface** *interface-type interface-number*	N/A
4. Configure a CAR list based CAR policy on the interface.	**qos car** { **inbound** | **outbound** } **carl** *carl-index* **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action*] [**red** *action*]	N/A

Step	**Command**	**Remarks**
5. Display the CAR information on the specified interface.	**display qos car interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Configuring ACL-based traffic policing

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Configure an ACL.	See "Configuring ACLs."	Configure rules for the ACL.
3. Enter interface view.	**interface** *interface-type interface-number*	N/A
4. Configure an ACL based CAR policy on the interface or port group.	**qos car** { **inbound** | **outbound** } **acl** [**ipv6**] *acl-number* **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**pir** *peak-information-rate*] [**green** *action*] [**red** *action*]	N/A
5. Display the CAR information on the specified interface.	**display qos car interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Configuring traffic policing for all traffic

Step	**Command**	**Remarks**
1. Enter system view.	**system-view**	N/A
2. Enter interface view.	**interface** *interface-type interface-number*	N/A
3. Configure a CAR action for all traffic on the interface.	**qos car** { **inbound** | **outbound** } **any cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action*] [**red** *action*]	N/A
4. Display the CAR information on the specified interface.	**display qos car interface** [*interface-type interface-number*] [**|** { **begin** | **exclude** | **include** } *regular-expression*]	Available in any view.

Configuration example for HQoS through nesting QoS policies

Network requirements

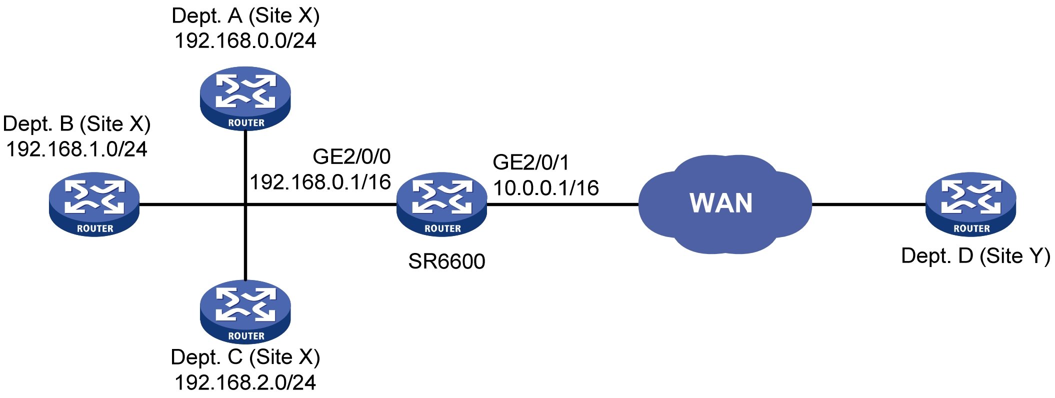
A company has agencies in site X and site Y, respectively. The agency in site X has three departments, A, B, and C. The agency in site Y has one department D. Site X and site Y are connected through a service provider WAN. Site X is connected to the WAN through a 50-Mbps Ethernet port.

Department A is on network segment 192.168.0.0/24, department B is on network segment 192.168.1.0/24, and department C is on network segment 192.168.2.0/24. Departments A and B are important departments, and need the IP voice communication with department D. The voice packets are UDP packets with destination port number 2000. Additionally, bandwidth must be guaranteed for the data traffic from departments A and B to department D. The communication between department C and department D is unimportant, and carries only data traffic.

Configure HQoS through nesting QoS policies to:

- Guarantee 15 Mbps of bandwidth and limit the bandwidth to 15 Mbps for the traffic from department A to department D, and guarantee 25 Mbps of bandwidth and limit the bandwidth to 25 Mbps for the traffic from department B to department D.
- Guarantee 1.5 Mbps of bandwidth for the voice traffic from department A to department D, and 2.5 Mbps of bandwidth for the voice traffic from department B to department D. Send voice traffic with the highest priority.

Figure 55 Network diagram



Configuration procedures

# Configure IP addresses for interfaces. (Details not shown)

# Create three ACLs to match packets sourced from IP segment 192.168.0.0/24 (department A), packets sourced from IP segment 192.168.1.0/24 (department B), and packets with destination UDP port number 2000.

```
<Router> system-view
[Router] acl number 3000 name A
[Router-acl-adv-3000-A] rule 0 permit ip source 192.168.0.0 0.0.0.255
[Router-acl-adv-3000-A] quit
[Router] acl number 3001 name B
[Router-acl-adv-3001-A] rule 0 permit ip source 192.168.1.0 0.0.0.255
```

```
[Router-acl-adv-3001-A] quit
[Router] acl number 3002 name voice
[Router-acl-adv-3002-voice] rule 0 permit udp destination-port eq 2000
[Router-acl-adv-3002-voice] quit
```

# Create four classes to match all traffic from department A, voice traffic from department A, all traffic from department B, and voice traffic from department B by using ACL 3000, ACL 3002, ACL 3001, and ACL 3002 as the match criterion, respectively.

```
[Router] traffic classifier Afather
[Router-classifier-Afather] if-match acl 3000
[Router-classifier-Afather] traffic classifier Ason
[Router-classifier-Ason] if-match acl 3002
[Router-classifier-Ason] traffic classifier Bfather
[Router-classifier-Bfather] if-match acl 3001
[Router-classifier-Bfather] traffic classifier Bson
[Router-classifier-Bson] if-match acl 3002
[Router-classifier-Bson] quit
```

# Configure traffic behaviors for the child QoS policy: use EF to guarantee 1.5 Mbps of bandwidth for the voice traffic from department A and 2.5 Mbps of bandwidth for voice traffic from department B.

```
[Router] traffic behavior Ason
[Router-behavior-Ason] queue ef bandwidth 1500
[Router-behavior-Ason] traffic behavior Bson
[Router-behavior-Bson] queue ef bandwidth 2500
[Router-behavior-Bson] quit
```

# Associate the configured traffic behavior with the corresponding traffic class in each child policy.

```
[Router] qos policy Ason
[Router-qospolicy-Ason] classifier Ason behavior Ason
[Router-qospolicy-Ason] qos policy Bson
[Router-qospolicy-Bson] classifier Bson behavior Bson
[Router-qospolicy-Bson] quit
```

# Configure traffic behaviors for the parent QoS policy, use the EF queuing and GTS to guarantee 15 Mbps of bandwidth and limit the bandwidth to 15 Mbps for traffic from department A and guarantee 25 Mbps of bandwidth and limit the bandwidth to 25 Mbps for traffic from department B, and nest the child QoS policies in the traffic behaviors.

```
[Router] traffic behavior Afather
[Router-behavior-Afather] gts cir 15000
[Router-behavior-Afather] queue ef bandwidth 15000
[Router-behavior-Afather] traffic-policy Ason
[Router-behavior-Afather] traffic behavior Bfather
[Router-behavior-Bfather] gts cir 25000
[Router-behavior-Bfather] queue ef bandwidth 25000
[Router-behavior-Bfather] traffic-policy Bson
[Router-behavior-Bfather] quit
```

# Associate the traffic behaviors with the corresponding classes in the parent QoS policy.

```
[Router] qos policy out
[Router-qospolicy-out] classifier Afather behavior Afather
[Router-qospolicy-out] classifier Bfather behavior Bfather
[Router-qospolicy-out] quit
```

# Set the maximum available bandwidth and maximum reserved bandwidth for interface GigabitEthernet 2/0/1, configure line rate on interface GigabitEthernet 2/0/1, and apply the parent QoS policy to the outgoing traffic of interface GigabitEthernet 2/0/1.

```
[Router] interface GigabitEthernet 2/0/1
[Router-GigabitEthernet2/0/1] qos max-bandwidth 50000
[Router-GigabitEthernet2/0/1] qos reserved-bandwidth pct 100
[Router-GigabitEthernet2/0/1] qos lr outbound cir 50000
[Router-GigabitEthernet2/0/1] qos apply policy out outbound
```

Configuration example for implementing hierarchical CAR through nesting QoS policies

Network requirements

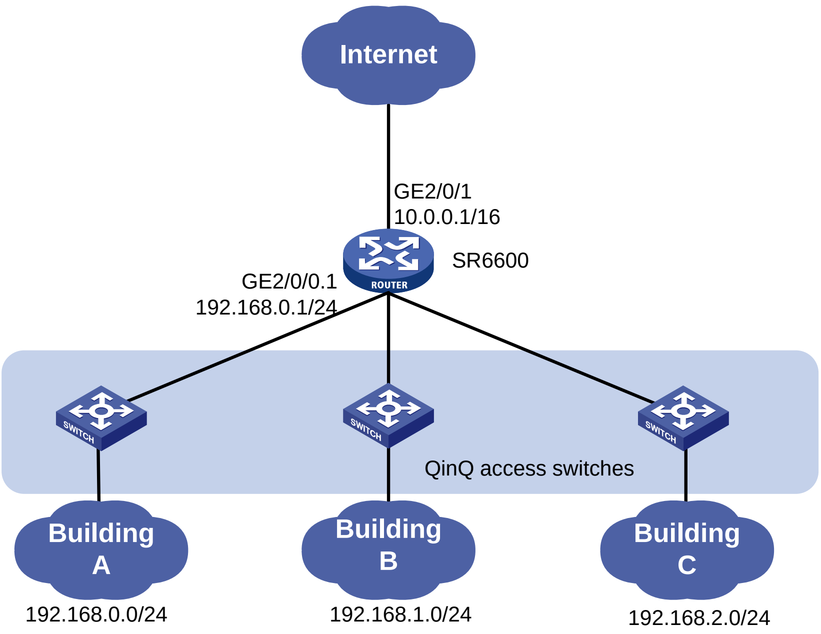
A university has three dormitory buildings, A, B, and C. Each dormitory building provides 200 dormitories, and each dormitory is assigned a VLAN ID (1 to 200). The dormitories of a building are uplinked through the access switch for the building. The access switch of a building performs QinQ for the packets from users, and tags the packets with a VLAN ID assigned to the building, which is 1 for building A, 2 for building B, and 3 for building C. The QinQ packets are sent to subinterfaces GigabitEthernet 2/0/0.1, GigabitEthernet 2/0/0.2, and GigabitEthernet 2/0/0.3 of the router. The router performs QinQ termination for the packets, and accesses the Internet through a 300-Mbps Ethernet interface.

Buildings A, B, and C are on network segments 192.168.0.0/24, 192.168.1.0/24, and 192.168.2.0/24, respectively. Because the bandwidth of the WAN is limited, you need to rate-limit the users: limit the rate of accessing the external network for each building, and limit the rate of accessing non-HTTP services.

Configure hierarchical CAR through nesting QoS policies to:

- Limit the rate of accessing the external network to 100 Mbps for each building.
- Limit the rate of accessing the non-HTTP services to 30 Mbps for each building.

Figure 56 Network diagram



Configuration procedures

1. Configure the QinQ access switches
2. Configure QinQ on the access switches.

   For more information, see the corresponding configuration guide for the switches.
3. Configure the router:

   This section takes subinterface GigabitEthernet 2/0/0.1 that connects to building A as an example. The configurations for buildings B and C are the same.

   # Configure IP addresses for interfaces as shown in the network diagram. (Details not shown)

   # Configure QinQ termination on subinterface GigabitEthernet 2/0/0.1.

   ```
   <Router> system-view
   [Router] interface GigabitEthernet 2/0/0.1
   [Router-GigabitEthernet2/0/0.1] vlan-type dot1q vid 1 second-dot1q 1 to 200
   [Router-GigabitEthernet2/0/0.1] vlan-termination broadcast enable
   [Router-GigabitEthernet2/0/0.1] quit
   ```

   # Configure ACLs to match the internal IP network segment and HTTP services, respectively.

   ```
   [Router] acl number 3000 name inner
   [Router-acl-adv-3000-inner] rule 0 permit ip source 192.168.0.0 0.0.3.255
   [Router-acl-adv-3000-inner] quit
   [Router] acl number 3001 name http
   [Router-acl-adv-3001-http] rule 0 permit tcp destination-port eq 80
   [Router-acl-adv-3001-http] quit
   ```

   # Configure a traffic class for the father QoS policy to match the traffic accessing the external network.

   ```
   [Router] traffic classifier A
   [Router-classifier-A] if-match acl 3000
   [Router-classifier-A] quit
   ```

   # Configure a class for the child QoS policy to match the traffic accessing non-HTTP services.

   ```
   [Router] traffic classifier http
   [Router-classifier-http] if-match not acl 3001
   [Router-classifier-http] quit
   ```

   # Configure a traffic behavior for the child QoS policy to limit the rate to 30 Mbps.

   ```
   [Router] traffic behavior http
   [Router-behavior-http] car cir 30000
   [Router-behavior-http] quit
   ```

   # Associate the traffic behavior with the corresponding traffic class in the child QoS policy.

   ```
   [Router] qos policy http
   [Router-qospolicy-http] classifier http behavior http
   [Router-qospolicy-http] quit
   ```

   # Configure a traffic behavior for the parent QoS policy to limit the rate to 100 Mbps, and nest the child QoS policy in the traffic behavior.

   ```
   [Router] traffic behavior A
   [Router-behavior-A] traffic-policy http
   [Router-behavior-A] car cir 100000
   [Router-behavior-A] quit
   ```

   # Associate the traffic behavior with the corresponding traffic class in the parent QoS policy, and apply the parent QoS policy to the incoming traffic of interface GigabitEthernet 2/0/0.1.

```
[Router] qos policy A
[Router-qospolicy-A] classifier A behavior A
[Router-qospolicy-A] quit
[Router] interface gigabitethernet 2/0/0.1
[Router-GigabitEthernet2/0/0.1] qos apply policy A inbound
```

Configuration example for implementing HQoS in an MPLS network through nesting QoS policies

Network requirements

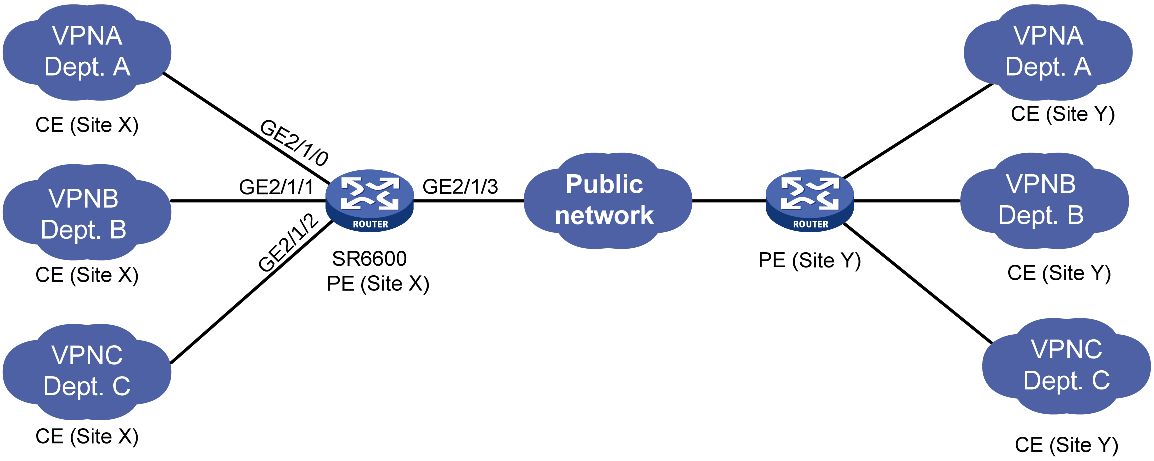
A company has agencies in site X and site Y, which communicate through a MPLS L3VPN. The routers function as provide edges (PEs) to connect to the public network. The company provides 100 Mbps of downstream bandwidth. Each site has three departments, A, B, and C, which belong to VPNA, VPNB, and VPNC and connect to GigabitEthernet 2/1/0, GigabitEthernet 2/1/1, and GigabitEthernet 2/1/2 of PE (Site X), respectively.

The service provider provides 100 Mbps of bandwidth for PE (Site X). Site X and site Y each have two types of traffic, voice traffic with IP precedence value being 7 and data traffic with IP precedence values being not 7. The voice traffic has higher priority than the data traffic. Department A and B are important departments, so bandwidth must be guaranteed for them. The services of department C are unimportant.

Configure HQoS through nesting QoS policies to:

- Guarantee 40 Mbps of bandwidth and limit the bandwidth to 40 Mbps for traffic from department A (VPNA) to site Y.
- Guarantee 4 Mbps of bandwidth for voice traffic from department A to site Y.
- Guarantee 35 Mbps of bandwidth and limit the bandwidth to 35 Mbps for traffic from department B (VPNB) to site Y.
- Guarantee 3 Mbps of bandwidth for voice traffic from department B to site Y.

Figure 57 Network diagram



Configuration procedures

# Configure MPLS L3VPN.

For more information, see *MPLS Configuration Guide*.

# Configure a QoS policy to mark the traffic from VPNA with local QoS ID 1.

```
<Router> system-view
[Router] traffic classifier any
[Router-classifier-any] if-match any
[Router] traffic behavior vpnA
[Router-behavior-vpnA] remark qos-local-id 1
[Router] qos policy vpnA
[Router-qospolicy-vpnA] classifier any behavior vpnA
[Router-qospolicy-vpnA] quit
```

# Configure a QoS policy to mark the traffic from VPNB with local QoS ID 2.

```
[Router] traffic behavior vpnB
[Router-behavior-vpnA] remark qos-local-id 2
[Router] qos policy vpnB
[Router-qospolicy-vpnB] classifier any behavior vpnB
[Router-qospolicy-vpnB] quit
```

# Apply the QoS policies to the incoming traffic of GigabitEthernet 2/1/0 and GigabitEthernet 2/1/1, through which VPNA and VPNB are connected to the public network.

```
[Router] interface GigabitEthernet 2/1/0
[Router-GigabitEthernet2/1/0] qos apply policy vpnA inbound
[Router] interface GigabitEthernet 2/1/1
[Router-GigabitEthernet2/1/1] qos apply policy vpnB inbound
[Router-GigabitEthernet2/1/1] quit
```

# Configure traffic classes for the parent QoS policy, which is to be applied to the public network interface, to match local QoS ID 1 and 2, respectively.

```
[Router] traffic classifier publicvpnA
[Router-classifier-publicvpnA] if-match qos-local-id 1
[Router-classifier-publicvpnA] quit
[Router] traffic classifier publicvpnB
[Router-classifier-publicvpnB] if-match qos-local-id 2
[Router-classifier-publicvpnB] quit
```

# Configure a class for the child QoS policy to match the voice traffic with MPLS EXP 7.

```
[Router] traffic classifier mplsvoice
[Router-classifier-mplsvoice] if-match mpls-exp 7
[Router-classifier-mplsvoice] quit
```

# Configure traffic behaviors for the child QoS policy, and configure EF to guarantee 4 Mbps and 3 Mbps of bandwidth.

```
[Router] traffic behavior vpnAvoice
[Router-behavior-vpnAvoice] queue ef bandwidth 4000
[Router-behavior-vpnAvoice] quit
[Router] traffic behavior vpnBvoice
[Router-behavior-vpnBvoice] queue ef bandwidth 3000
[Router-behavior-vpnBvoice] quit
```

# Associate the traffic behavior with the corresponding traffic class in each child QoS policy.

```
[Router] qos policy vpnAvoice
[Router-qospolicy-vpnAvoice] classifier mplsvoice behavior vpnAvoice
[Router-qospolicy-vpnAvoice] quit
[Router] qos policy vpnBvoice
[Router-qospolicy-vpnBvoice] classifier mplsvoice behavior vpnBvoice
```

```
[Router-qospolicy-vpnBvoice] quit
```

# Nest the child QoS policies in the corresponding traffic behaviors of the parent QoS policy, and configure EF in the traffic behaviors to guarantee 40 Mbps of bandwidth for VPN A and 35 Mbps of bandwidth for VPNB.

```
[Router] traffic behavior publicvpnA
[Router-behavior-publicvpnA] gts cir 40000
[Router-behavior-publicvpnA] queue ef bandwidth 40000
[Router-behavior-publicvpnA] traffic-policy vpnAvoice
[Router-behavior-publicvpnA] quit
[Router] traffic behavior publicvpnB
[Router-behavior-publicvpnB] gts cir 35000
[Router-behavior-publicvpnB] queue ef bandwidth 35000
[Router-behavior-publicvpnB] traffic-policy vpnBvoice
[Router-behavior-publicvpnB] quit
```

# Associate the traffic classes with the corresponding traffic behaviors in the parent QoS policy.

```
[Router] qos policy publicvpn
[Router-qospolicy-publicvpn] classifier publicvpnA behavior publicvpnA
[Router-qospolicy-publicvpn] classifier publicvpnB behavior publicvpnB
[Router-qospolicy-publicvpn] quit
```

# Configure line rate on interface GigabitEthernet 2/1/3, set the maximum available bandwidth and the maximum reserved bandwidth for interface GigabitEthernet 2/1/3, and apply the parent QoS policy to the outgoing packets of interface GigabitEthernet 2/1/3.

```
[Router]interface GigabitEthernet 2/1/3
[Router-GigabitEthernet2/1/3] qos max-bandwidth 100000
[Router-GigabitEthernet2/1/3] qos reserved-bandwidth pct 100
[Router-GigabitEthernet2/1/3] qos lr outbound cir 100000
[Router-GigabitEthernet2/1/3] qos apply policy publicvpn outbound
```

Configuration example for reserving and sharing bandwidth through interface-level hierarchical CAR

Network requirements

A company has two agencies in site X and site Y. The agency in site X has two departments, A and B. The agency in site Y has one department, C. Site X and site Y are connected through a service provider WAN. Site X and site Y connect to the WAN through a 50-Mbps Ethernet link provided by the service provider.

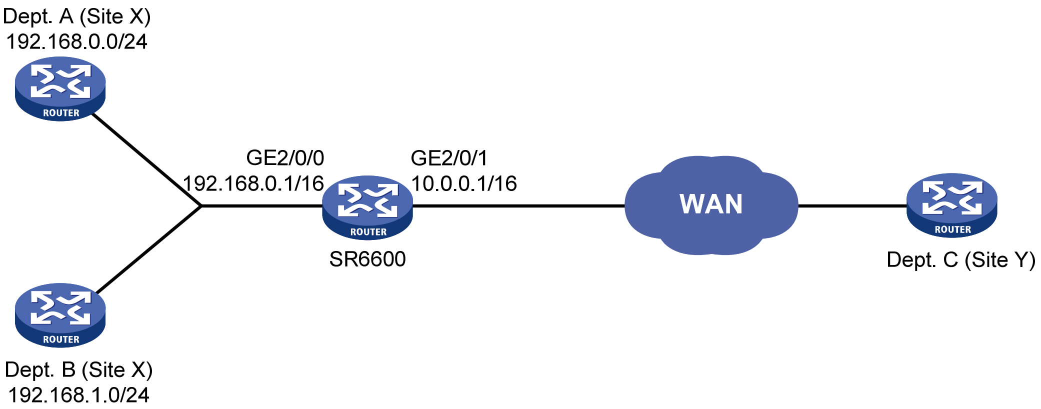
Department A is on network segment 192.168.0.0/24, and department B is on network segment 192.168.1.0/24. Each of department A and department B sends to department C three types of traffic, voice, video, and data, whose IP precedence values are 7, 6, and 0, respectively. Each of the two departments must reserve certain bandwidth for the voice, video, and data services, and share the idle bandwidth when any service does not transmit traffic.

Configure interface-level hierarchical CAR to:

- Reserve 3 Mbps of bandwidth for the voice traffic of department A and 4 Mbps of bandwidth for the voice traffic of department B.

- Reserve 7 Mbps of bandwidth for the video traffic of department A and 8 Mbps of bandwidth for the video traffic of department B.
- Reserve 5 Mbps of bandwidth for the data traffic of department A and 8 Mbps of bandwidth for the data traffic of department B.
- Share the remaining 15 Mbps of bandwidth among these services. When a service does not transmit traffic, the other service can preempt the idle bandwidth.

Figure 58 Network diagram



Configuration procedures

# Configure IP addresses for interfaces according to the network diagram. (Details not shown)

# Configure six ACLs to match the voice traffic from department A, video traffic from department A, data traffic from department A, voice traffic from department B, video traffic from department B, and data traffic from department B.

```
<Router> system-view
[Router] acl number 3000 name Avoice
[Router-acl-adv-3000-Avoice] rule 0 permit ip source 192.168.0.0 0.0.0.255 precedence 7
[Router-acl-adv-3000-Avoice] quit
[Router] acl number 3001 name Avideo
[Router-acl-adv-3001-Avideo] rule 0 permit ip source 192.168.0.0 0.0.0.255 precedence 6
[Router-acl-adv-3001-Avideo] quit
[Router] acl number 3002 name Adata
[Router-acl-adv-3002-Adata] rule 0 permit ip source 192.168.0.0 0.0.0.255 precedence 0
[Router-acl-adv-3002-Adata] quit
[Router] acl number 3003 name Bvoice
[Router-acl-adv-3003-Bvoice] rule 0 permit ip source 192.168.1.0 0.0.0.255 precedence 7
[Router-acl-adv-3003-Bvoice] quit
[Router] acl number 3004 name Bvideo
[Router-acl-adv-3004-Bvideo] rule 0 permit ip source 192.168.1.0 0.0.0.255 precedence 6
[Router-acl-adv-3004-Bvideo] quit
[Router] acl number 3005 name Bdata
[Router-acl-adv-3005-Bdata] rule 0 permit ip source 192.168.1.0 0.0.0.255 precedence 0
[Router-acl-adv-3005-Bdata] quit
```

# Configure interface-level hierarchical CAR on interface GigabitEthernet 2/0/0 as follows:

- Reserve 3 Mbps of bandwidth for voice traffic of department A, 7 Mbps of bandwidth for the video traffic of department A, and 5 Mbps of bandwidth for data traffic of department A.

- Reserve 4 Mbps of bandwidth for voice traffic of department B, 8 Mbps of bandwidth for video traffic of department B, and 8 Mbps of bandwidth for data traffic of department B.
- Limit the bandwidth to 50 Mbps for all the traffic between site X and site Y.

```
[Router] interface GigabitEthernet 2/0/0
[Router-GigabitEthernet2/0/0] qos car inbound acl 3000 cir 3000 green continue red continue
[Router-GigabitEthernet2/0/0] qos car inbound acl 3001 cir 7000 green continue red continue
[Router-GigabitEthernet2/0/0] qos car inbound acl 3002 cir 5000 green continue red continue
[Router-GigabitEthernet2/0/0] qos car inbound acl 3003 cir 4000 green continue red continue
[Router-GigabitEthernet2/0/0] qos car inbound acl 3004 cir 8000 green continue red continue
[Router-GigabitEthernet2/0/0] qos car inbound acl 3005 cir 8000 green continue red continue
[Router-GigabitEthernet2/0/0] qos car inbound any cir 50000 green pass red discard
```

Configuration example for implementing per-IP bandwidth reservation and sharing through interface-level hierarchical CAR

Network requirements

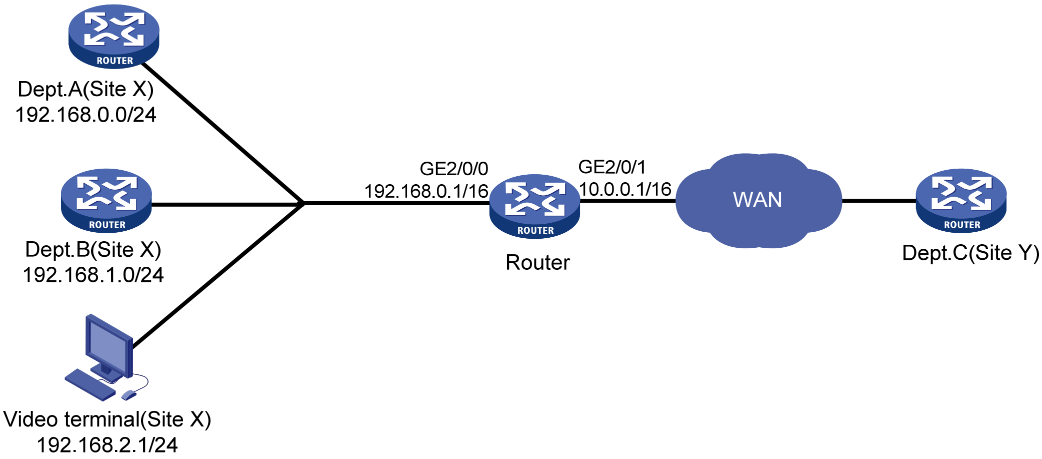
A company has agencies in site X and site Y. The agency in site X has departments A and B. The agency in site Y has one department, C. Site X and site Y are connected through a service provider WAN. Site X uses an 6600/HSR6600 router to connect to site Y through a 100-Mbps Ethernet link provided by the service provider.

Department A is on network segment 192.168.0.0/24 and has 100 employees, whose IP addresses are 192.168.0.2 through 192.168.0.101. Department B is on network segment 192.168.1.0/24 and has 200 employees, whose IP addresses are 192.168.1.1 through 192.168.1.200. Department A and department B only exchange data traffic whose IP precedence is 0 with department C in site Y. Site X has a video terminal, which is dedicated to video conferencing with department C. The video terminal is configured with IP address 192.168.2.1, and sends video traffic in UDP packets whose destination port number is 3000. Configure interface-level hierarchical CAR to reserve bandwidth for video traffic, reserve bandwidth for employees of departments A and B, and allow the idle bandwidth to be shared.

Configure interface-level hierarchical CAR to:

- Reserve 10 Mbps of bandwidth for the video traffic.
- Reserve 300 kbps of bandwidth for each employee of department A, and reserve 30 Mbps of bandwidth in all.
- Reserve 200 kbps of bandwidth for each employee in department B, and reserve 40 Mbps of bandwidth in all.

Figure 59 Network diagram



Configuration procedures

# Configure IP addresses for interfaces according to the network diagram. (Details not shown)

# Configure two CAR lists to match the traffic of the employees of department A and the traffic of the employees of department B, respectively.

```
<Router> system-view
[Router] qos carl 1 source-ip-address range 192.168.0.2 to 192.168.0.101 per-address
[Router] qos carl 2 source-ip-address range 192.168.1.1 to 192.168.1.200 per-address
```

# Configure an ACL to match the video traffic.

```
[Router] acl number 3000 name video
[Router-acl-adv-3000] rule 0 permit udp destination-port eq 3000
[Router-acl-adv-3000] quit
```

# Configure hierarchical CAR on interface GigabitEthernet 2/0/0 of the 6600/HSR6600 router: reserve 10 Mbps of bandwidth for the video traffic, reserve 300 kbps of bandwidth for each employee of department A, and reserve 200 kbps of bandwidth for each employee of department B. Limit the bandwidth to 100 Mbps for all the traffic between site X and site Y.

```
[Router] interface GigabitEthernet 2/0/0
[Router-GigabitEthernet2/0/0] qos car inbound acl 3000 cir 10000 green continue red
continue
[Router-GigabitEthernet2/0/0] qos car inbound carl 1 cir 300 green continue red continue
[Router-GigabitEthernet2/0/0] qos car inbound carl 2 cir 200 green continue red continue
[Router-GigabitEthernet2/0/0]qos car inbound any cir 100000 green pass red discard
```

Configuration example for implementing intelligent load sharing through interface-level hierarchical CAR

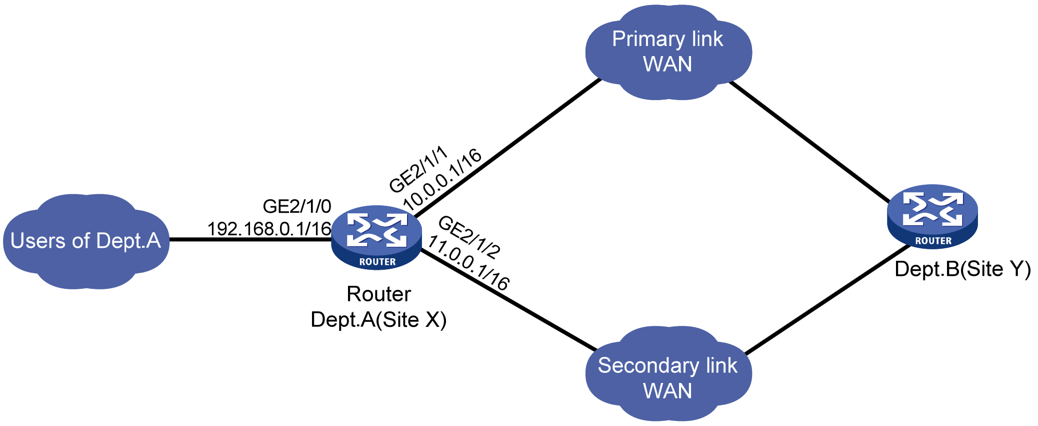
Network requirements

A company has agencies in site X and site Y, respectively. The agency in site X has department A and the agency in site Y has department B. The two departments are connected through two WAN links leased from a service provider, which back up each other. The primary link provides 100 Mbps of bandwidth and is configured with gateway address 10.0.0.2. The secondary link provides 40 Mbps of bandwidth and is configured with gateway address 11.0.0.2. Department A and department B need to transmit two types of traffic: data traffic, and video conferencing traffic with UDP port number 3000. Transmit the video traffic over only the secondary link. When all or partial bandwidth of the secondary link is idle, data traffic can also be transmitted over the secondary link. In this way, both the primary link and secondary link can be 100% utilized.

Configure interface-level hierarchical CAR to:

- Limit the video traffic rate to 30 Mbps, drop the video traffic exceeding 30 Mbps, and transmit the video traffic only over the secondary link.
- Transmit data traffic over the secondary link when the all or partial bandwidth of the secondary link is idle.

Figure 60 Network diagram



Configuration procedures

# Configure IP addresses for interfaces according to the network diagram, and configure the default gateway address as 10.0.0.2 on the router (Site X), so that the packets are transmitted over the primary link by default. (Details not shown)

# Configure ACL 3000 and ACL 3001 to match video traffic and non-video traffic, respectively.

```
<Router> system-view
[Router] acl number 3000 name video
[Router-acl-adv-3000-video] rule 0 permit udp destination-port eq 3000
[Router-acl-adv-3000-video] quit
[Router] acl number 3001 name notvideo
[Router-acl-adv-3001-video] rule 0 deny udp destination-port eq 3000
[Router-acl-adv-3001-video] rule 1 permit ip
[Router-acl-adv-3001-video] quit
```

# Configure hierarchical CAR in the inbound direction of interface GigabitEthernet 2/1/0 to limit the rate of video traffic to 30 Mbps, reserve 10 Mbps of bandwidth for non-video traffic, reserve 40 Mbps of bandwidth for all traffic, and mark the traffic within the specification of 40 Mbps with IP precedence 7, prioritizing the green packets that have been processed by an upper-level CAR.

```
[Router] interface GigabitEthernet 2/1/0
[Router-GigabitEthernet2/1/0] qos car inbound acl 3000 cir 30000 green continue red discard
[Router-GigabitEthernet2/1/0] qos car inbound acl 3001 cir 10000 green continue red
continue
```

```
[Router-GigabitEthernet2/1/0] qos car inbound any cir 40000 green remark-prec-pass 7 red
pass
[Router-GigabitEthernet2/1/0] quit
```

# Configure ACL 3002 to match traffic with IP precedence 7.

```
[Router] acl number 3002
[Router-acl-adv-3002] rule 0 permit ip precedence 7
[Router-acl-adv-3002] quit
```

# Configure a routing policy to transmit the packets with IP precedence 7 over the secondary link, and apply the policy to interface GigabitEthernet 2/1/0.

```
[Router] policy-based-route backup node 1
 % New sequence of this list.
[Router-pbr-backup-1] if-match acl 3002
[Router-pbr-backup-1] apply ip-address next-hop 11.0.0.2
[Router-pbr-backup-1] quit
[Router] interface GigabitEthernet 2/1/0
[Router-GigabitEthernet2/1/0] ip policy-based-route backup
```

# Limit the rate to 100 Mbps on interface GigabitEthernet 2/1/1.

```
[Router] interface GigabitEthernet 2/1/1
[Router-GigabitEthernet2/1/1] qos car outbound any cir 100000 green pass red discard
```

Document conventions and icons

Conventions

This section describes the conventions used in the documentation.

Port numbering in examples

The port numbers in this document are for illustration only and might be unavailable on your device.

Command conventions

Convention	Description
Boldface	**Bold** text represents commands and keywords that you enter literally as shown.
Italic	*Italic* text represents arguments that you replace with actual values.
[ ]	Square brackets enclose syntax choices (keywords or arguments) that are optional.
{ x | y | ... }	Braces enclose a set of required syntax choices separated by vertical bars, from which you select one.
[x | y | ...]	Square brackets enclose a set of optional syntax choices separated by vertical bars, from which you select one or none.
{ x | y | ... } *	Asterisk marked braces enclose a set of required syntax choices separated by vertical bars, from which you select at least one.
[x | y | ...] *	Asterisk marked square brackets enclose optional syntax choices separated by vertical bars, from which you select one choice, multiple choices, or none.
&<1-n>	The argument or keyword and argument combination before the ampersand (&) sign can be entered 1 to n times.
#	A line that starts with a pound (#) sign is comments.

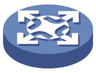
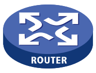
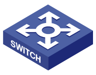
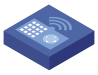
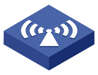
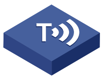
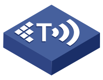
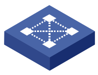
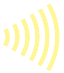
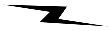
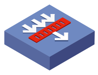
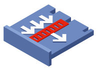
GUI conventions

Convention	Description
Boldface	Window names, button names, field names, and menu items are in Boldface. For example, the **New User** window appears; click **OK**.
>	Multi-level menus are separated by angle brackets. For example, **File** > **Create** > **Folder**.

Symbols

Convention	Description
⚠ **WARNING!**	An alert that calls attention to important information that if not understood or followed can result in personal injury.
△ **CAUTION:**	An alert that calls attention to important information that if not understood or followed can result in data loss, data corruption, or damage to hardware or software.
ⓘ **IMPORTANT:**	An alert that calls attention to essential information.
NOTE:	An alert that contains additional or supplementary information.
💡 **TIP:**	An alert that provides helpful information.

Network topology icons

Convention	Description
	Represents a generic network device, such as a router, switch, or firewall.
	Represents a routing-capable device, such as a router or Layer 3 switch.
	Represents a generic switch, such as a Layer 2 or Layer 3 switch, or a router that supports Layer 2 forwarding and other Layer 2 features.
	Represents an access controller, a unified wired-WLAN module, or the access controller engine on a unified wired-WLAN switch.
	Represents an access point.
	Represents a wireless terminator unit.
	Represents a wireless terminator.
	Represents a mesh access point.
	Represents omnidirectional signals.
	Represents directional signals.
	Represents a security product, such as a firewall, UTM, multiservice security gateway, or load balancing device.
	Represents a security card, such as a firewall, load balancing, NetStream, SSL VPN, IPS, or ACG card.

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  www.hpe.com/assistance

- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates, go to either of the following:
  - Hewlett Packard Enterprise Support Center **Get connected with updates** page:

    www.hpe.com/support/e-updates

  - Software Depot website:

    www.hpe.com/support/softwaredepot

- To view and update your entitlements, and to link your contracts, Care Packs, and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  www.hpe.com/support/AccessToSupportMaterials

IMPORTANT:

Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

Websites

Website	Link
Networking websites	
Hewlett Packard Enterprise Information Library for Networking	www.hpe.com/networking/resourcefinder
Hewlett Packard Enterprise Networking website	www.hpe.com/info/networking
Hewlett Packard Enterprise My Networking website	www.hpe.com/networking/support
Hewlett Packard Enterprise My Networking Portal	www.hpe.com/networking/mynetworking
Hewlett Packard Enterprise Networking Warranty	www.hpe.com/networking/warranty
General websites	
Hewlett Packard Enterprise Information Library	www.hpe.com/info/enterprise/docs
Hewlett Packard Enterprise Support Center	www.hpe.com/support/hpesc
Hewlett Packard Enterprise Support Services Central	ssc.hpe.com/portal/site/ssc/
Contact Hewlett Packard Enterprise Worldwide	www.hpe.com/assistance
Subscription Service/Support Alerts	www.hpe.com/support/e-updates
Software Depot	www.hpe.com/support/softwaredepot
Customer Self Repair (not applicable to all devices)	www.hpe.com/support/selfrepair
Insight Remote Support (not applicable to all devices)	www.hpe.com/info/insightremotesupport/docs

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty, Care Pack Service, or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

For more information and device support details, go to the following website:

www.hpe.com/info/insightremotesupport/docs

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (docsfeedback@hpe.com). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

Index