

ACL and QoS Configuration Guide

This configuration guide is applicable to the following switches and software versions:

H3C S5560S-EI switch series (Release 6309P01 and later)

H3C S5560S-SI switch series (Release 6310 and later)

H3C S5500V3-SI switch series (Release 6310 and later)

H3C MS4520V2 switch series (MS4520V2-28S and MS4520V2-24TP switches) (Release 6310 and later)

H3C WS5850-WiNet switch series (Release 6308P01 and later)

New H3C Technologies Co., Ltd.
<http://www.h3c.com>

Software version: Release 63xx
Document version: 6W101-20200915

Copyright © 2020, New H3C Technologies Co., Ltd. and its licensors

All rights reserved

No part of this manual may be reproduced or transmitted in any form or by any means without prior written consent of New H3C Technologies Co., Ltd.

Trademarks

Except for the trademarks of New H3C Technologies Co., Ltd., any trademarks that may be mentioned in this document are the property of their respective owners.

Notice

The information in this document is subject to change without notice. All contents in this document, including statements, information, and recommendations, are believed to be accurate, but they are presented without warranty of any kind, express or implied. H3C shall not be liable for technical or editorial errors or omissions contained herein.

Preface

This configuration guide describes ACL and QoS fundamentals and configuration procedures, including ACL, QoS, data buffer, and time range.

This preface includes the following topics about the documentation:

- [Audience](#)
- [Conventions](#)
- [Documentation feedback](#)

Audience

This documentation is intended for:

- Network planners.
- Field technical support and servicing engineers.
- Network administrators.

Conventions

The following information describes the conventions used in the documentation.

Command conventions

Convention	Description
Boldface	Bold text represents commands and keywords that you enter literally as shown.
<i>Italic</i>	<i>Italic</i> 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 a minimum of 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 opens; 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 module, such as a firewall, load balancing, NetStream, SSL VPN, IPS, or ACG module.

Examples provided in this document

Examples in this document might use devices that differ from your device in hardware model, configuration, or software version. It is normal that the port numbers, sample output, screenshots, and other information in the examples differ from what you have on your device.

Documentation feedback

You can e-mail your comments about product documentation to info@h3c.com.

We appreciate your comments.

Contents

Configuring ACLs.....	1
About ACLs.....	1
Numbering and naming ACLs.....	1
ACL types.....	1
Match order.....	1
Rule numbering.....	2
Fragment filtering with ACLs.....	3
Restrictions and guidelines: ACL configuration.....	3
ACL tasks at a glance.....	4
Configuring a basic ACL.....	4
About basic ACLs.....	4
Configuring an IPv4 basic ACL.....	4
Configuring an IPv6 basic ACL.....	5
Configuring an advanced ACL.....	5
About advanced ACLs.....	5
Configuring an IPv4 advanced ACL.....	6
Configuring an IPv6 advanced ACL.....	6
Configuring a Layer 2 ACL.....	7
Copying an ACL.....	8
Configuring packet filtering with ACLs.....	8
About packet filtering with ACLs.....	8
Applying an ACL to an interface for packet filtering.....	9
Configuring logging and SNMP notifications for packet filtering.....	9
Setting the packet filtering default action.....	9
Display and maintenance commands for ACL.....	10
ACL configuration examples.....	10
Example: Configuring interface-based packet filter.....	10

Configuring ACLs

About ACLs

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

ACLs are primarily used for packet filtering. You can also use ACLs in QoS, security, routing, and other modules for identifying traffic. The packet drop or forwarding decisions depend on the modules that use ACLs.

Numbering and naming ACLs

When creating an ACL, you must assign it a number or name for identification. You can specify an existing ACL by its number or name. Each ACL type has a unique range of ACL numbers.

For basic or advanced ACLs with the same number, you must use the `ipv6` keyword to distinguish them. For ACLs with the same name, you must use the `ipv6` or `mac` keywords to distinguish them.

ACL types

Type	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.
Layer 2 ACLs	4000 to 4999	IPv4 and IPv6	Layer 2 header fields, such as source and destination MAC addresses, 802.1p priority, and link layer protocol type.

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 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 Sort ACL rules in depth-first order

ACL type	Sequence of tie breakers
IPv4 basic ACL	<ol style="list-style-type: none">1. VPN instance.2. More 0s in the source IPv4 address wildcard (more 0s means a narrower IPv4 address range).3. Rule configured earlier.
IPv4 advanced ACL	<ol style="list-style-type: none">1. VPN instance.2. Specific protocol number.3. More 0s in the source IPv4 address wildcard mask.4. More 0s in the destination IPv4 address wildcard.5. Narrower TCP/UDP service port number range.6. Rule configured earlier.
IPv6 basic ACL	<ol style="list-style-type: none">1. VPN instance.2. Longer prefix for the source IPv6 address (a longer prefix means a narrower IPv6 address range).3. Rule configured earlier.
IPv6 advanced ACL	<ol style="list-style-type: none">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.
Layer 2 ACL	<ol style="list-style-type: none">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 number 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 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.

The rule numbering step sets the increment by which the system numbers rules automatically. If you do not specify a rule ID when creating an ACL rule, the system automatically assigns it a rule ID. This rule ID is the nearest higher multiple of the numbering step to the current highest rule ID, starting from the start rule ID. For example, if the rule numbering step is 5 and the current highest rule ID is 12, the rule is numbered 15.

The wider the numbering step, the more rules you can insert between two rules. Whenever the step or start rule ID changes, the rules are renumbered, starting from the start rule ID. For example, if there are five rules numbered 0, 5, 9, 10, and 15, changing the step from 5 to 2 causes the rules to be renumbered 0, 2, 4, 6, and 8.

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 step is 5, and there are five rules numbered 0, 5, 9, 10, and 12, the newly defined rule is numbered 15. If the ACL does not contain a rule, the first rule is numbered 0.

Whenever the step changes, the rules are renumbered, starting from 0. For example, changing the step from 5 to 2 renumbers rules 5, 10, 13, and 15 as rules 0, 2, 4, and 6.

For an ACL of the match order **auto**, rules are sorted in depth-first order, and are renumbered based on the match order. For example, rules are in the match order of 0, 10, and 5. Changing the numbering step to 2 renumbers rules 0, 10, and 5 (not 0, 5, and 10) as rules 0, 2, 4.

Fragment 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 risks, the ACL feature is designed as follows:

- Filters all fragments by default, including non-first fragments.
- Allows for matching criteria modification for efficiency. For example, you can configure the ACL to filter only non-first fragments.

Restrictions and guidelines: ACL configuration

- If you create a numbered ACL, you can enter the view of the ACL by using either of the following commands:
 - `acl [ipv6] number acl-number.`
 - `acl { [ipv6] { advanced | basic } | mac } acl-number.`
- If you create a ACL by using the `acl [ipv6] number acl-number name acl-name` command, you can enter the view of the ACL by using either of the following commands:
 - `acl [ipv6] name acl-name` (for only basic ACLs and advanced ACLs).
 - `acl [ipv6] number acl-number [name acl-name]`.
 - `acl { [ipv6] { advanced | basic } | mac } name acl-name.`
- If you create a named ACL by using the `acl { [ipv6] { advanced | basic } | mac } name acl-name` command, you can enter the view of the ACL by using either of the following commands:
 - `acl [ipv6] name acl-name` (for only basic ACLs and advanced ACLs).
 - `acl { [ipv6] { advanced | basic } | mac } name acl-name.`
- Matching packets are forwarded through slow forwarding if an ACL rule contains match criteria or has functions enabled in addition to the following match criteria and functions:
 - Source and destination IP addresses.
 - Source and destination ports.
 - Transport layer protocol.
 - ICMP or ICMPv6 message type, message code, and message name.

- VPN instance.
- Logging.
- Time range.

Slow forwarding requires packets to be sent to the control plane for forwarding entry calculation, which affects the device forwarding performance.

ACL tasks at a glance

To configure an ACL, perform the following tasks:

- Configure ACLs according to the characteristics of the packets to be matched
 - [Configuring a basic ACL](#)
 - [Configuring an advanced ACL](#)
 - [Configuring a Layer 2 ACL](#)
- (Optional.) [Copying an ACL](#)
- (Optional.) [Configuring packet filtering with ACLs](#)

Configuring a basic ACL

About basic ACLs

Basic ACLs match packets based only on source IP addresses.

Configuring an IPv4 basic ACL

1. Enter system view.

```
system-view
```

2. Create an IPv4 basic ACL and enter its view. Choose one option as needed:

- Create an IPv4 basic ACL by specifying an ACL number.

```
acl number acl-number [ name acl-name ] [ match-order { auto | config } ]
```

- Create an IPv4 basic ACL by specifying the **basic** keyword.

```
acl basic { acl-number | name acl-name } [ match-order { auto | config } ]
```

3. (Optional.) Configure a description for the IPv4 basic ACL.

```
description text
```

By default, an IPv4 basic ACL does not have a description.

4. (Optional.) Set the rule numbering step.

```
step step-value [ start start-value ]
```

By default, the rule numbering step is 5 and the start rule ID is 0.

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 ] *
```

The **logging** keyword takes effect only when the module (for example, packet filtering) that uses the ACL supports logging.

- (Optional.) Add or edit a rule comment.
`rule rule-id comment text`
 By default, no rule comment is configured.

Configuring an IPv6 basic ACL

Restrictions and guidelines

If an IPv6 basic ACL is used for QoS traffic classification or packet filtering:

- Do not specify the **fragment** keyword.
- Do not specify the **routing** keyword if the ACL is for outbound application.

Procedure

- Enter system view.
`system-view`
- Create an IPv6 basic ACL view and enter its view. Choose one option as needed:
 - Create an IPv6 basic ACL by specifying an ACL number.
`acl ipv6 number acl-number [name acl-name] [match-order { auto | config }]`
 - Create an IPv6 basic ACL by specifying the **basic** keyword.
`acl ipv6 basic { acl-number | name acl-name } [match-order { auto | config }]`
- (Optional.) Configure a description for the IPv6 basic ACL.
`description text`
 By default, an IPv6 basic ACL does not have a description.
- (Optional.) Set the rule numbering step.
`step step-value [start start-value]`
 By default, the rule numbering step is 5 and the start rule ID is 0.
- 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] *`
 The **logging** keyword takes effect only when the module (for example, packet filtering) that uses the ACL supports logging.
- (Optional.) Add or edit a rule comment.
`rule rule-id comment text`
 By default, no rule comment is configured.

Configuring an advanced ACL

About advanced ACLs

Advanced ACLs match packets based on the following criteria:

- Source IP addresses.
- Destination IP addresses.

- Packet priorities.
- Protocol types.
- Other protocol header information, such as TCP/UDP source and destination port numbers, TCP flags, ICMP message types, and ICMP message codes.

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

Configuring an IPv4 advanced ACL

1. Enter system view.

```
system-view
```

2. Create an IPv4 advanced ACL and enter its view. Choose one option as needed:

- Create a numbered IPv4 advanced ACL by specifying an ACL number.

```
acl number acl-number [ name acl-name ] [ match-order { auto | config } ]
```

- Create an IPv4 advanced ACL by specifying the **advanced** keyword.

```
acl advanced { acl-number | name acl-name } [ match-order { auto | config } ]
```

3. (Optional.) Configure a description for the IPv4 advanced ACL.

```
description text
```

By default, an IPv4 advanced ACL does not have a description.

4. (Optional.) Set the rule numbering step.

```
step step-value [ start start-value ]
```

By default, the rule numbering step is 5 and the start rule ID is 0.

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-wildcard | any } | destination-port operator port | { dscp dscp | { precedence precedence | tos tos } * } | fragment | icmp-type { icmp-type [ icmp-code ] | icmp-message } | logging | source { source-address source-wildcard | any } | source-port operator port | time-range time-range-name ] *
```

The **logging** keyword takes effect only when the module (for example, packet filtering) that uses the ACL supports logging.

6. (Optional.) Add or edit a rule comment.

```
rule rule-id comment text
```

By default, no rule comment is configured.

Configuring an IPv6 advanced ACL

Restrictions and guidelines

If an IPv6 advanced ACL is for QoS traffic classification or packet filtering:

- Do not specify the **fragment** keyword.
- Do not specify the **vpn-instance**, **routing**, **hop-by-hop**, or **flow-label** keyword if the ACL is for outbound application.
- Do not specify **ipv6-ah** for the *protocol* argument, or set its value to 0, 43, 44, 51, or 60 if the ACL is for outbound application.

Procedure

1. Enter system view.
system-view
2. Create an IPv6 advanced ACL and enter its view. Choose one option as needed:
 - Create a numbered IPv6 advanced ACL by specifying an ACL number.
acl ipv6 number *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]
 - Create an IPv6 advanced ACL by specifying the **advanced** keyword.
acl ipv6 advanced { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]
3. (Optional.) Configure a description for the IPv6 advanced ACL.
description *text*
By default, an IPv6 advanced ACL does not have a description.
4. (Optional.) Set the rule numbering step.
step *step-value* [**start** *start-value*]
By default, the rule numbering step is 5 and the start rule ID is 0.
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* *port* | | **dscp** *dscp* | **flow-label** *flow-label-value* | **fragment** | **icmp6-type** { *icmp6-type* *icmp6-code* | *icmp6-message* } | **logging** | **routing** [**type** *routing-type*] | **hop-by-hop** [**type** *hop-type*] | **source** { *source-address* *source-prefix* | *source-address/source-prefix* | **any** } | **source-port** *operator* *port* | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *
The **logging** keyword takes effect only when the module (for example, packet filtering) that uses the ACL supports logging.
6. (Optional.) Add or edit a rule comment.
rule *rule-id* **comment** *text*
By default, no rule comment is configured.

Configuring a Layer 2 ACL

About Layer 2 ACLs

Layer 2 ACLs, also called Ethernet frame header ACLs, match packets based on Layer 2 Ethernet header fields, such as:

- Source MAC address.
- Destination MAC address.
- 802.1p priority (VLAN priority).
- Link layer protocol type.
- Encapsulation type.

Procedure

1. Enter system view.
system-view

2. Create a Layer 2 ACL and enter its view. Choose one option as needed:
 - o Create a Layer 2 ACL by specifying an ACL number.


```
acl number acl-number [ name acl-name ] [ match-order { auto | config } ]
```
 - o Create a Layer 2 ACL by specifying the **mac** keyword.


```
acl mac { acl-number | name acl-name } [ match-order { auto | config } ]
```
3. (Optional.) Configure a description for the Layer 2 ACL.


```
description text
```

By default, a Layer 2 ACL does not have a description.
4. (Optional.) Set the rule numbering step.


```
step step-value [ start start-value ]
```

By default, the rule numbering step is 5 and the start rule ID is 0.
5. Create or edit a rule.


```
rule [ rule-id ] { deny | permit } [ cos dot1p | 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 ] *
```
6. (Optional.) Add or edit a rule comment.


```
rule rule-id comment text
```

By default, no rule comment is configured.

Copying an ACL

About 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 uses a different number or name than the source ACL.

Restrictions and guidelines

To successfully copy an ACL, make sure:

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

Procedure

1. Enter system view.


```
system-view
```
2. Copy an existing ACL to create a new ACL.


```
acl [ ipv6 | mac ] copy { source-acl-number | name source-acl-name } to { dest-acl-number | name dest-acl-name }
```

Configuring packet filtering with ACLs

About packet filtering with ACLs

This section describes procedures for using an ACL to filtering packets. For example, you can apply an ACL to an interface to filter incoming or outgoing packets.

Applying an ACL to an interface for packet filtering

Restrictions and guidelines

To the same direction of an interface, you can apply a maximum of three ACLs: one IPv4 ACL, one IPv6 ACL, and one Layer 2 ACL.

The term "interface" in this section collectively refers to Layer 2 Ethernet interfaces and VLAN interfaces.

Procedure

1. Enter system view.
system-view
2. Enter interface view.
interface *interface-type interface-number*
3. Apply an ACL to the interface to filter packets.
packet-filter [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* } { **inbound** | **outbound** } [**hardware-count**]
By default, an interface does not filter packets.

Configuring logging and SNMP notifications for packet filtering

About configuring logging and SNMP notifications for packet filtering

You can configure the ACL module to generate log entries or SNMP notifications for packet filtering and output them to the information center or SNMP module at the output interval. The log entry or notification records the number of matching packets and the matched ACL rules. When the first packet of a flow matches an ACL rule, the output interval starts, and the device immediately outputs a log entry or notification for this packet. When the output interval ends, the device outputs a log entry or notification for subsequent matching packets of the flow.

For more information about the information center and SNMP, see *Network Management and Monitoring Configuration Guide*.

Procedure

1. Enter system view.
system-view
2. Set the interval for outputting packet filtering logs or notifications.
acl { **logging** | **trap** } **interval** *interval*
The default setting is 0 minutes. By default, the device does not generate log entries or SNMP notifications for packet filtering.

Setting the packet filtering default action

1. Enter system view.
system-view
2. Set the packet filtering default action to deny.
packet-filter default deny
By default, the packet filter permits packets that do not match any ACL rule to pass.

Display and maintenance commands for ACL

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display ACL configuration and match statistics.	display acl [ipv6 mac] { <i>acl-number</i> all name acl-name }
Display ACL application information for packet filtering.	display packet-filter interface [<i>interface-type interface-number</i>] [inbound outbound] [slot slot-number]
Display match statistics for packet filtering ACLs.	display packet-filter statistics interface <i>interface-type interface-number</i> { inbound outbound } [[ipv6 mac] { <i>acl-number</i> name acl-name }] [brief]
Display the accumulated statistics for packet filtering ACLs.	display packet-filter statistics sum { inbound outbound } [ipv6 mac] { <i>acl-number</i> name acl-name } [brief]
Display detailed ACL packet filtering information.	display packet-filter verbose interface <i>interface-type interface-number</i> { inbound outbound } [[ipv6 mac] { <i>acl-number</i> name acl-name }] [slot slot-number]
Display QoS and ACL resource usage.	display qos-acl resource [slot slot-number]
Clear match statistics for packet filtering ACLs.	reset packet-filter statistics interface [<i>interface-type interface-number</i>] { inbound outbound } [[ipv6 mac] { <i>acl-number</i> name acl-name }]

ACL configuration examples

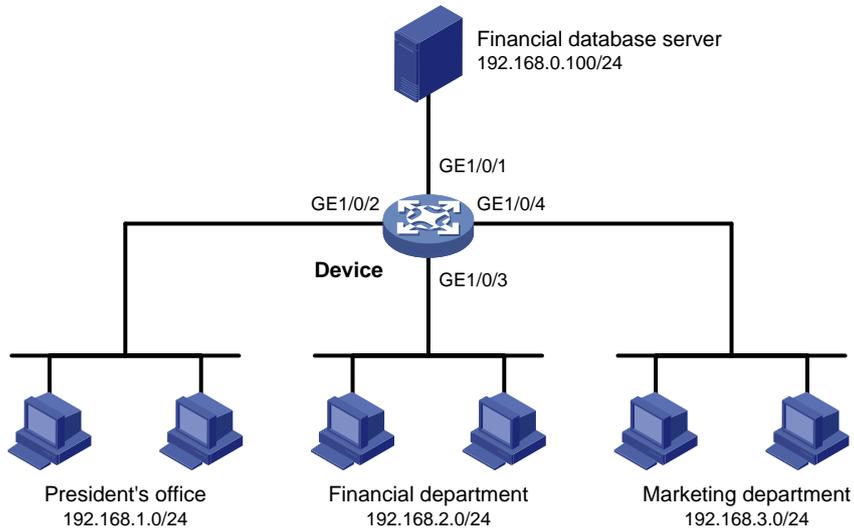
Example: Configuring interface-based packet filter

Network configuration

A company interconnects its departments through the device. Configure a packet filter to:

- Permit access from the President's office at any time to the financial database server.
- Permit access from the Finance 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



Procedure

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

```
<Device> system-view
```

```
[Device] time-range work 08:0 to 18:00 working-day
```

Create an IPv4 advanced ACL numbered 3000.

```
[Device] acl advanced 3000
```

Configure a rule to permit access from the President's office to the financial database server.

```
[Device-acl-ipv4-adv-3000] rule permit ip source 192.168.1.0 0.0.0.255 destination 192.168.0.100 0
```

Configure a rule to permit access from the Finance department to the database server during working hours.

```
[Device-acl-ipv4-adv-3000] rule permit ip source 192.168.2.0 0.0.0.255 destination 192.168.0.100 0 time-range work
```

Configure a rule to deny access to the financial database server.

```
[Device-acl-ipv4-adv-3000] rule deny ip source any destination 192.168.0.100 0
[Device-acl-ipv4-adv-3000] quit
```

Apply IPv4 advanced ACL 3000 to filter outgoing packets on interface GigabitEthernet 1/0/1.

```
[Device] interface gigabitethernet 1/0/1
[Device-GigabitEthernet1/0/1] packet-filter 3000 outbound
[Device-GigabitEthernet1/0/1] quit
```

Verifying the configuration

Verify that a PC in the Finance department can ping the database server 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

Verify that a PC in the Marketing department cannot ping the database server 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),

Display configuration and match statistics for IPv4 advanced ACL 3000 on the device during working hours.

[Device] display acl 3000

Advanced IPv4 ACL 3000, 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
(Active)

rule 10 deny ip destination 192.168.0.100 0

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

Contents

QoS overview	1
QoS service models	1
Best-effort service model	1
IntServ model	1
DiffServ model	1
QoS techniques in a network	1
QoS processing flow in a device	2
QoS configuration approaches	3
Configuring a QoS policy	4
About QoS policies	4
QoS policy tasks at a glance	4
Defining a traffic class	4
Defining a traffic behavior	4
Defining a QoS policy	5
Applying the QoS policy	5
Application destinations	5
Restrictions and guidelines for applying a QoS policy	5
Applying the QoS policy to an interface	5
Applying the QoS policy to VLANs	6
Applying the QoS policy globally	6
Applying the QoS policy to a user profile	7
Display and maintenance commands for QoS policies	7
Configuring priority mapping	9
About priority mapping	9
About priorities	9
Priority maps	9
Priority mapping configuration methods	10
Priority mapping process	10
Priority mapping tasks at a glance	11
Configuring a priority map	11
Configuring a port to trust packet priority for priority mapping	12
Changing the port priority of an interface	12
Display and maintenance commands for priority mapping	13
Priority mapping configuration examples	13
Example: Configuring a priority trust mode	13
Example: Configuring priority mapping tables and priority marking	14
Configuring traffic policing, GTS, and rate limit	18
About traffic policing, GTS, and rate limit	18
Traffic evaluation and token buckets	18
Traffic policing	19
GTS	20
Rate limit	21
Restrictions and guidelines: Traffic policing, GTS, and rate limit configuration	22
Configuring traffic policing	22
Configuring GTS	23
Configuring the rate limit	24
Display and maintenance commands for traffic policing, GTS, and rate limit	24
Traffic policing, GTS, and rate limit configuration examples	24
Example: Configuring traffic policing and GTS	24
Configuring congestion management	28
About congestion management	28
Cause, negative results, and countermeasure of congestion	28
Congestion management methods	28

Congestion management tasks at a glance	30
Configuring queuing on an interface	30
Restrictions and guidelines for queuing configuration	30
Configuring SP queuing	30
Configuring WRR queuing	31
Configuring SP+WRR queuing	31
Configuring a queue scheduling profile	32
About queue scheduling profiles	32
Restrictions and guidelines for queue scheduling profile configuration	32
Configuring a queue scheduling profile	32
Applying a queue scheduling profile	33
Example: Configuring a queue scheduling profile	33
Display and maintenance commands for congestion management	34
Configuring traffic filtering	35
About traffic filtering	35
Restrictions and guidelines: Traffic filtering configuration	35
Procedure	35
Traffic filtering configuration examples	36
Example: Configuring traffic filtering	36
Configuring priority marking	38
About priority marking	38
Configuring priority marking	38
Priority marking configuration examples	39
Example: Configuring priority marking	39
Configuring nesting	42
About nesting	42
Restrictions and guidelines: Nesting configuration	42
Procedure	42
Nesting configuration examples	43
Example: Configuring nesting	43
Configuring traffic redirecting	45
About traffic redirecting	45
Restrictions and guidelines: Traffic redirecting configuration	45
Procedure	45
Traffic redirecting configuration examples	46
Example: Configuring traffic redirecting	46
Configuring global CAR	49
About global CAR	49
Aggregate CAR	49
Hierarchical CAR	49
Restrictions and guidelines: Global CAR configuration	50
Configuring aggregate CAR	50
Display and maintenance commands for global CAR	51
Configuring class-based accounting	52
About class-based accounting	52
Restrictions and guidelines: Class-based accounting configuration	52
Procedure	52
Class-based accounting configuration examples	53
Example: Configuring class-based accounting	53
Appendixes	55
Appendix A Acronyms	55
Appendix B Default priority maps	56
Appendix C Introduction to packet precedence	57
IP precedence and DSCP values	57
802.1p priority	58

QoS overview

In data communications, Quality of Service (QoS) provides differentiated service guarantees for diversified traffic in terms of bandwidth, delay, jitter, and drop rate, all of which can affect QoS.

QoS manages network resources and prioritizes traffic to balance system resources.

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

QoS service models

This section describes several typical QoS service models.

Best-effort service model

The best-effort model is a single-service model. The best-effort model is not as reliable as other models and does not guarantee delay-free delivery.

The best-effort service model is the default model for 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 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. However, it is not suitable for large-sized networks, for example, the core layer of the Internet, where billions of flows are present.

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.

QoS techniques in a network

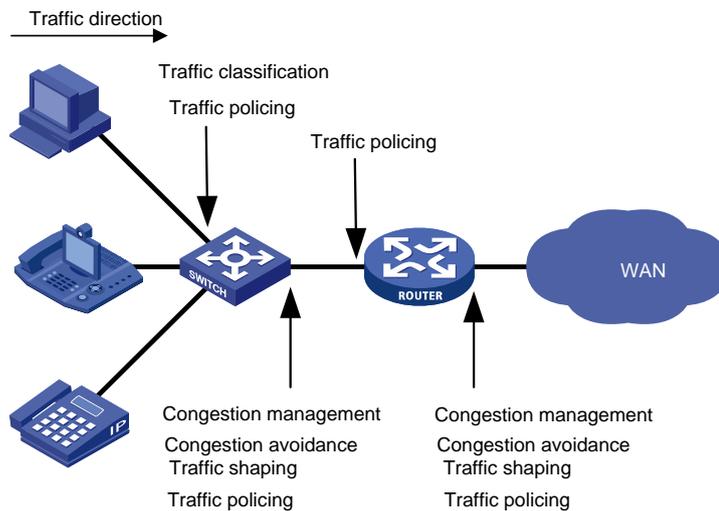
The QoS techniques include the following features:

- Traffic classification.
- Traffic policing.
- Traffic shaping.
- Rate limit.
- Congestion management.
- Congestion avoidance.

The following section briefly introduces these QoS techniques.

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

Figure 1 Position of the QoS techniques in a network



As shown in [Figure 1](#), traffic classification, traffic shaping, traffic policing, congestion management, and congestion avoidance mainly implement the following functions:

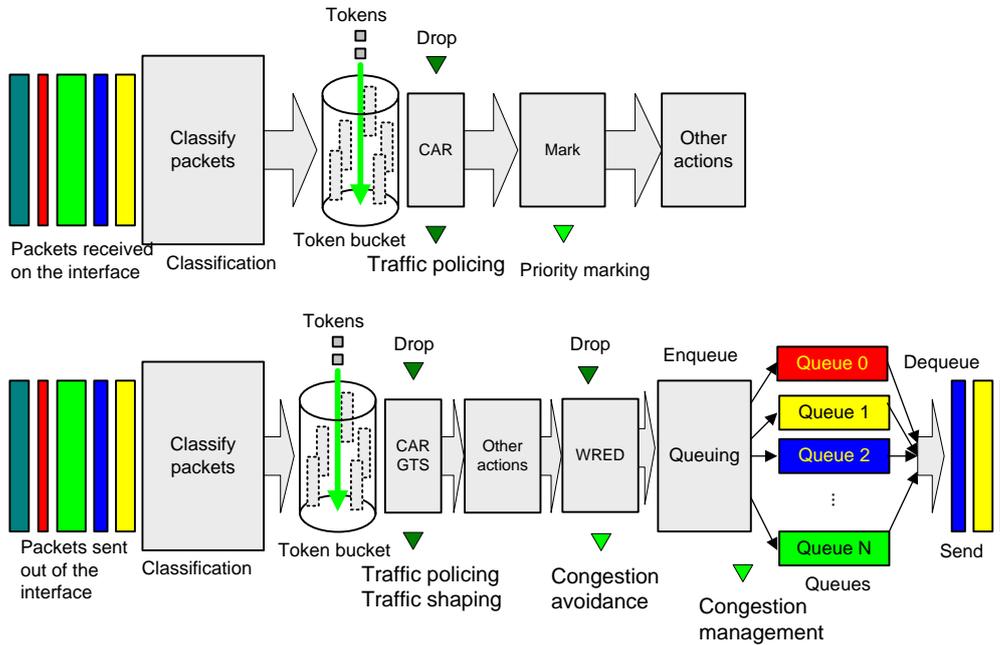
- **Traffic classification**—Uses match criteria to assign packets with the same characteristics to a traffic class. Based on traffic classes, you can provide differentiated services.
- **Traffic policing**—Policing flows and imposes penalties to prevent aggressive use of network resources. You can apply traffic policing to both incoming and outgoing traffic of a port.
- **Traffic shaping**—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. It 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 2](#) 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 the following operations:
 - Traffic policing for incoming traffic.
 - Traffic shaping for outgoing traffic.
 - Congestion avoidance before congestion occurs.
 - Congestion management when congestion occurs.

Figure 2 QoS processing flow



QoS configuration approaches

You can configure QoS by using the MQC approach or non-MQC approach.

In the modular QoS configuration (MQC) approach, you configure QoS service parameters by using QoS policies. A QoS policy defines QoS actions to take on different classes of traffic and can be applied to an object (such as an interface) to control traffic.

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

About QoS policies

A QoS policy has the following components:

- **Traffic class**—Defines criteria to match packets.
- **Traffic behavior**—Defines QoS actions to take on matching packets.

By associating a traffic class with a traffic behavior, a QoS policy can perform the QoS actions on matching packets.

A QoS policy can have multiple class-behavior associations.

QoS policy tasks at a glance

To configure a QoS policy, perform the following tasks:

1. [Defining a traffic class](#)
2. [Defining a traffic behavior](#)
3. [Defining a QoS policy](#)
4. [Applying the QoS policy](#)
 - [Applying the QoS policy to an interface](#)
 - [Applying the QoS policy to VLANs](#)
 - [Applying the QoS policy globally](#)
 - [Applying the QoS policy to a user profile](#)

Defining a traffic class

1. Enter system view.
system-view
2. Create a traffic class and enter traffic class view.
traffic classifier *classifier-name* [**operator** { **and** | **or** }]
3. (Optional.) Configure a description for the traffic class.
description *text*
By default, no description is configured for a traffic class.
4. Configure a match criterion.
if-match *match-criteria*
By default, no match criterion is configured.
For more information, see the **if-match** command in *ACL and QoS Command Reference*.

Defining a traffic behavior

1. Enter system view.
system-view
2. Create a traffic behavior and enter traffic behavior view.

traffic behavior *behavior-name*

3. Configure an action in the traffic behavior.

By default, no action is configured for a traffic behavior.

For more information about configuring an action, see the subsequent chapters for traffic policing, traffic filtering, priority marking, class-based accounting, and so on.

Defining a QoS policy

1. Enter system view.

system-view

2. Create a QoS policy and enter QoS policy view.

qos policy *policy-name*

3. Associate a traffic class with a traffic behavior to create a class-behavior association in the QoS policy.

classifier *classifier-name* **behavior** *behavior-name* [**insert-before** *before-classifier-name*]

By default, a traffic class is not associated with a traffic behavior.

Repeat this step to create more class-behavior associations.

Applying the QoS policy

Application destinations

You can apply a QoS policy to the following destinations:

- **Interface**—The QoS policy takes effect on the traffic sent or received on the interface.
- **VLAN**—The QoS policy takes effect on the traffic sent or received on all ports in the VLAN.
- **Globally**—The QoS policy takes effect on the traffic sent or received on all ports.
- **User profile**—The QoS policy takes effect on the traffic sent or received by the online users of the user profile.

Restrictions and guidelines for applying a QoS policy

You can modify traffic classes, traffic behaviors, and class-behavior associations in a QoS policy even after it is applied (except that it is applied to a user profile). If a traffic class uses an ACL for traffic classification, you can delete or modify the ACL.

Applying the QoS policy to an interface

Restrictions and guidelines

A QoS policy can be applied to multiple interfaces. However, only one QoS policy can be applied to one direction (inbound or outbound) of an interface.

The QoS policy applied to the outgoing traffic on an interface does not regulate local packets. Local packets refer to critical protocol packets sent by the local system for operation maintenance. The most common local packets include link maintenance, RIP, LDP, and SSH packets.

The term "interface" in this section refers to Layer 2 Ethernet interfaces.

Procedure

1. Enter system view.
system-view
2. Enter interface view.
interface *interface-type* *interface-number*
3. Apply the QoS policy to the interface.
qos apply policy *policy-name* { **inbound** | **outbound** }
By default, no QoS policy is applied to an interface.

Applying the QoS policy to VLANs

About QoS policy application to VLANs

You can apply a QoS policy to VLANs to regulate the traffic on all ports of the VLANs.

Restrictions and guidelines

QoS policies cannot be applied to dynamic VLANs, including VLANs created by GVRP.

When you apply a QoS policy to VLANs, the QoS policy is applied to the specified VLANs on all IRF member devices. If the hardware resources of an IRF member device are insufficient, applying a QoS policy to VLANs might fail on the IRF member device. The system does not automatically roll back the QoS policy configuration already applied to other IRF member devices. To ensure consistency, use the **undo qos vlan-policy** *vlan* command to manually remove the QoS policy configuration applied to them.

Procedure

1. Enter system view.
system-view
2. Apply the QoS policy to VLANs.
qos vlan-policy *policy-name* **vlan** *vlan-id-list* { **inbound** | **outbound** }
By default, no QoS policy is applied to a VLAN.

Applying the QoS policy globally

About global QoS policy application

You can apply a QoS policy globally to the inbound or outbound direction of all ports.

Restrictions and guidelines

If the hardware resources of an IRF member device are insufficient, applying a QoS policy globally might fail on the IRF member device. The system does not automatically roll back the QoS policy configuration already applied to other IRF member devices. To ensure consistency, you must use the **undo qos apply policy global** command to manually remove the QoS policy configuration applied to them.

Procedure

1. Enter system view.
system-view
2. Apply the QoS policy globally.
qos apply policy *policy-name* **global** { **inbound** | **outbound** }
By default, no QoS policy is applied globally.

Applying the QoS policy to a user profile

About QoS policy application to a user profile

When a user profile is configured, you can perform traffic policing based on users. After a user passes authentication, the authentication server sends the name of the user profile associated with the user to the device. The QoS policy configured in user profile view takes effect only when users come online.

Restrictions and guidelines

You can apply a QoS policy to multiple user profiles. In one direction of each user profile, only one policy can be applied. To modify a QoS policy already applied to a direction, first remove the applied QoS policy.

Procedure

1. Enter system view.
system-view
 2. Enter user profile view.
user-profile *profile-name*
 3. Apply the QoS policy to the user profile.
qos apply policy *policy-name* { **inbound** | **outbound** }
- By default, no QoS policy is applied to a user profile.

Parameter	Description
inbound	Applies a QoS policy to the traffic received by the device from the user profile.
outbound	Applies a QoS policy to the traffic sent by the device to the user profile.

Display and maintenance commands for QoS policies

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display QoS policy configuration.	display qos policy user-defined [<i>policy-name</i>] [classifier <i>classifier-name</i>] [slot <i>slot-number</i>]
Display information about QoS policies applied globally.	display qos policy global [slot <i>slot-number</i>] [inbound outbound]
Display information about QoS policies applied to interfaces.	display qos policy interface [<i>interface-type</i> <i>interface-number</i>] [inbound outbound]
Display information about QoS policies applied to user profiles.	display qos policy user-profile [name <i>profile-name</i>] [user-id <i>user-id</i>] [slot <i>slot-number</i>] [inbound outbound]
Display information	display qos vlan-policy { name <i>policy-name</i> vlan

Task	Command
about QoS policies applied to VLANs.	[<i>vlan-id</i>] } [slot <i>slot-number</i>] [inbound outbound]
Display QoS and ACL resource usage.	display qos-acl resource [slot <i>slot-number</i>]
Display traffic behavior configuration.	display traffic behavior user-defined [<i>behavior-name</i>] [slot <i>slot-number</i>]
Display traffic class configuration.	display traffic classifier user-defined [<i>classifier-name</i>] [slot <i>slot-number</i>]
Clear the statistics of the QoS policy applied in a certain direction of a VLAN.	reset qos vlan-policy [vlan <i>vlan-id</i>] [inbound outbound]
Clear the statistics for a QoS policy applied globally.	reset qos policy global [inbound outbound]
Clear the statistics for a QoS policy applied globally.	reset qos policy global [inbound outbound]
Clear the statistics of the QoS policy applied in a certain direction of a VLAN.	reset qos vlan-policy [vlan <i>vlan-id</i>] [inbound outbound]

Configuring priority mapping

About priority mapping

When a packet arrives, a device assigns a set of QoS priority parameters to the packet based on either of the following:

- A priority field carried in the packet.
- 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 according to the priority mapping rules. The set of QoS priority parameters decides the scheduling priority and forwarding priority of the packet.

Priority mapping is implemented with priority maps and involves the following priorities:

- 802.1p priority.
- DSCP.
- EXP.
- IP precedence.
- Local precedence.
- Drop priority.

About priorities

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

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

Locally assigned priorities only have local significance. They are assigned by the device only for scheduling. These priorities include the local precedence, drop priority, and user priority, as follows:

- **Local precedence**—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 priority**—Used for making packet drop decisions. Packets with the highest drop priority are dropped preferentially.
- **User priority**—Precedence that the device automatically extracts from a priority field of the packet according to its forwarding path. It is a parameter for determining the scheduling priority and forwarding priority of the packet. The user priority represents the following items:
 - The 802.1p priority for Layer 2 packets.
 - The IP precedence for Layer 3 packets.
 - The EXP for MPLS packets.

The device supports only local precedence for scheduling.

Priority maps

The device provides various types of priority maps. By looking through a priority map, the device decides which priority value to assign to a packet for subsequent packet processing.

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

Priority mapping configuration methods

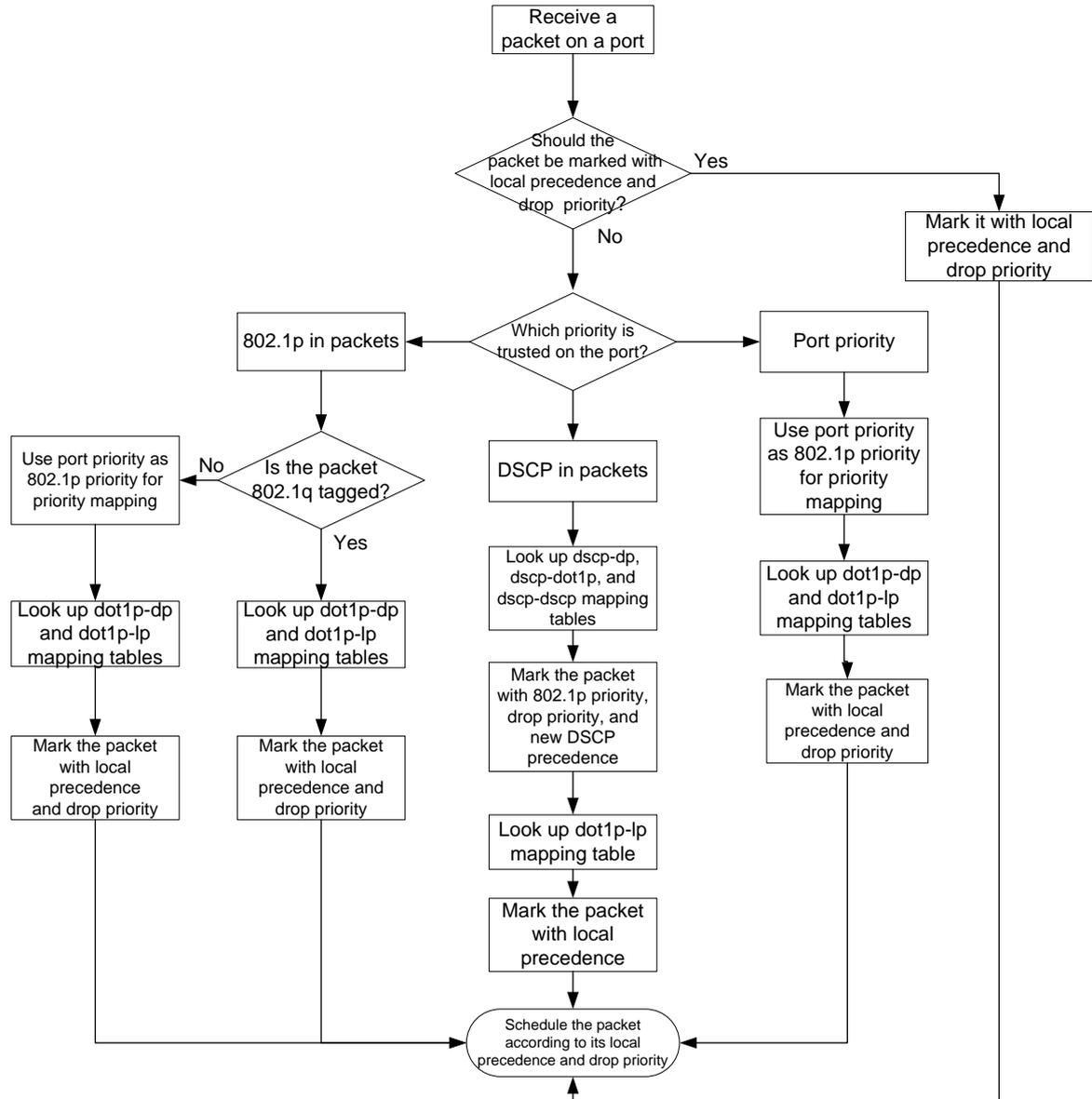
You can configure priority mapping by using any of the following methods:

- **Configuring priority trust mode**—In this method, you can configure a port to look up a trusted priority type (802.1p, for example) in incoming packets in the priority maps. Then, the system maps the trusted priority to the target priority types and values.
- **Changing port priority**—If no packet priority is trusted, the port priority of the incoming port is used. By changing the port priority of a port, you change the priority of the incoming packets on the port.

Priority mapping process

On receiving an Ethernet packet on a port, the switch marks the scheduling priorities (local precedence and drop precedence) for the Ethernet packet. This procedure is done according to the priority trust mode of the receiving port and the 802.1Q tagging status of the packet, as shown in [Figure 3](#).

Figure 3 Priority mapping process for an Ethernet packet



For information about priority marking, see "[Configuring priority marking](#)."

Priority mapping tasks at a glance

To configure priority mapping, perform the following tasks:

1. (Optional.) [Configuring a priority map](#)
2. Configure a priority mapping method:
 - [Configuring a port to trust packet priority for priority mapping](#)
 - [Changing the port priority of an interface](#)

Configuring a priority map

1. Enter system view.

- system-view**
2. Enter priority map view.
qos map-table { **dot1p-lp** | **dscp-dot1p** | **dscp-dscp** }
 3. Configure mappings for the priority map.
import *import-value-list* **export** *export-value*
- By default, the default priority maps are used. For more information, see "[Appendix B Default priority maps.](#)"
- If you execute this command multiple times, the most recent configuration takes effect.

Configuring a port to trust packet priority for priority mapping

About configuring a port to trust packet priority

You can configure the device to trust a particular priority field carried in packets for priority mapping on ports or globally. When you configure the trusted packet priority type on an interface, use the following available keywords:

- **dot1p**—Uses the 802.1p priority of received packets for mapping.
- **dscp**—Uses the DSCP precedence of received IP packets for mapping.

Restrictions and guidelines

The term "interface" in this section refers to Layer 2 Ethernet interfaces.

Procedure

1. Enter system view.
system-view
 2. Enter interface view.
interface *interface-type interface-number*
 3. Configure the trusted packet priority type.
qos trust { **dot1p** | **dscp** }
- By default, an interface does not trust any packet priority and uses the port priority as the 802.1p priority for mapping.

Changing the port priority of an interface

About port priority

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

Procedure

1. Enter system view.
system-view
2. Enter interface view.
interface *interface-type interface-number*
3. Set the port priority of the interface.
qos priority *priority-value*

The default setting is 0.

Display and maintenance commands for priority mapping

Execute `display` commands in any view.

Task	Command
Display priority map configuration.	<code>display qos map-table [dot1p-lp dscp-dot1p dscp-dscp]</code>
Display the trusted packet priority type on a port.	<code>display qos trust interface [interface-type interface-number]</code>

Priority mapping configuration examples

Example: Configuring a priority trust mode

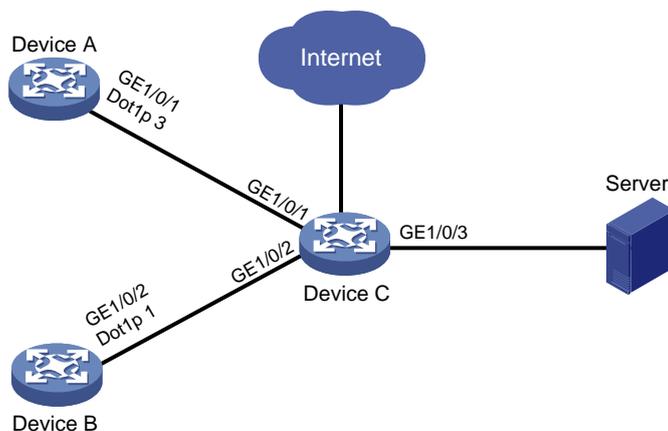
Network configuration

As shown in [Figure 4](#):

- The 802.1p priority of traffic from Device A to Device C is 3.
- The 802.1p priority of traffic from Device B to Device C is 1.

Configure Device C to preferentially process packets from Device A to the server when GigabitEthernet 1/0/3 of Device C is congested.

Figure 4 Network diagram



Procedure

(Method 1) Configure Device C to trust packet priority

Configure GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2 to trust the 802.1p priority for priority mapping.

```
<DeviceC> system-view
[DeviceC] interface gigabitethernet 1/0/1
[DeviceC-GigabitEthernet1/0/1] qos trust dot1p
```

```
[DeviceC-GigabitEthernet1/0/1] quit
[DeviceC] interface gigabitethernet 1/0/2
[DeviceC-GigabitEthernet1/0/2] qos trust dot1p
[DeviceC-GigabitEthernet1/0/2] quit
```

(Method 2) Configure Device C to trust port priority

Assign port priority to GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2. Make sure the following requirements are met:

- The priority of GigabitEthernet 1/0/1 is higher than that of GigabitEthernet 1/0/2.
- No trusted packet priority type is configured on GigabitEthernet 1/0/1 or GigabitEthernet 1/0/2.

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

Example: Configuring priority mapping tables and priority marking

Network configuration

As shown in [Figure 5](#):

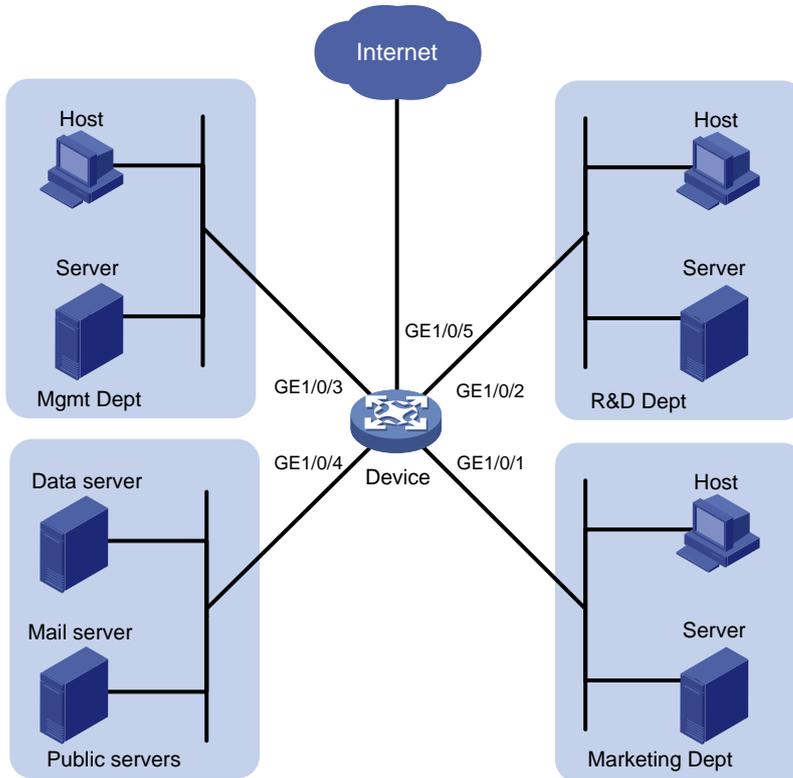
- The Marketing department connects to GigabitEthernet 1/0/1 of Device, which sets the 802.1p priority of traffic from the Marketing department to 3.
- The R&D department connects to GigabitEthernet 1/0/2 of Device, which sets the 802.1p priority of traffic from the R&D department to 4.
- The Management department connects to GigabitEthernet 1/0/3 of Device, 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 1](#).

Table 1 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	Medium
		Marketing department	2	Low
Internet	Management department > Marketing department > R&D department	R&D department	2	Low
		Management department	6	High
		Marketing department	4	Medium

Figure 5 Network diagram



Procedure

1. Configure trusting port priority:

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

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

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

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

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

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

2. 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 servers.

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

3. Configure priority marking to mark the packets from Management department, Marketing department, and R&D department to the Internet with 802.1p priority values 4, 5, and 3.

This guarantees the Management department, Marketing department, and R&D department decreased priorities to access the Internet.

Create ACL 3000, and configure a rule to match HTTP packets.

```
[Device] acl advanced 3000
[Device-acl-adv-3000] rule permit tcp destination-port eq 80
[Device-acl-adv-3000] quit
```

Create a traffic class named **http**, and use ACL 3000 as a match criterion.

```
[Device] traffic classifier http
[Device-classifier-http] if-match acl 3000
[Device-classifier-http] quit
```

Create a traffic behavior named **admin**, and configure a marking action for the Management department.

```
[Device] traffic behavior admin
[Device-behavior-admin] remark dot1p 4
[Device-behavior-admin] quit
```

Create a QoS policy named **admin**, and associate traffic class **http** with traffic behavior **admin** in QoS policy **admin**.

```
[Device] qos policy admin
[Device-qospolicy-admin] classifier http behavior admin
[Device-qospolicy-admin] quit
```

Apply QoS policy **admin** to the inbound direction of GigabitEthernet 1/0/3.

```
[Device] interface gigabitethernet 1/0/3
[Device-GigabitEthernet1/0/3] qos apply policy admin inbound
```

Create a traffic behavior named **market**, and configure a marking action for the Marketing department.

```
[Device] traffic behavior market
[Device-behavior-market] remark dot1p 5
[Device-behavior-market] quit
```

Create a QoS policy named **market**, and associate traffic class **http** with traffic behavior **market** in QoS policy **market**.

```
[Device] qos policy market
[Device-qospolicy-market] classifier http behavior market
[Device-qospolicy-market] quit
```

Apply QoS policy **market** to the inbound direction of GigabitEthernet 1/0/1.

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

Create a traffic behavior named **rd**, and configure a marking action for the R&D department.

```
[Device] traffic behavior rd
[Device-behavior-rd] remark dot1p 3
[Device-behavior-rd] quit
```

Create a QoS policy named **rd**, and associate traffic class **http** with traffic behavior **rd** in QoS policy **rd**.

```
[Device] qos policy rd
[Device-qospolicy-rd] classifier http behavior rd
[Device-qospolicy-rd] quit
```

Apply QoS policy **rd** to the inbound direction of GigabitEthernet 1/0/2.

```
[Device] interface gigabitethernet 1/0/2  
[Device-GigabitEthernet1/0/2] qos apply policy rd inbound
```

Configuring traffic policing, GTS, and rate limit

About traffic policing, GTS, and rate limit

Traffic limit helps assign network resources (including bandwidth) and increase network performance. 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 the traffic rate and resource usage according to traffic 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 (called conforming traffic).
- The corresponding tokens are taken away from the bucket.

Otherwise, the traffic does not conform to the specification (called excess traffic).

A token bucket has the following configurable parameters:

- Mean rate at which tokens are put into the bucket, which is 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. It is 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.

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. The following are main mechanisms used for complicated evaluation:

- **Single rate two color**—Uses one token bucket and 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.

When a packet arrives, the following rules apply:

- If bucket C has enough tokens to forward the packet, the packet is colored green.
- Otherwise, the packet is colored red.
- **Single rate three color**—Uses two token buckets and 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.
- **EBS**—Size of bucket E minus size of bucket C, which specifies the transient burst of traffic that bucket E can forward. The EBS cannot be 0. The size of E bucket is the sum of the CBS and EBS.

When a packet arrives, the following rules apply:

- If bucket C has enough tokens, the packet is colored green.
- If bucket C does not have enough tokens but bucket E has enough tokens, the packet is colored yellow.
- If neither bucket C nor bucket E has sufficient tokens, the packet is colored red.
- **Two rate three color**—Uses two token buckets and 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.
 - **PIR**—Rate at which tokens are put into bucket E, which specifies the average packet transmission or forwarding rate allowed by bucket E.
 - **EBS**—Size of bucket E, which specifies the transient burst of traffic that bucket E can forward.

When a packet arrives, the following rules apply:

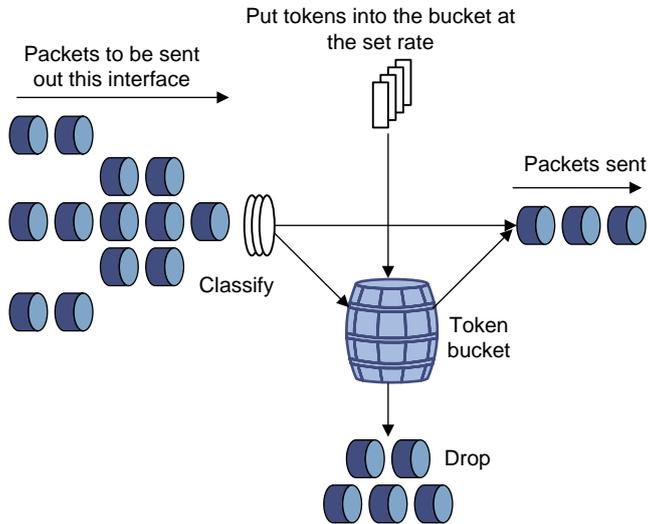
- If bucket C has enough tokens, the packet is colored green.
- If bucket C does not have enough tokens but bucket E has enough tokens, the packet is colored yellow.
- If neither bucket C nor bucket E has sufficient tokens, the packet is colored red.

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 traffic entering a network and limit it within a reasonable range. Another application is to "discipline" the extra traffic to prevent aggressive use of network resources by an application. For example, you can limit bandwidth for HTTP packets to less than 50% of the total. If the traffic of a session exceeds the limit, traffic policing can drop the packets or reset the IP precedence of the packets. [Figure 6](#) shows an example of policing outbound traffic on an interface.

Figure 6 Traffic policing



Traffic policing is widely used in policing traffic entering the ISP networks. 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 precedence re-marked if the evaluation result is "conforming."

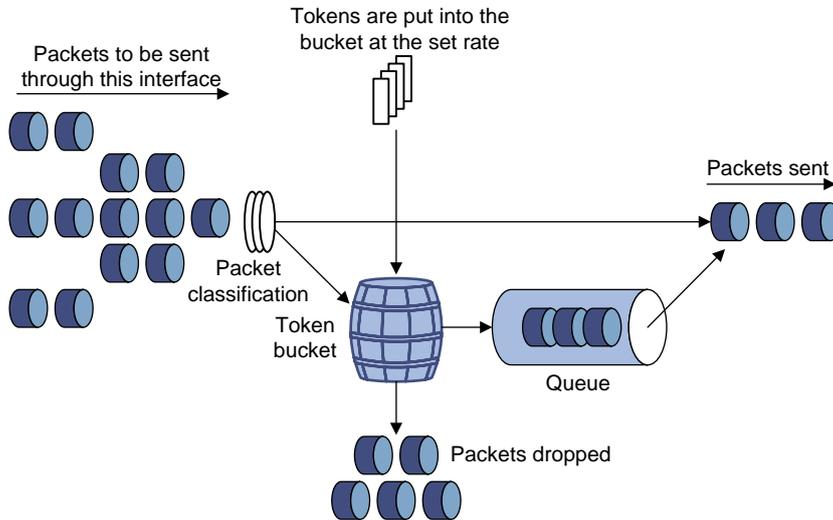
GTS

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

The differences between traffic policing and GTS are as follows:

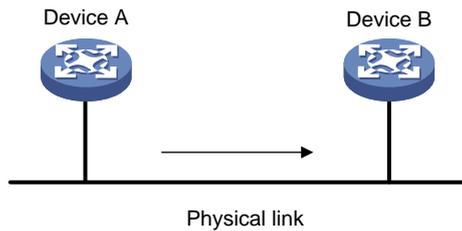
- Packets to be dropped with traffic policing are retained in a buffer or queue with GTS, as shown in [Figure 7](#). When enough tokens are in the token bucket, the buffered packets are sent at an even rate.
- GTS can result in additional delay and traffic policing does not.

Figure 7 GTS



For example, in [Figure 8](#), Device B performs traffic policing on packets from Device A and drops packets exceeding the limit. To avoid packet loss, you can perform GTS on the outgoing interface of Device A so that packets exceeding the limit are cached in Device A. Once resources are released, GTS takes out the cached packets and sends them out.

Figure 8 GTS application



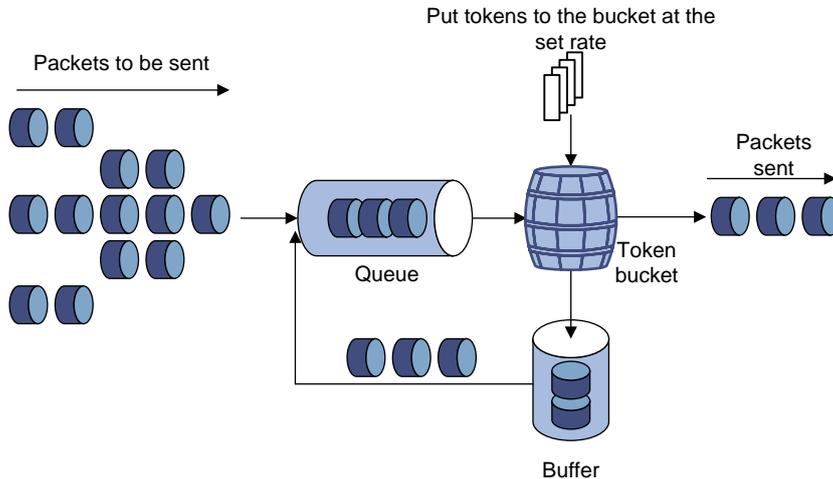
Rate limit

Rate limit controls the rate of inbound and outbound traffic. The outbound traffic is taken for example.

The rate limit of an interface specifies the maximum rate for forwarding packets (excluding critical packets).

Rate limit also uses token buckets for traffic control. When rate limit is configured on an interface, a token bucket handles all packets to be sent through the interface 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 interface is controlled.

Figure 9 Rate limit implementation



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 sufficient 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 an interface. It is easier to use than traffic policing in controlling the total traffic rate.

Restrictions and guidelines: Traffic policing, GTS, and rate limit configuration

The specified CIR does not take traffic transmitted in interframe gaps into account, and the actually allowed rate on an interface is greater than the specified CIR.

An interframe gap is a time interval for transmitting 12 bits between frames. This gap serves the following roles:

- Allows the device to differentiate one frame from another.
- Allows for time for the device to process the current frame and to prepare for receiving the next frame.

Configuring traffic policing

Restrictions and guidelines

The device supports the following application destinations for traffic policing:

- Interface.
- VLANs.
- Globally.
- User profile.

Procedure

1. Enter system view.
system-view
2. Define a traffic class.

- a. Create a traffic class and enter traffic class view.
`traffic classifier classifier-name [operator { and | or }]`
- b. Configure a match criterion.
`if-match match-criteria`
 By default, no match criterion is configured.
 For more information about the `if-match` command, see *ACL and QoS Command Reference*.
- c. Return to system view.
`quit`
3. Define a traffic behavior.
 - a. Create a traffic behavior and enter traffic behavior view.
`traffic behavior behavior-name`
 - b. Configure a traffic policing action.
`car cir committed-information-rate [cbs committed-burst-size [ebs excess-burst-size]] [green action | red action | yellow action] *`
`car cir committed-information-rate [cbs committed-burst-size] pir peak-information-rate [ebs excess-burst-size] [green action | red action | yellow action] *`
 By default, no traffic policing action is configured.
 - c. Return to system view.
`quit`
4. Define a QoS policy.
 - a. Create a QoS policy and enter QoS policy view.
`qos policy policy-name`
 - b. Associate the traffic class with the traffic behavior in the QoS policy.
`classifier classifier-name behavior behavior-name`
 By default, a traffic class is not associated with a traffic behavior.
 - c. Return to system view.
`quit`
5. Apply the QoS policy.
 For more information, see "[Applying the QoS policy.](#)"
 By default, no QoS policy is applied.

Configuring GTS

Restrictions and guidelines

The term "interface" in this section refers to Layer 2 Ethernet interfaces.

Procedure

1. Enter system view.
`system-view`
2. Enter interface view.
`interface interface-type interface-number`
3. Configure GTS for a queue.

```
qos gts queue queue-id cir committed-information-rate [ cbs
committed-burst-size ]
```

```
undo qos gts queue queue-id
```

By default, GTS is not configured on an interface.

Configuring the rate limit

Restrictions and guidelines

The term "interface" in this section refers to Layer 2 Ethernet interfaces.

Procedure

1. Enter system view.

```
system-view
```

2. Enter interface view.

```
interface interface-type interface-number
```

3. Configure the rate limit for the interface.

```
qos lr { inbound | outbound } cir committed-information-rate [ cbs
committed-burst-size ]
```

By default, no rate limit is configured on an interface.

Display and maintenance commands for traffic policing, GTS, and rate limit

Execute **display** commands in any view.

Task	Command
Display GTS configuration and statistics for interfaces.	display qos gts interface [<i>interface-type</i> <i>interface-number</i>]
Display rate limit configuration and statistics.	display qos lr interface [<i>interface-type</i> <i>interface-number</i>]
Display QoS and ACL resource usage.	display qos-acl resource [<i>slot slot-number</i>]
Display traffic behavior configuration.	display traffic behavior user-defined [<i>behavior-name</i>]

Traffic policing, GTS, and rate limit configuration examples

Example: Configuring traffic policing and GTS

Network requirements

As shown in [Figure 10](#):

- The server, Host A, and Host B can access the Internet through Device A and Device B.

- The server, Host A, and GigabitEthernet 1/0/1 of Device A are in the same network segment.
- Host B and GigabitEthernet 1/0/2 of Device A are in the same network segment.

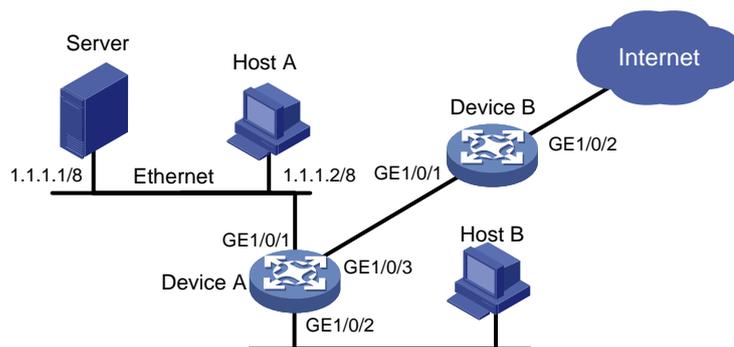
Perform traffic control for the packets that GigabitEthernet 1/0/1 of Device A receives from the server and Host A using the following guidelines:

- Limit the rate of packets from the server to 10240 kbps. When the traffic rate is below 10240 kbps, the traffic is forwarded. When the traffic rate exceeds 10240 kbps, the excess packets are marked with DSCP value 0 and then forwarded.
- Limit the rate of packets from Host A to 2560 kbps. When the traffic rate is below 2560 kbps, the traffic is forwarded. When the traffic rate exceeds 2560 kbps, the excess packets are dropped.

Perform traffic control on GigabitEthernet 1/0/1 and GigabitEthernet 1/0/2 of Device B using the following guidelines:

- Limit the incoming traffic rate on GigabitEthernet 1/0/1 to 20480 kbps, and the excess packets are dropped.
- Limit the outgoing traffic rate on GigabitEthernet 1/0/2 to 10240 kbps, and the excess packets are dropped.

Figure 10 Network diagram



Configuration procedure

1. Configure Device A:

Configure ACL 2001 and ACL 2002 to permit the packets from the server and Host A, respectively.

```
[DeviceA] acl basic 2001
[DeviceA-acl-ipv4-basic-2001] rule permit source 1.1.1.1 0
[DeviceA-acl-ipv4-basic-2001] quit
[DeviceA] acl basic 2002
[DeviceA-acl-ipv4-basic-2002] rule permit source 1.1.1.2 0
[DeviceA-acl-ipv4-basic-2002] quit
```

Create a traffic class named **server**, and use ACL 2001 as the match criterion.

```
[DeviceA] traffic classifier server
[DeviceA-classifier-server] if-match acl 2001
[DeviceA-classifier-server] quit
```

Create a traffic class named **host**, and use ACL 2002 as the match criterion.

```
[DeviceA] traffic classifier host
[DeviceA-classifier-host] if-match acl 2002
[DeviceA-classifier-host] quit
```

Create a traffic behavior named **server**, and configure a traffic policing action (CIR 10240 kbps).

```
[DeviceA] traffic behavior server
```

```
[DeviceA-behavior-server] car cir 10240 red remark-dscp-pass 0
[DeviceA-behavior-server] quit
```

Create a traffic behavior named **host**, and configure a traffic policing action (CIR 2560 kbps).

```
[DeviceA] traffic behavior host
[DeviceA-behavior-host] car cir 2560
[DeviceA-behavior-host] quit
```

Create a QoS policy named **car**, and associate traffic classes **server** and **host** with traffic behaviors **server** and **host** in QoS policy **car**, respectively.

```
[DeviceA] qos policy car
[DeviceA-qospolicy-car] classifier server behavior server
[DeviceA-qospolicy-car] classifier host behavior host
[DeviceA-qospolicy-car] quit
```

Apply QoS policy **car** to the inbound direction of GigabitEthernet 1/0/1.

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

2. Configure Device B:

Create ACL 3001, and configure a rule to match HTTP packets.

```
<DeviceB> system-view
[DeviceB] acl advanced 3001
[DeviceB-acl-adv-3001] rule permit tcp destination-port eq 80
[DeviceB-acl-adv-3001] quit
```

Create a traffic class named **http**, and use ACL 3001 as a match criterion.

```
[DeviceB] traffic classifier http
[DeviceB-classifier-http] if-match acl 3001
[DeviceB-classifier-http] quit
```

Create a traffic class named **class**, and configure the traffic class to match all packets.

```
[DeviceB] traffic classifier class
[DeviceB-classifier-class] if-match any
[DeviceB-classifier-class] quit
```

Create a traffic behavior named **car_inbound**, and configure a traffic policing action (CIR 20480 kbps).

```
[DeviceB] traffic behavior car_inbound
[DeviceB-behavior-car_inbound] car cir 20480
[DeviceB-behavior-car_inbound] quit
```

Create a traffic behavior named **car_outbound**, and configure a traffic policing action (CIR 10240 kbps).

```
[DeviceB] traffic behavior car_outbound
[DeviceB-behavior-car_outbound] car cir 10240
[DeviceB-behavior-car_outbound] quit
```

Create a QoS policy named **car_inbound**, and associate traffic class **class** with traffic behavior **car_inbound** in QoS policy **car_inbound**.

```
[DeviceB] qos policy car_inbound
[DeviceB-qospolicy-car_inbound] classifier class behavior car_inbound
[DeviceB-qospolicy-car_inbound] quit
```

Create a QoS policy named **car_outbound**, and associate traffic class **http** with traffic behavior **car_outbound** in QoS policy **car_outbound**.

```
[DeviceB] qos policy car_outbound
[DeviceB-qospolicy-car_outbound] classifier http behavior car_outbound
```

```
[DeviceB-qospolicy-car_outbound] quit
# Apply QoS policy car_inbound to the inbound direction of GigabitEthernet 1/0/1.
[DeviceB] interface gigabitethernet 1/0/1
[DeviceB-GigabitEthernet1/0/1] qos apply policy car_inbound inbound
# Apply QoS policy car_outbound to the outbound direction of GigabitEthernet 1/0/2.
[DeviceB] interface gigabitethernet 1/0/2
[DeviceB-GigabitEthernet1/0/2] qos apply policy car_outbound outbound
```

Configuring congestion management

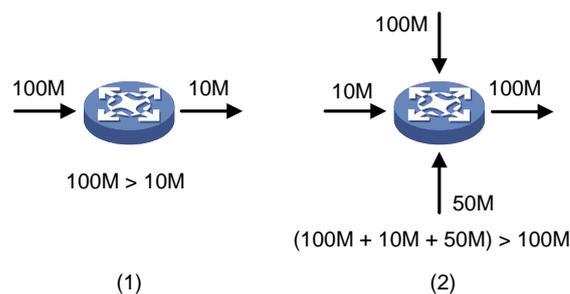
About congestion management

Cause, negative results, and countermeasure 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 11 shows two typical congestion scenarios.

Figure 11 Traffic congestion scenarios



Congestion produces 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 multiuser application environments. To improve the service performance of your network, take measures to manage and control it.

The key to congestion management is defining a resource dispatching policy to prioritize packets for forwarding when congestion occurs.

Congestion management methods

Congestion management uses queuing and scheduling algorithms to classify and sort traffic leaving a port.

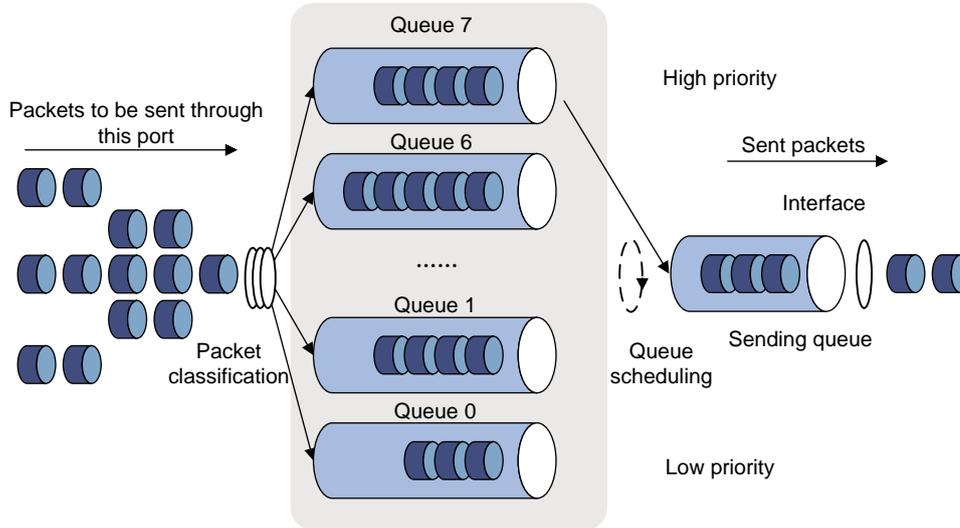
The device supports the following queuing mechanisms:

- SP.
- WRR.

SP queuing

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

Figure 12 SP queuing



In [Figure 12](#), SP queuing classifies eight queues on a port into eight classes, numbered 7 to 0 in descending priority order.

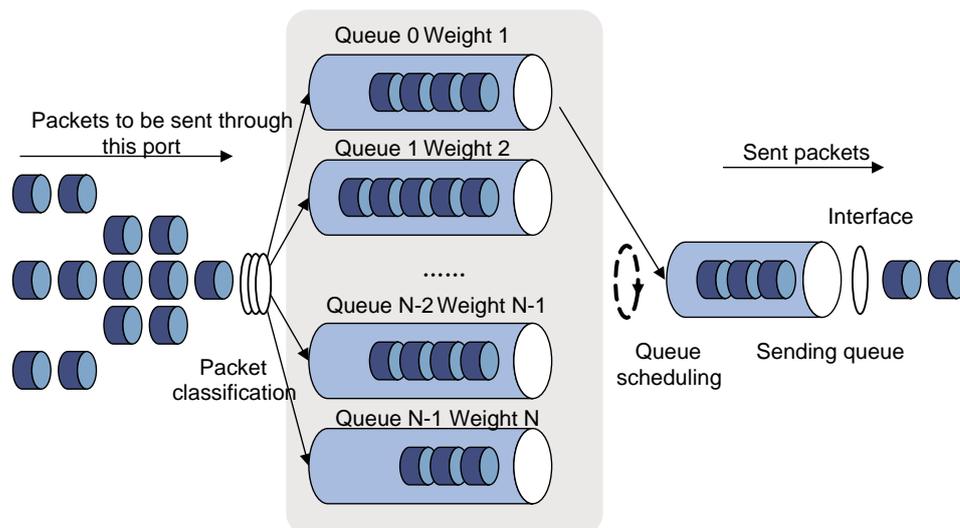
SP queuing schedules the eight queues in the descending order of priority. SP queuing sends packets in the queue with the highest priority first. When the queue with the highest priority is empty, it sends packets in the queue with the second highest priority, and so on. You can assign mission-critical packets to a high priority queue to make sure they are always served first. Common service packets can be assigned to low priority queues to be transmitted when 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. In the worst case, lower priority traffic might never get serviced.

WRR queuing

WRR queuing schedules all the queues in turn to ensure that every queue is served for a certain time, as shown in [Figure 13](#).

Figure 13 WRR queuing



Assume a port provides eight output queues. WRR assigns each queue a weight value (represented by w_7 , w_6 , w_5 , w_4 , w_3 , w_2 , w_1 , or w_0). The weight value of a queue decides the proportion of resources assigned to the queue. On a 100 Mbps port, you can set the weight values to 50, 30, 10, 10, 50, 30, 10, and 10 for w_7 through w_0 . In this way, the queue with the lowest priority can get a minimum of 5 Mbps of bandwidth. WRR solves the problem that SP queuing might fail to serve packets in low-priority queues 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 set the weight 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 output queues 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. Only WRR priority queue group 1 is supported in the current software version.

On an interface enabled with group-based WRR queuing, you can assign queues to the SP group. Queues in the SP group are scheduled with SP. The SP group has higher scheduling priority than the WRR groups.

Congestion management tasks at a glance

To configure congestion management, perform the following tasks:

- [Configuring queuing on an interface](#)
 - [Configuring SP queuing](#)
 - [Configuring WRR queuing](#)
 - [Configuring SP+WRR queuing](#)
- [Configuring a queue scheduling profile](#)

Configuring queuing on an interface

Restrictions and guidelines for queuing configuration

The term "interface" in this section refers to Layer 2 and Layer 3 Ethernet interfaces.

The queue ID, queue name, group, and weight in the `display qos queue interface` command output form a queue scheduling template. A queue scheduling template corresponds to a unique combination of queue parameter settings on an interface.

The device supports a maximum of eight queue scheduling templates, including the default queue scheduling template, the queue scheduling template for IRF physical interfaces, and predefined queue scheduling template for the CPU. If multiple interfaces use the same user-created queue scheduling template, make sure at least one other queue scheduling template has not been used on any interface.

If all queue scheduling templates are used, you can configure congestion management through a queue scheduling profile (see "[Configuring a queue scheduling profile](#)").

Configuring SP queuing

1. Enter system view.

- system-view**
- 2. Enter interface view.
interface *interface-type interface-number*
- 3. Configure SP queuing.
qos sp
By default, an interface uses byte-count WRR queuing.

Configuring WRR queuing

- 1. Enter system view.
system-view
- 2. Enter interface view.
interface *interface-type interface-number*
- 3. Enable WRR queuing.
qos wrr weight
By default, an interface uses packet-count WRR queuing.
- 4. Assign a queue to a WRR group, and configure scheduling parameters for the queue.
qos wrr *queue-id* **group 1 weight** *schedule-value*
By default, all queues on a WRR-enabled interface are in WRR group 1, and queues 0 through 7 have a weight of 1, 2, 3, 4, 5, 9, 13, and 15, respectively.

Configuring SP+WRR queuing

Restrictions and guidelines

To configure the scheduling weight, you must specify the same scheduling unit as specified when enabling WRR queuing.

Procedure

- 1. Enter system view.
system-view
- 2. Enter interface view.
interface *interface-type interface-number*
- 3. Enable byte-count or packet-count WRR queuing.
qos wrr weight
By default, an interface uses packet-count WRR queuing.
- 4. Assign a queue to the SP group.
qos wrr *queue-id* **group sp**
By default, all queues on a WRR-enabled interface are in WRR group 1.
- 5. Assign a queue to a WRR group, and configure a scheduling weight for the queue.
qos wrr *queue-id* **group 1 weight** *schedule-value*
By default, all queues on a WRR-enabled interface are in WRR group 1, and queues 0 through 7 have a weight of 1, 2, 3, 4, 5, 9, 13, and 15, respectively.

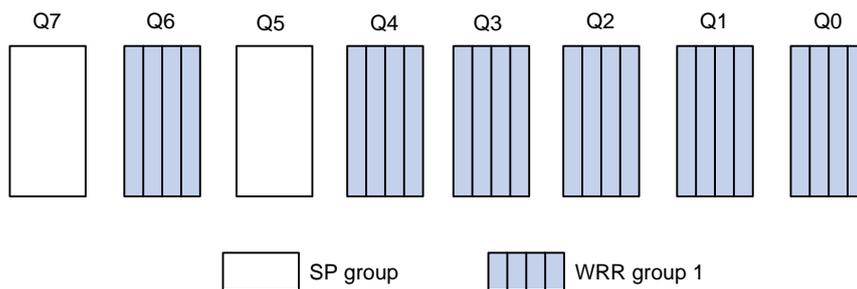
Configuring a queue scheduling profile

About queue scheduling profiles

In a queue scheduling profile, you can configure scheduling parameters for each queue. By applying the queue scheduling profile to an interface, you can implement congestion management on the interface.

Queue scheduling profiles support two queue scheduling algorithms: SP and WRR. In a queue scheduling profile, you can also configure SP+WRR. For information about each scheduling algorithm, see "About congestion management." When SP and WRR groups are configured in a queue scheduling profile, Figure 14 shows the scheduling order.

Figure 14 Queue scheduling profile configured with both SP and WRR



- Queue 7 has the highest priority in the SP group. Its packets are sent preferentially.
- Queue 5 has the second highest priority in the SP group. Packets in queue 5 are sent when queue 7 is empty.
- All queues in WRR group 1 are scheduled according to their weights. When queue 7 and queue 5 are empty, WRR group 1 is scheduled.

Restrictions and guidelines for queue scheduling profile configuration

When you configure a queue scheduling profile, follow these restrictions and guidelines:

- The term "interface" in this section refers to Layer 2 Ethernet interfaces.
- Only one queue scheduling profile can be applied to an interface.
- You can modify the scheduling parameters in a queue scheduling profile already applied to an interface.

Configuring a queue scheduling profile

1. Enter system view.
`system-view`
2. Create a queue scheduling profile and enter queue scheduling profile view.
`qos qmprofile profile-name`
3. (Optional.) Configure queue scheduling parameters.
 - Configure a queue to use SP.
`queue queue-id sp`
 - Configure a queue to use WRR.

```
queue queue-id wrr group group-id { weight | byte-count }
schedule-value
```

By default, all queues in a queue scheduling profile use SP queuing.

The default setting is 64 kbps.

Applying a queue scheduling profile

1. Enter system view.
2. Enter queue scheduling profile view.
3. Execute the following commands in sequence to apply the queue scheduling profile to the outbound direction of an interface:

```
interface interface-type interface-number
qos apply qmprofile profile-name
```

By default, no queue scheduling profile is applied to an interface.

Example: Configuring a queue scheduling profile

Network configuration

Configure a queue scheduling profile to meet the following requirements on GigabitEthernet 1/0/1:

- Queue 7 has the highest priority, and its packets are sent preferentially.
- Queue 0 through queue 6 are in the WRR group and are scheduled according to their packet-count weights, which are 2, 1, 2, 4, 6, 8, and 10, respectively. When queue 7 is empty, the WRR group is scheduled.

Procedure

Enter system view.

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

Create a queue scheduling profile named **qm1**.

```
[Sysname] qos qmprofile qm1
```

```
[Sysname-qmprofile-qm1]
```

Configure queue 7 to use SP queuing.

```
[Sysname-qmprofile-qm1] queue 7 sp
```

Assign queue 0 through queue 6 to WRR group 1, with their packet-count weights as 2, 1, 2, 4, 6, 8, and 10, respectively.

```
[Sysname-qmprofile-qm1] queue 0 wrr group 1 weight 2
```

```
[Sysname-qmprofile-qm1] queue 1 wrr group 1 weight 1
```

```
[Sysname-qmprofile-qm1] queue 2 wrr group 1 weight 2
```

```
[Sysname-qmprofile-qm1] queue 3 wrr group 1 weight 4
```

```
[Sysname-qmprofile-qm1] queue 4 wrr group 1 weight 6
```

```
[Sysname-qmprofile-qm1] queue 5 wrr group 1 weight 8
```

```
[Sysname-qmprofile-qm1] queue 6 wrr group 1 weight 10
```

```
[Sysname-qmprofile-qm1] quit
```

Apply queue scheduling profile **qm1** to GigabitEthernet 1/0/1.

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

```
[Sysname-GigabitEthernet1/0/1] qos apply qmprofile qm1
```

After the configuration is completed, GigabitEthernet 1/0/1 performs queue scheduling as specified in queue scheduling profile **qm1**.

Display and maintenance commands for congestion management

Execute **display** commands in any view.

Task	Command
Display the configuration of queue scheduling profiles.	display qos qmprofile configuration [<i>profile-name</i>] [slot <i>slot-number</i>]
Display the queue scheduling profiles applied to interfaces.	display qos qmprofile interface [<i>interface-type</i> <i>interface-number</i>]
Display outbound queue-based traffic statistics for interfaces.	display qos queue-statistics interface outbound
Display SP queuing configuration.	display qos queue sp interface [<i>interface-type</i> <i>interface-number</i>]
Display WRR queuing configuration.	display qos queue wrr interface [<i>interface-type</i> <i>interface-number</i>]

Configuring traffic filtering

About traffic filtering

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

Restrictions and guidelines: Traffic filtering configuration

The device supports the following application destinations for traffic filtering:

- Interface.
- VLANs.
- Globally.
- User profile.

Procedure

1. Enter system view.
system-view
2. Define a traffic class.
 - a. Create a traffic class and enter traffic class view.
traffic classifier *classifier-name* [**operator** { **and** | **or** }]
 - b. Configure a match criterion.
if-match *match-criteria*
By default, no match criterion is configured.
For more information about configuring match criteria, see *ACL and QoS Command Reference*.
 - c. Return to system view.
quit
3. Define a traffic behavior.
 - a. Create a traffic behavior and enter traffic behavior view.
traffic behavior *behavior-name*
 - b. Configure the traffic filtering action.
filter { **deny** | **permit** }
By default, no traffic filtering action is configured.
If a traffic behavior has the **filter deny** action, all other actions in the traffic behavior except class-based accounting do not take effect.
 - c. Return to system view.
quit
4. Define a QoS policy.
 - a. Create a QoS policy and enter QoS policy view.

```
qos policy policy-name
```

- b. Associate the traffic class with the traffic behavior in the QoS policy.

```
classifier classifier-name behavior behavior-name
```

By default, a traffic class is not associated with a traffic behavior.

- c. Return to system view.

```
quit
```

5. Apply the QoS policy.

For more information, see "[Applying the QoS policy.](#)"

By default, no QoS policy is applied.

6. (Optional.) Display the traffic filtering configuration.

```
display traffic behavior user-defined [ behavior-name ]
```

This command is available in any view.

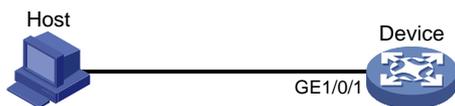
Traffic filtering configuration examples

Example: Configuring traffic filtering

Network configuration

As shown in [Figure 15](#), configure traffic filtering on GigabitEthernet 1/0/1 to deny the incoming packets with a source port number other than 21.

Figure 15 Network diagram



Procedure

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

```
<Device> system-view
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule 0 permit tcp source-port neq 21
[Device-acl-ipv4-adv-3000] quit
```

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

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

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

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

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

```
[Device] qos policy policy
```

```
[Device-qospolicy-policy] classifier classifier_1 behavior behavior_1
[Device-qospolicy-policy] quit
# Apply QoS policy policy to the incoming traffic of GigabitEthernet 1/0/1.
[Device] interface gigabitethernet 1/0/1
[Device-GigabitEthernet1/0/1] qos apply policy policy inbound
```

Configuring priority marking

About priority marking

Priority marking sets the priority fields or flag bits of packets to modify the priority of packets. For example, you can use priority marking to set IP precedence or DSCP for a class of IP packets to control the forwarding of these packets.

To configure priority marking to set the priority fields or flag bits for a class of packets, perform the following tasks:

1. Configure a traffic behavior with a priority marking action.
2. Associate the traffic class with the traffic behavior.

Priority marking can be used together with priority mapping. For more information, see "[Configuring priority mapping](#)."

Configuring priority marking

Restrictions and guidelines

The device supports the following application destinations for priority marking:

- Interface.
- VLANs.
- Globally.
- User profile.

Procedure

1. Enter system view.
system-view
2. Define a traffic class.
 - a. Create a traffic class and enter traffic class view.
traffic classifier *classifier-name* [**operator** { **and** | **or** }]
 - b. Configure a match criterion.
if-match *match-criteria*
By default, no match criterion is configured.
For more information about the **if-match** command, see *ACL and QoS Command Reference*.
 - c. Return to system view.
quit
3. Define a traffic behavior.
 - a. Create a traffic behavior and enter traffic behavior view.
traffic behavior *behavior-name*
 - b. Configure a priority marking action.
For configurable priority marking actions, see the **remark** commands in *ACL and QoS Command Reference*.
 - c. Return to system view.
quit

4. Define a QoS policy.
 - a. Create a QoS policy and enter QoS policy view.
`qos policy policy-name`
 - b. Associate the traffic class with the traffic behavior in the QoS policy.
`classifier classifier-name behavior behavior-name`
By default, a traffic class is not associated with a traffic behavior.
 - c. Return to system view.
`quit`
5. Apply the QoS policy.
For more information, see "[Applying the QoS policy.](#)"
By default, no QoS policy is applied.
6. (Optional.) Display the priority marking configuration.
`display traffic behavior user-defined [behavior-name]`
This command is available in any view.

Priority marking configuration examples

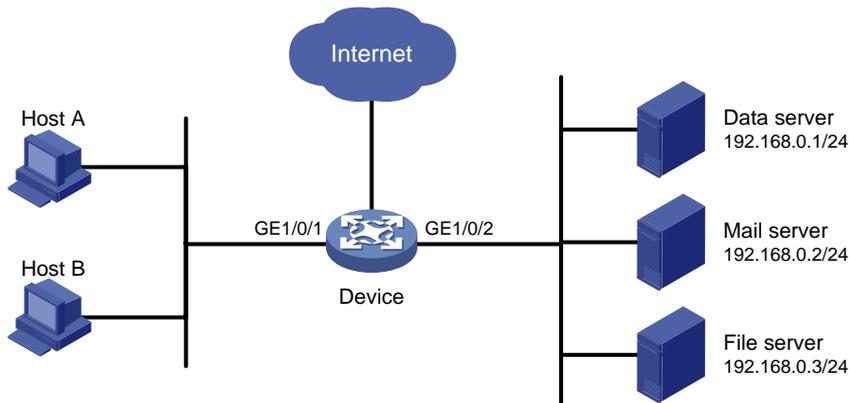
Example: Configuring priority marking

Network configuration

As shown in [Figure 16](#), configure priority marking on the device 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 16 Network diagram



Procedure

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

```
<Device> system-view
```

```

[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule permit ip destination 192.168.0.1 0
[Device-acl-ipv4-adv-3000] quit

# Create advanced ACL 3001, and configure a rule to match packets with destination IP address
192.168.0.2.
[Device] acl advanced 3001
[Device-acl-ipv4-adv-3001] rule permit ip destination 192.168.0.2 0
[Device-acl-ipv4-adv-3001] quit

# Create advanced ACL 3002, and configure a rule to match packets with destination IP address
192.168.0.3.
[Device] acl advanced 3002
[Device-acl-ipv4-adv-3002] rule permit ip destination 192.168.0.3 0
[Device-acl-ipv4-adv-3002] quit

# Create a traffic class named classifier_dbserver, and use ACL 3000 as the match criterion in the
traffic class.
[Device] traffic classifier classifier_dbserver
[Device-classifier-classifier_dbserver] if-match acl 3000
[Device-classifier-classifier_dbserver] quit

# Create a traffic class named classifier_mserver, and use ACL 3001 as the match criterion in the
traffic class.
[Device] traffic classifier classifier_mserver
[Device-classifier-classifier_mserver] if-match acl 3001
[Device-classifier-classifier_mserver] quit

# Create a traffic class named classifier_fserver, and use ACL 3002 as the match criterion in the
traffic class.
[Device] traffic classifier classifier_fserver
[Device-classifier-classifier_fserver] if-match acl 3002
[Device-classifier-classifier_fserver] quit

# Create a traffic behavior named behavior_dbserver, and configure the action of setting the local
precedence value to 4.
[Device] traffic behavior behavior_dbserver
[Device-behavior-behavior_dbserver] remark local-precedence 4
[Device-behavior-behavior_dbserver] quit

# Create a traffic behavior named behavior_mserver, and configure the action of setting the local
precedence value to 3.
[Device] traffic behavior behavior_mserver
[Device-behavior-behavior_mserver] remark local-precedence 3
[Device-behavior-behavior_mserver] quit

# Create a traffic behavior named behavior_fserver, and configure the action of setting the local
precedence value to 2.
[Device] traffic behavior behavior_fserver
[Device-behavior-behavior_fserver] remark local-precedence 2
[Device-behavior-behavior_fserver] quit

# Create a QoS policy named policy_server, and associate traffic classes with traffic behaviors in
the QoS policy.
[Device] qos policy policy_server
[Device-qospolicy-policy_server] classifier classifier_dbserver behavior
behavior_dbserver

```

```
[Device-qospolicy-policy_server] classifier classifier_mserver behavior
behavior_mserver
[Device-qospolicy-policy_server] classifier classifier_fserver behavior
behavior_fserver
[Device-qospolicy-policy_server] quit
# Apply QoS policy policy_server to the incoming traffic of GigabitEthernet 1/0/1.
[Device] interface gigabitethernet 1/0/1
[Device-GigabitEthernet1/0/1] qos apply policy policy_server inbound
[Device-GigabitEthernet1/0/1] quit
```

Configuring nesting

About nesting

Nesting adds a VLAN tag to the matching packets to allow the VLAN-tagged packets to pass through the corresponding VLAN. For example, you can add an outer VLAN tag to packets from a customer network to a service provider network. This allows the packets to pass through the service provider network by carrying a VLAN tag assigned by the service provider.

Restrictions and guidelines: Nesting configuration

The device supports the following application destinations in the inbound direction for nesting:

- Interface.
- VLANs.
- Globally.
- User profile.

If QinQ has been enabled on the interface and the nesting action is associated with a VLAN ID match criterion, the nesting action might not take effect. For the nesting action to take effect, the interface must allow the matching packets to pass tagged.

Procedure

1. Enter system view.
system-view
2. Define a traffic class.
 - a. Create a traffic class and enter traffic class view.
traffic classifier *classifier-name* [**operator** { **and** | **or** }]
 - b. Configure a match criterion.
if-match *match-criteria*
By default, no match criterion is configured for a traffic class.
For more information about the match criteria, see the **if-match** command in *ACL and QoS Command Reference*.
 - c. Return to system view.
quit
3. Define a traffic behavior.
 - a. Create a traffic behavior and enter traffic behavior view.
traffic behavior *behavior-name*
 - b. Configure an outer VLAN tag adding action.
nest top-most vlan *vlan-id*
By default, no outer VLAN tag adding action is configured for a traffic behavior.
 - c. Return to system view.
quit
4. Define a QoS policy.

- a. Create a QoS policy and enter QoS policy view.
`qos policy policy-name`
 - b. Associate the traffic class with the traffic behavior in the QoS policy.
`classifier classifier-name behavior behavior-name`
By default, a traffic class is not associated with a traffic behavior.
 - c. Return to system view.
`quit`
5. Apply the QoS policy.
For more information, see "[Applying the QoS policy.](#)"
By default, no QoS policy is applied.
 6. (Optional.) Display the nesting configuration.
`display traffic behavior user-defined [behavior-name]`
This command is available in any view.

Nesting configuration examples

Example: Configuring nesting

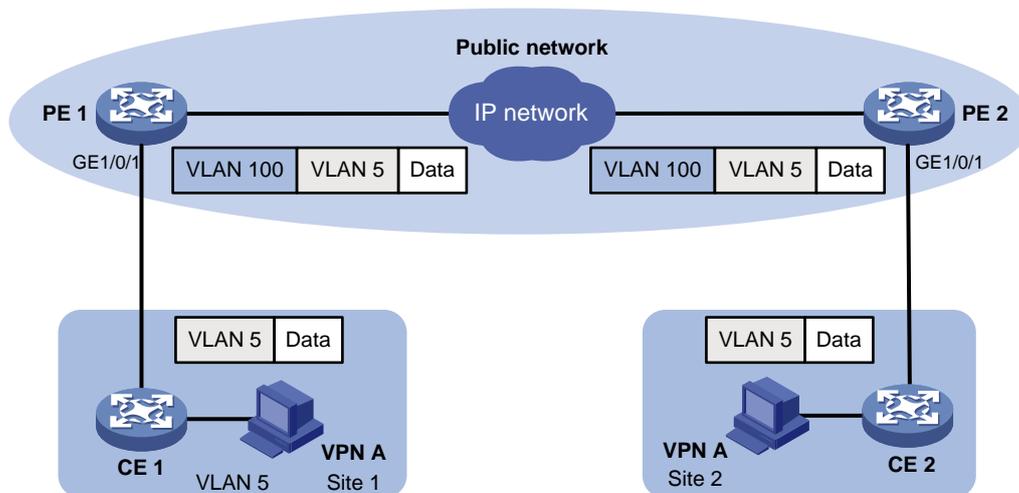
Network configuration

As shown in [Figure 17](#):

- Site 1 and Site 2 in VPN A are two branches of a company. They use VLAN 5 to transmit traffic.
- Because Site 1 and Site 2 are located in different areas, the two sites use the VPN access service of a service provider. The service provider assigns VLAN 100 to the two sites.

Configure nesting, so that the two branches can communicate through the service provider network.

Figure 17 Network diagram



Procedure

1. Configuring PE 1:
Create a traffic class named **test** to match traffic with VLAN ID 5.

```
<PE1> system-view
[PE1] traffic classifier test
```

```
[PE1-classifier-test] if-match service-vlan-id 5
[PE1-classifier-test] quit
```

Configure an action to add outer VLAN tag 100 in traffic behavior **test**.

```
[PE1] traffic behavior test
[PE1-behavior-test] nest top-most vlan 100
[PE1-behavior-test] quit
```

Create a QoS policy named **test**, and associate class **test** with behavior **test** in the QoS policy.

```
[PE1] qos policy test
[PE1-qospolicy-test] classifier test behavior test
[PE1-qospolicy-test] quit
```

Configure the downlink port (GigabitEthernet 1/0/1) as a hybrid port, and assign the port to VLAN 100 as an untagged member.

```
[PE1] interface gigabitethernet 1/0/1
[PE1-GigabitEthernet1/0/1] port link-type hybrid
[PE1-GigabitEthernet1/0/1] port hybrid vlan 100 untagged
```

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

```
[PE1-GigabitEthernet1/0/1] qos apply policy test inbound
[PE1-GigabitEthernet1/0/1] quit
```

Configure the uplink port (GigabitEthernet 1/0/2) as a trunk port, and assign it to VLAN 100.

```
[PE1] interface gigabitethernet 1/0/2
[PE1-GigabitEthernet1/0/2] port link-type trunk
[PE1-GigabitEthernet1/0/2] port trunk permit vlan 100
[PE1-GigabitEthernet1/0/2] quit
```

2. Configuring PE 2:

Configure PE 2 in the same way PE 1 is configured.

Configuring traffic redirecting

About traffic redirecting

Traffic redirecting redirects packets matching the specified match criteria to a location for processing.

You can redirect packets to the following destinations:

- CPU.
- Interface.

Restrictions and guidelines: Traffic redirecting configuration

- The device supports the following application destinations in the inbound direction for traffic redirecting:
 - Interface.
 - VLANs.
 - Globally.
 - User profile.
- If you execute the **redirect** command multiple times, the most recent configuration takes effect.

Procedure

1. Enter system view.
system-view
2. Define a traffic class.
 - a. Create a traffic class and enter traffic class view.
traffic classifier *classifier-name* [**operator** { **and** | **or** }]
 - b. Configure a match criterion.
if-match *match-criteria*
By default, no match criterion is configured for a traffic class.
For more information about the match criteria, see the **if-match** command in *ACL and QoS Command Reference*.
 - c. Return to system view.
quit
3. Define a traffic behavior.
 - a. Create a traffic behavior and enter traffic behavior view.
traffic behavior *behavior-name*
 - b. Configure a traffic redirecting action.
redirect { **cpu** | **interface** *interface-type interface-number* }
By default, no traffic redirecting action is configured for a traffic behavior.

- c. Return to system view.
`quit`
4. Define a QoS policy.
 - a. Create a QoS policy and enter QoS policy view.
`qos policy policy-name`
 - b. Associate the traffic class with the traffic behavior in the QoS policy.
`classifier classifier-name behavior behavior-name`
By default, a traffic class is not associated with a traffic behavior.
 - c. Return to system view.
`quit`
5. Apply the QoS policy.
For more information, see "[Applying the QoS policy.](#)"
By default, no QoS policy is applied.
6. (Optional.) Display traffic redirecting configuration.
`display traffic behavior user-defined [behavior-name]`
This command is available in any view.

Traffic redirecting configuration examples

Example: Configuring traffic redirecting

Network configuration

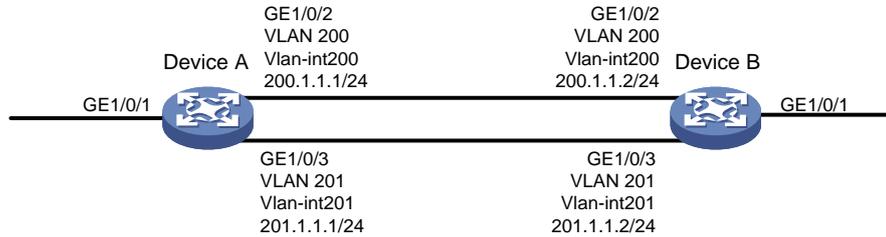
As shown in [Figure 18](#):

- Device A is connected to Device B through two links. Device A and Device B are each connected to other devices.
- GigabitEthernet 1/0/1 of Device A is a trunk port and belongs to VLAN 200 and VLAN 201.
- GigabitEthernet 1/0/2 of Device A and GigabitEthernet 1/0/2 of Device B belong to VLAN 200.
- GigabitEthernet 1/0/3 of Device A and GigabitEthernet 1/0/3 of Device B belong to VLAN 201.
- On Device A, the IP address of VLAN-interface 200 is 200.1.1.1/24, and that of VLAN-interface 201 is 201.1.1.1/24.
- On Device B, the IP address of VLAN-interface 200 is 200.1.1.2/24, and that of VLAN-interface 201 is 201.1.1.2/24.

Configure the actions of redirecting traffic to an interface to meet the following requirements:

- Packets with source IP address 2.1.1.1 received on GigabitEthernet 1/0/1 of Device A are forwarded to GigabitEthernet 1/0/2.
- Packets with source IP address 2.1.1.2 received on GigabitEthernet 1/0/1 of Device A are forwarded to GigabitEthernet 1/0/3.
- Other packets received on GigabitEthernet 1/0/1 of Device A are forwarded according to the routing table.

Figure 18 Network diagram



Procedure

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

```
<DeviceA> system-view
[DeviceA] acl basic 2000
[DeviceA-acl-ipv4-basic-2000] rule permit source 2.1.1.1 0
[DeviceA-acl-ipv4-basic-2000] quit
```

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

```
[DeviceA] acl basic 2001
[DeviceA-acl-ipv4-basic-2001] rule permit source 2.1.1.2 0
[DeviceA-acl-ipv4-basic-2001] quit
```

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

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

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

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

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

```
[DeviceA] traffic behavior behavior_1
[DeviceA-behavior-behavior_1] redirect interface gigabitethernet 1/0/2
[DeviceA-behavior-behavior_1] quit
```

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

```
[DeviceA] traffic behavior behavior_2
[DeviceA-behavior-behavior_2] redirect interface gigabitethernet 1/0/3
[DeviceA-behavior-behavior_2] quit
```

Create a QoS policy named **policy**.

```
[DeviceA] qos policy policy
```

Associate traffic class **classifier_1** with traffic behavior **behavior_1** in the QoS policy.

```
[DeviceA-qospolicy-policy] classifier classifier_1 behavior behavior_1
```

Associate traffic class **classifier_2** with traffic behavior **behavior_2** in the QoS policy.

```
[DeviceA-qospolicy-policy] classifier classifier_2 behavior behavior_2
[DeviceA-qospolicy-policy] quit
```

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

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

Configuring global CAR

About global CAR

Global committed access rate (CAR) is an approach to policing traffic flows globally. It adds flexibility to common CAR where traffic policing is performed only on a per-traffic class or per-interface basis. In this approach, CAR actions are created in system view and each can be used to police multiple traffic flows as a whole.

Global CAR provides the following CAR actions: aggregate CAR and hierarchical CAR.

Aggregate CAR

An aggregate CAR action is created globally. It can be directly applied to interfaces or used in the traffic behaviors associated with different traffic classes to police multiple traffic flows as a whole. The total rate of the traffic flows must conform to the traffic policing specifications set in the aggregate CAR action.

Hierarchical CAR

A hierarchical CAR action is created globally. It must be used in conjunction with a common CAR or aggregate CAR action. With a hierarchical CAR action, you can limit the total traffic of multiple traffic classes.

A hierarchical CAR action can be used in the common or aggregate CAR action for a traffic class in either AND mode or OR mode.

- In AND mode, the rate of the traffic class is strictly limited under the common or aggregate CAR. This mode applies to flows that must be strictly rate limited.
- In OR mode, the traffic class can use idle bandwidth of other traffic classes associated with the hierarchical CAR. This mode applies to high priority, bursty traffic like video.

By using the two modes appropriately, you can improve bandwidth efficiency.

For example, suppose two flows exist: a low priority data flow and a high priority, bursty video flow. Their total traffic rate cannot exceed 4096 kbps and the video flow must be assured of at least 2048 kbps bandwidth. You can perform the following tasks:

- Configure common CAR actions to set the traffic rate to 2048 kbps for the two flows.
- Configure a hierarchical CAR action to limit their total traffic rate to 4096 kbps.
- Use the action in AND mode in the common CAR action for the data flow.
- Use the action in OR mode in the common CAR action for the video flow.

The video flow is assured of 2048 kbps bandwidth and can use idle bandwidth of the data flow.

In a bandwidth oversubscription scenario, the uplink port bandwidth is lower than the total downlink port traffic rate. You can use hierarchical CAR to meet the following requirements:

- Limit the total rate of downlink port traffic.
- Allow each downlink port to forward traffic at the maximum rate when the other ports are idle.

For example, you can perform the following tasks:

- Use common CAR actions to limit the rates of Internet access flow 1 and flow 2 to both 128 kbps.
- Use a hierarchical CAR action to limit their total traffic rate to 192 kbps.

- Use the hierarchical CAR action for both flow 1 and flow 2 in AND mode.

When flow 1 is not present, flow 2 is transmitted at the maximum rate, 128 kbps. When both flows are present, the total rate of the two flows cannot exceed 192 kbps. As a result, the traffic rate of flow 2 might drop below 128 kbps.

Restrictions and guidelines: Global CAR configuration

- Only aggregate CAR is supported in the current software version.
- The device supports the following application destinations in the inbound direction for aggregate CAR:
 - Interface.
 - VLANs.
 - Globally.
 - User profile.

Configuring aggregate CAR

1. Enter system view.
system-view
2. Define a traffic class.
 - a. Create a traffic class and enter traffic class view.
traffic classifier *classifier-name* [**operator** { **and** | **or** }]
 - b. Configure a match criterion.
if-match *match-criteria*
By default, no match criterion is configured.
For configurable match criteria, see the **if-match** command in *ACL and QoS Command Reference*.
 - c. Return to system view.
quit
3. Configure an aggregate CAR action.
qos car *car-name* **aggregative cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green action** | **red action** | **yellow action**] *
qos car *car-name* **aggregative cir** *committed-information-rate* [**cbs** *committed-burst-size*] **pir** *peak-information-rate* [**ebs** *excess-burst-size*] [**green action** | **red action** | **yellow action**] *
By default, no aggregate CAR action is configured.
4. Define a traffic behavior.
 - a. Enter traffic behavior view.
traffic behavior *behavior-name*
 - b. Use the aggregate CAR in the traffic behavior.
car name *car-name*
By default, no aggregate CAR action is used in a traffic behavior.
5. Apply the QoS policy.

For more information, see "[Applying the QoS policy.](#)"

By default, no QoS policy is applied.

Display and maintenance commands for global CAR

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display statistics for global CAR actions.	display qos car name [<i>car-name</i>]
Clear statistics for global CAR actions.	reset qos car name [<i>car-name</i>]

Configuring class-based accounting

About class-based accounting

Class-based accounting collects statistics 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.

Restrictions and guidelines: Class-based accounting configuration

The device supports the following application destinations for class-based accounting:

- Interface.
- VLANs.
- Globally.
- User profile.

Procedure

1. Enter system view.
system-view
2. Define a traffic class.
 - a. Create a traffic class and enter traffic class view.
traffic classifier *classifier-name* [**operator** { **and** | **or** }]
 - b. Configure a match criterion.
if-match *match-criteria*
By default, no match criterion is configured.
For more information about the **if-match** command, see *ACL and QoS Command Reference*.
 - c. Return to system view.
quit
3. Define a traffic behavior.
 - a. Create a traffic behavior and enter traffic behavior view.
traffic behavior *behavior-name*
 - b. Configure an accounting action.
accounting { **byte** | **packet** }
By default, no traffic accounting action is configured.
 - c. Return to system view.
quit
4. Define a QoS policy.
 - a. Create a QoS policy and enter QoS policy view.
qos policy *policy-name*

- b. Associate the traffic class with the traffic behavior in the QoS policy.
`classifier classifier-name behavior behavior-name`
 By default, a traffic class is not associated with a traffic behavior.
 - c. Return to system view.
`quit`
5. Apply the QoS policy.
 For more information, see "[Applying the QoS policy.](#)"
 By default, no QoS policy is applied.
 6. (Optional.) Display the class-based accounting configuration.
`display traffic behavior user-defined [behavior-name]`

Class-based accounting configuration examples

Example: Configuring class-based accounting

Network configuration

As shown in [Figure 19](#), configure class-based accounting on GigabitEthernet 1/0/1 to collect statistics for incoming traffic from 1.1.1.1/24.

Figure 19 Network diagram



Procedure

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

```
<Device> system-view
[Device] acl basic 2000
[Device-acl-ipv4-basic-2000] rule permit source 1.1.1.1 0
[Device-acl-ipv4-basic-2000] quit
```

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

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

Create a traffic behavior named **behavior_1**, and configure the class-based accounting action.

```
[Device] traffic behavior behavior_1
[Device-behavior-behavior_1] accounting packet
[Device-behavior-behavior_1] quit
```

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

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

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

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

Display traffic statistics to verify the configuration.

```
[Device] 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)
```

Appendixes

Appendix A Acronyms

Table 2 Appendix A Acronyms

Acronym	Full spelling
BE	Best Effort
BQ	Bandwidth Queuing
CAR	Committed Access Rate
CBS	Committed Burst Size
CBQ	Class Based Queuing
CE	Congestion Experienced
CIR	Committed Information Rate
CQ	Custom Queuing
DiffServ	Differentiated Service
DSCP	Differentiated Services Code Point
EBS	Excess Burst Size
ECN	Explicit Congestion Notification
FIFO	First in First out
FQ	Fair Queuing
GTS	Generic Traffic Shaping
IntServ	Integrated Service
ISP	Internet Service Provider
LSP	Label Switched Path
MPLS	Multiprotocol Label Switching
PE	Provider Edge
PIR	Peak Information Rate
PQ	Priority Queuing
QoS	Quality of Service
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RTP	Real-Time Transport Protocol
SP	Strict Priority
ToS	Type of Service
VoIP	Voice over IP
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing

Acronym	Full spelling
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin

Appendix B Default priority maps

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

Table 3 Default dot1p-lp priority map

Input priority value	dot1p-lp map
dot1p	lp
0	2
1	0
2	1
3	3
4	4
5	5
6	6
7	7

Table 4 Default dscp-dot1p priority map

Input priority value	dscp-dot1p map
dscp	dot1p
0 to 7	0
8 to 15	1
16 to 23	2
24 to 31	3
32 to 39	4
40 to 47	5
48 to 55	6
56 to 63	7

Table 5 Default port priority-local priority map

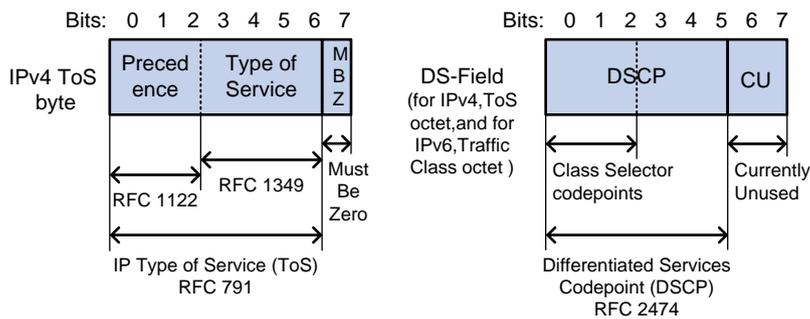
Port priority	Local precedence
0	0
1	1
2	2
3	3

Port priority	Local precedence
4	4
5	5
6	6
7	7

Appendix C Introduction to packet precedence

IP precedence and DSCP values

Figure 20 ToS and DS fields



As shown in [Figure 20](#), the ToS field in the IP header contains 8 bits. The first 3 bits (0 to 2) represent IP precedence from 0 to 7. According to RFC 2474, the ToS field is redefined as the differentiated services (DS) field. A DSCP value is represented by the first 6 bits (0 to 5) of the DS field and is in the range 0 to 63. The remaining 2 bits (6 and 7) are reserved.

Table 6 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 7 DSCP values

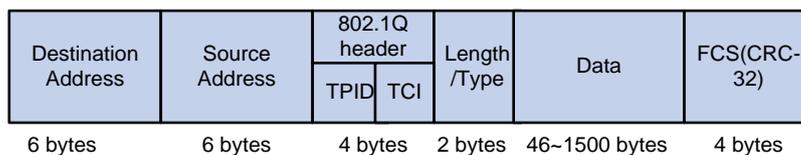
DSCP value (decimal)	DSCP value (binary)	Description
46	101110	ef
10	001010	af11
12	001100	af12

DSCP value (decimal)	DSCP value (binary)	Description
14	001110	af13
18	010010	af21
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. It applies to occasions where Layer 3 header analysis is not needed and QoS must be assured at Layer 2.

Figure 21 An Ethernet frame with an 802.1Q tag header



As shown in [Figure 21](#), the 4-byte 802.1Q tag header contains the 2-byte tag protocol identifier (TPID) and the 2-byte tag control information (TCI). The value of the TPID is 0x8100. [Figure 22](#) shows the format of the 802.1Q tag header. The Priority field in the 802.1Q tag header is called 802.1p priority, because its use is defined in IEEE 802.1p. [Table 8](#) shows the values for 802.1p priority.

Figure 22 802.1Q tag header

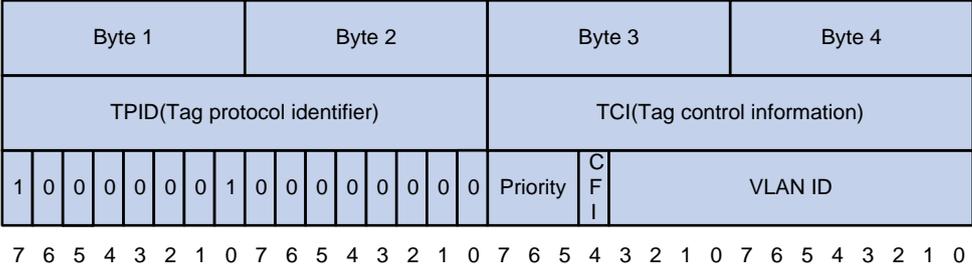


Table 8 Description on 802.1p priority

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

Contents

Configuring data buffers	1
About data buffers.....	1
Data buffer types.....	1
Cell resources and packet resources.....	1
Fixed area and shared area	1
Restrictions and guidelines: Data buffer configuration.....	2
Data buffer tasks at a glance	2
Enabling the Burst feature.....	2
Configuring data buffers manually	3
Display and maintenance commands for data buffers.....	4

Configuring data buffers

About data buffers

Data buffer types

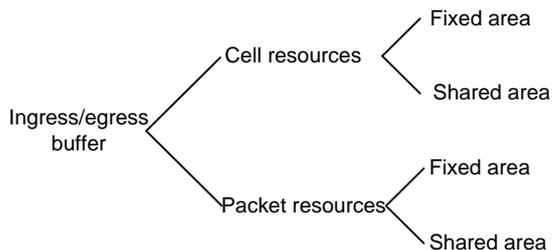
Data buffers temporarily store packets to avoid packet loss.

The following data buffers are available:

- **Ingress buffer**—Stores incoming packets when the CPU is busy.
- **Egress buffer**—Stores outgoing packets when network congestion occurs.

Figure 1 shows the structure of ingress and egress buffers.

Figure 1 Data buffer structure



Cell resources and packet resources

A buffer uses the following types of resources:

- **Cell resources**—Store packets. The buffer uses cell resources based on packet sizes. Suppose a cell resource provides 208 bytes. The buffer allocates one cell resource to a 128-byte packet and two cell resources to a 300-byte packet.
- **Packet resources**—Store packet pointers. A packet pointer indicates where the packet is located in cell resources. The buffer uses one packet resource for each incoming or outgoing packet.

Fixed area and shared area

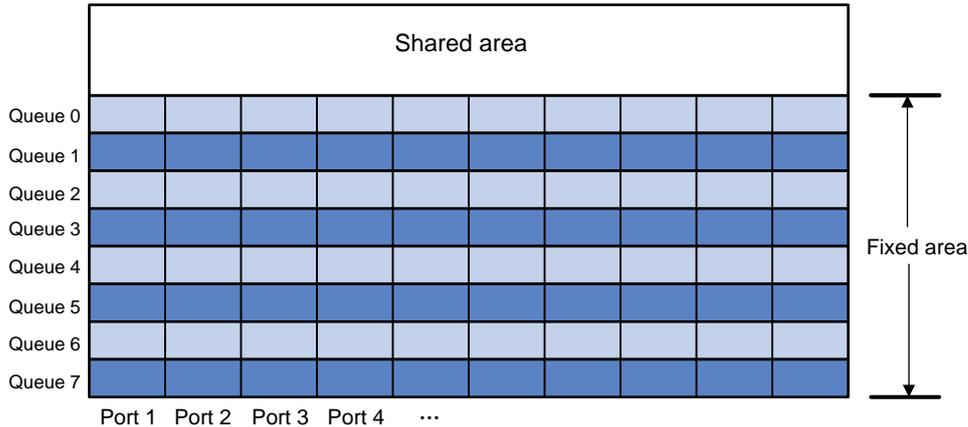
Each type of resources has a fixed area and a shared area.

- **Fixed area**—Partitioned into queues, each of which is equally divided by all the interfaces on the switch, as shown in Figure 2. When congestion occurs or the CPU is busy, the following rules apply:
 - a. An interface first uses the relevant queues of the fixed area to store packets.
 - b. When a queue is full, the interface uses the corresponding queue of the shared area.
 - c. When the queue in the shared area is also full, the interface discards subsequent packets.The system allocates the fixed area among queues as specified by the user. Even if a queue is not full, other queues cannot preempt its space. Similarly, the share of a queue for an interface cannot be preempted by other interfaces even if it is not full.
- **Shared area**—Partitioned into queues, each of which is not equally divided by the interfaces, as shown in Figure 2. The system determines the actual shared-area space for each queue

according to user configuration and the number of packets actually received and sent. If a queue is not full, other queues can preempt its space.

The system puts packets received or sent on all interfaces into a queue in the order they arrive. When the queue is full, subsequent packets are dropped.

Figure 2 Fixed area and shared area



Restrictions and guidelines: Data buffer configuration

You can configure data buffers either manually or automatically by enabling the Burst feature. If you have configured data buffers in one way, delete the configuration before using the other way. Otherwise, the new configuration does not take effect.

Inappropriate data buffer changes can cause system problems. Before manually changing data buffer settings, make sure you understand its impact on your device. As a best practice, use the `burst-mode enable` command if the system requires large buffer spaces.

Data buffer tasks at a glance

To configure the data buffer, perform the following tasks:

- [Enabling the Burst feature](#)
- [Configuring data buffers manually](#)

Enabling the Burst feature

About the Burst feature

The Burst feature enables the device to automatically allocate cell and packet resources. It is well suited to the following scenarios:

- Broadcast or multicast traffic is intensive, resulting in bursts of traffic.
- Traffic comes in and goes out in one of the following ways:
 - Enters a device from a high-speed interface and goes out of a low-speed interface.
 - Enters from multiple same-rate interfaces at the same time and goes out of an interface with the same rate.

The default data buffer settings are changed after the Burst feature is enabled. You can display the data buffer settings by using the **display buffer** command.

Procedure

1. Enter system view.
system-view
2. Enable the Burst feature.
burst-mode enable

By default, the Burst feature is disabled.

Configuring data buffers manually

About manual data buffer configuration

Each type of resources of a buffer, packet or cell, has a fixed size. After you set the shared-area size for a type of resources, the rest is automatically assigned to the fixed area.

By default, all queues have an equal share of the shared area and the fixed area. You can change the maximum shared-area space and the fixed-area for a queue. The unconfigured queues use the default settings.

Restrictions and guidelines

In Release 6126P13 and later versions, you can set the following parameters to 100% in a multicast video scenario to relieve the problem of stuck pictures:

- The maximum shared-area ratio of cell resources for a queue.
- The total shared-area ratio of cell resources.
- The maximum shared-area ratio of packet resources for a queue.
- The total shared-area ratio of packet resources.

The preceding settings are mutually exclusive with the Burst function. Disable the Burst function before configuring these settings.

Procedure

1. Enter system view.
system-view
2. Configure buffer assignment rules. Choose the options to configure as needed:
 - Set the total shared-area ratio.
buffer egress [slot slot-number] { cell | packet } total-shared ratio ratio
If this command is not configured, you can display the default value by using the **display buffer** command.
 - Set the maximum shared-area ratio for a queue.
buffer egress [slot slot-number] { cell | packet } [queue queue-id] shared ratio ratio
The default setting is 10% for both cell resources and packet resources.
The actual maximum shared-area space for each queue is determined based on your configuration and the number of packets to be received and sent.
 - Set the fixed-area ratio for a queue.
buffer egress [slot slot-number] { cell | packet } queue queue-id guaranteed ratio ratio
The default setting is 12% for both cell resources and packet resources.

The sum of fixed-area ratios configured for all queues cannot exceed the total fixed-area ratio. Otherwise, the configuration fails.

3. Apply buffer assignment rules.

buffer apply

You cannot directly modify the applied configuration. To modify the configuration, you must cancel the application, reconfigure data buffers, and reapply the configuration.

Display and maintenance commands for data buffers

Execute **display** commands in any view.

Task	Command
Display buffer size settings.	display buffer [slot <i>slot-number</i>] [queue [<i>queue-id</i>]]
Display data buffer usage.	display buffer usage [slot <i>slot-number</i>]

Contents

Configuring time ranges.....	1
About time ranges.....	1
Restrictions and guidelines: Time range configuration	1
Procedure.....	1
Display and maintenance commands for time ranges	1
Time range configuration examples.....	2
Example: Configuring a time range.....	2

Configuring time ranges

About time ranges

You can implement a service based on the time of the day by applying a time range to it. A time-based service takes effect only in time periods specified by the time range. For example, you can implement time-based ACL rules by applying a time range to them.

The following basic types of time ranges are available:

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

The active period of a time range 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.

Restrictions and guidelines: Time range configuration

When you configure the ACL hardware mode, follow these restrictions and guidelines:

- If a time range does not exist, the service based on the time range does not take effect.
- You can create a maximum of 1024 time ranges, each with a maximum of 32 periodic statements and 12 absolute statements.

Procedure

1. Enter system view.

```
system-view
```

2. Create or edit 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 }
```

If an existing time range name is provided, this command adds a statement to the time range.

Display and maintenance commands for time ranges

Execute the **display** command in any view.

Task	Command
Display time range configuration and status.	<pre>display time-range { <i>time-range-name</i> all }</pre>

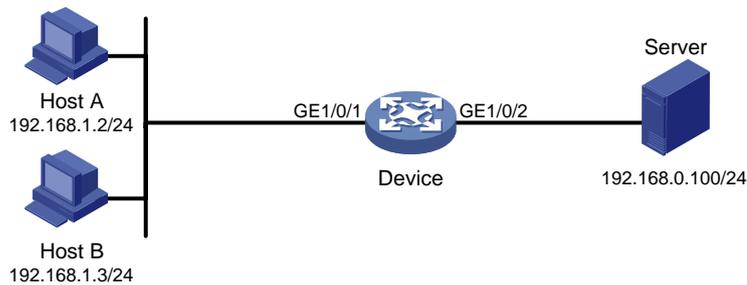
Time range configuration examples

Example: Configuring a time range

Network configuration

As shown in [Figure 1](#), configure an ACL on the device to allow Host A to access the server only during 8:00 and 18:00 on working days from June 2015 to the end of the year.

Figure 1 Network diagram



Procedure

Create a periodic time range during 8:00 and 18:00 on working days from June 2015 to the end of the year.

```
<Device> system-view
```

```
[Device] time-range work 8:0 to 18:0 working-day from 0:0 6/1/2015 to 24:00 12/31/2015
```

Create an IPv4 basic ACL numbered 2001, and configure a rule in the ACL to permit packets only from 192.168.1.2/32 during the time range **work**.

```
[Device] acl basic 2001
```

```
[Device-acl-ipv4-basic-2001] rule permit source 192.168.1.2 0 time-range work
```

```
[Device-acl-ipv4-basic-2001] rule deny source any time-range work
```

```
[Device-acl-ipv4-basic-2001] quit
```

Apply IPv4 basic ACL 2001 to filter outgoing packets on GigabitEthernet 1/0/2.

```
[Device] interface gigabitethernet 1/0/2
```

```
[Device-GigabitEthernet1/0/2] packet-filter 2001 outbound
```

```
[Device-GigabitEthernet1/0/2] quit
```

Verifying the configuration

Verify that the time range **work** is active on the device.

```
[Device] display time-range all
```

```
Current time is 13:58:35 6/19/2015 Friday
```

```
Time-range : work (Active)
```

```
08:00 to 18:00 working-day
```

```
from 00:00 6/1/2015 to 00:00 1/1/2016
```