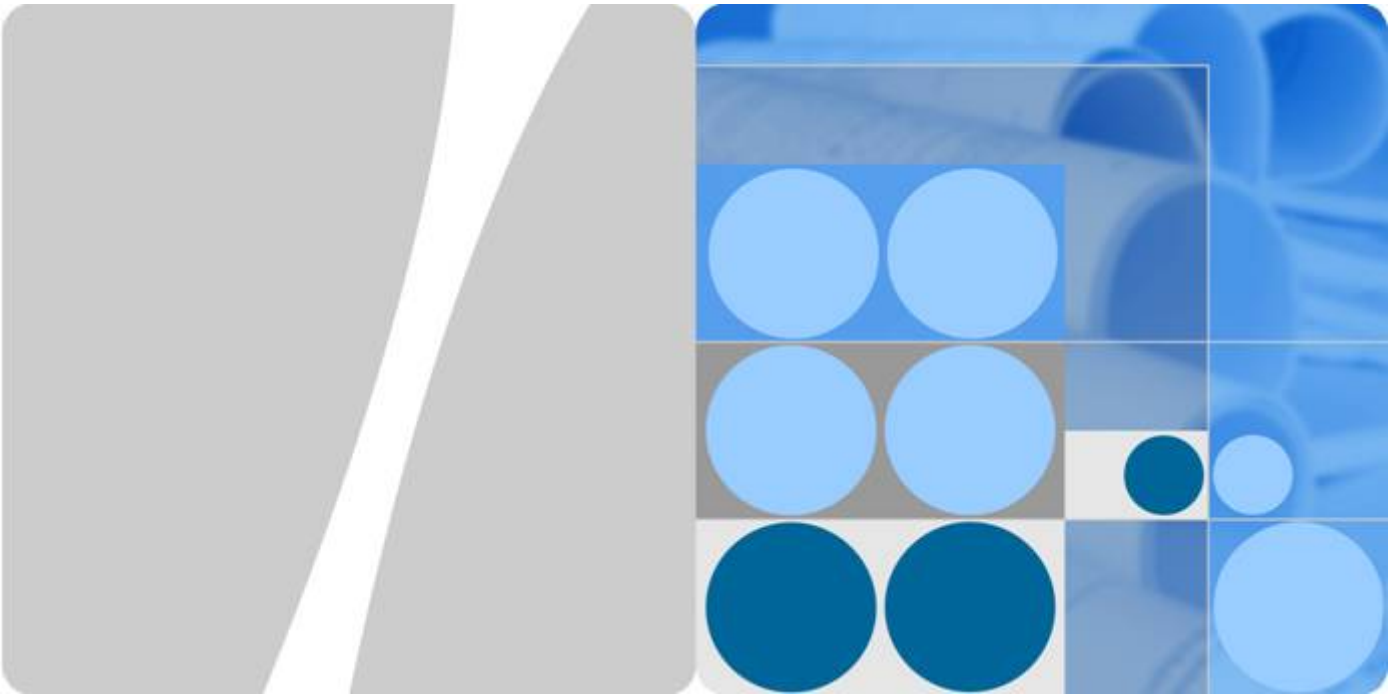


S7700 Smart Routing Switch

V200R001C00

Configuration Guide - QoS

Issue 04

Date 2013-04-10

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://enterprise.huawei.com

About This Document

Intended Audience

This document provides the basic concepts, configuration procedures, and configuration examples in different application scenarios of the QoS supported by the S7700.

This document describes how to configure the QoS.

This document is intended for:

- Data configuration engineers
- Commissioning engineers
- Network monitoring engineers
- System maintenance engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
DANGER	Indicates a hazard with a high level of risk, which if not avoided, will result in death or serious injury.
WARNING	Indicates a hazard with a medium or low level of risk, which if not avoided, could result in minor or moderate injury.
CAUTION	Indicates a potentially hazardous situation, which if not avoided, could result in equipment damage, data loss, performance degradation, or unexpected results.
TIP	Indicates a tip that may help you solve a problem or save time.
NOTE	Provides additional information to emphasize or supplement important points of the main text.

Command Conventions

The command conventions that may be found in this document are defined as follows.

Convention	Description
Boldface	The keywords of a command line are in **boldface**.
Italic	Command arguments are in *italics*.
[ ]	Items (keywords or arguments) in brackets [ ] are optional.
{ x | y | ... }	Optional items are grouped in braces and separated by vertical bars. One item is selected.
[x | y | ...]	Optional items are grouped in brackets and separated by vertical bars. One item is selected or no item is selected.
{ x | y | ... }*	Optional items are grouped in braces and separated by vertical bars. A minimum of one item or a maximum of all items can be selected.
[x | y | ...]*	Optional items are grouped in brackets and separated by vertical bars. Several items or no item can be selected.
&<1-n>	The parameter before the & sign can be repeated 1 to n times.
#	A line starting with the # sign is comments.

Change History

Updates between document issues are cumulative. Therefore, the latest document issue contains all updates made in previous issues.

Changes in Issue 04 (2013-04-10)

The fourth commercial release has the following updates:

The following contents are modified:

- **1.4.4 Configuring a Traffic Policy**
- **2.3.3 Configuring a Traffic Policing Action on the**

Changes in Issue 03 (2012-10-20)

The third commercial release has the following updates:

The following contents are modified:

- **3.5.1 Example for Configuring Congestion Avoidance and Congestion Management**
- **3.4.1 Displaying the Queue-based Statistics**
- **2.4.3 (Optional) Setting the Length of the Interface Queue**

- **Creating a Traffic Classifier Based on an ACL**
- **3.3.3 Setting the Scheduling Mode for an Interface Queue**
- **Configuring the Redirection Action**

Changes in Issue 02 (2012-05-23)

The second commercial release has the following updates:

The following contents are modified:

- **1.4.3 Configuring a Traffic Behavior**

Changes in Issue 01 (2012-03-15)

Initial commercial release.

Contents

1 Class-based QoS Configuration

About This Chapter

This chapter describes the basic concepts of class-based quality of service (QoS), including the traffic classifier, traffic behavior, traffic policy, DiffServ domain, and priority mapping. It also describes configuration methods and provides configuration examples of the traffic policy based on complex traffic classification and priority mapping based on simple traffic classification.

1.1 Introduction to Class-based QoS

Class-based QoS is used to classify packets sharing common features into one class and provide the same QoS service for traffic of the same type by matching packets with certain rules. In this manner, differentiated services are provided.

1.2 Class-based QoS Features Supported by the S7700

The S7700 supports simple traffic classification, complex traffic classification, and priority mapping.

Simple Traffic Classification

On the S7700, you can perform simple traffic classification for packets based on the mappings between packet priorities and Per-Hop Behaviors (PHBs) defined in a Differentiated Services (DiffServ) domain. If packets come from an upstream device, the S7700 binds a DiffServ domain to the inbound interface. In the DiffServ domain, the S7700 maps packet priorities to PHBs and colors. On the S7700, congestion management is based on PHBs and congestion avoidance is based on colors. If packets are sent to a downstream device, the S7700 binds a DiffServ domain to the outbound interface. In the DiffServ domain, the S7700 maps PHBs and colors to priorities so that the downstream device provides QoS services based on packet priorities.

Simple traffic classification is based on the following:

- DSCP priority of IP packets
- 802.1p priority of packets in a VLAN

Complex Traffic Classification

Complex traffic classification is performed based on Layer 2 or Layer 3 information in packets or by using access control lists (ACLs). You can bind a traffic classifier to a traffic behavior to process packets matching the traffic classifier.

A traffic behavior is related to the current phase of packets and the current network load. For example, when packets enter a node, the S7700 performs traffic policing and access control based on the committed information rate (CIR). When packets leave a node, the S7700 shapes the traffic and re-marks the priorities.

Complex traffic classification is based on the following:

- 802.1p priority in VLAN packets
- VLAN ID in packets
- 802.1p priority in CVLAN packets
- CVLAN ID in CVLAN packets
- Double tags of VLAN packets
- Inbound or outbound interface
- IP precedence in IP packets
- DSCP priority in IP packets
- EXP priority in MPLS packets

- SYN Flag field in Transmission Control Protocol (TCP) packets
- Source MAC address
- Destination MAC address
- Protocol type field encapsulated in Layer 2 packets
- Layer 3 protocol type
- ACL
- Discarded packets

You can dynamically add, modify, or delete the matching rules of a traffic classifier on the S7700.

Priority Mapping

Different packets carry different precedence fields. For example, VLAN packets carry the 802.1p field, IP packets carry the DSCP field or IP precedence, and MPLS packets carry the EXP field. The mappings between priority fields must be configured on gateways to retain priorities of packets when the packets traverse different networks.

To ensure QoS for different packets, the S7700 maps packet priorities or the default 802.1p priority of an interface to local priorities. The S7700 then determines the queues that packets enter based on the mappings between internal priorities and queues. It then performs traffic shaping, congestion avoidance, and queue scheduling. In addition, the S7700 can re-mark priorities of outgoing packets so that the downstream device can provide differentiated QoS based on packet priorities.

Table 1-1 and **Table 1-2** show the mappings between internal priorities and queues.

Table 1-1 Mappings between internal priorities and queues on the X40SFC

Internal Priority	Queue Index
BE (unknown unicast packets, broadcast packets, and multicast packets)	0
AF1 (unknown unicast packets, broadcast packets, and multicast packets)	1
AF2 (unknown unicast packets, broadcast packets, and multicast packets)	1
AF3 (unknown unicast packets, broadcast packets, and multicast packets)	1
AF4 (unknown unicast packets, broadcast packets, and multicast packets)	2
EF (unknown unicast packets, broadcast packets, and multicast packets)	2
CS6 (unknown unicast packets, broadcast packets, and multicast packets)	6
CS7 (unknown unicast packets, broadcast packets, and multicast packets)	6

Internal Priority	Queue Index
BE (known unicast packets)	0
AF1 (known unicast packets)	1
AF2 (known unicast packets)	2
AF3 (known unicast packets)	3
AF4 (known unicast packets)	4
EF (known unicast packets)	5
CS6 (known unicast packets)	6
CS7 (known unicast packets)	7

Table 1-2 Mappings between internal priorities and queues on other boards

Internal Priority	Queue Index
BE	0
AF1	1
AF2	2
AF3	3
AF4	4
EF	5
CS6	6
CS7	7

NOTE

A color is used to determine whether the packets are discarded, and is independent of the mappings between internal priorities and queues.

You can dynamically modify priority mappings on the S7700.

Traffic Behavior

Complex traffic classification is required to provide differentiated services. Complex traffic classification takes effect only when it is associated with a traffic control action or a resource allocation action.

The S7700 provides the following traffic behaviors based on complex traffic classification:

- Deny/Permit

The permit/deny action is the simplest traffic control action. The S7700 controls network traffic by forwarding or discarding packets.

- Re-marking

  Re-marking refers to the action taken to set the precedence field in a packet. Packets carry different precedence fields on various networks. For example, packets carry the 802.1p field in a VLAN, the EXP field on an MPLS network, and the DSCP field on an IP network. Therefore, the S7700 is required to mark precedence fields of packets based on the network type.

  Generally, a device at the border of a network needs to re-mark the precedence fields of incoming packets. The device at the core of a network provides corresponding QoS services based on precedence fields marked by the border device, or it re-marks the precedence fields based on its configuration rule.

- Redirection

  This traffic control action redirects packets to the CPU, the specified interface, the specified next hop address, or the Label Switching Path (LSP). The S7700 does not forward packets based on the destination IP address. The S7700 can specify a maximum of four next hops.

  By using redirection, you can implement policy-based routing (PBR). The policy-based route is a static route. When the next hop is unavailable, the S7700 forwards packets based on the original forwarding path.

  The S7700 can redirect only incoming packets.

- Traffic policing

  This traffic control action limits the volume of traffic and the resources used by the traffic to monitor the traffic rate. By using traffic policing, the S7700 can discard, and re-mark the colors and CoS of packets whose rate exceeds the rate limit.

  Here, traffic policing based on traffic classification is implemented. For details about traffic policing, see **2 Traffic Policing and Traffic Shaping Configuration**.

- Flow mirroring

  This traffic control action copies the specified data packets to a specified destination to detect and troubleshoot faults on a network.

  For details about flow mirroring, see Mirroring Configuration in the *S7700 Smart Routing Switch Configuration Guide - Device Management*.

- Traffic statistics

  This traffic control action collects data packets matching defined complex traffic classification rules on the S7700.

  You can dynamically add, modify, or delete the actions of a traffic behavior on the S7700.

- Disabling Unicast Reverse Path Forward (URPF)

  This traffic control action is used to disable URPF check for the flows matching traffic classification on the S7700

  For details about URPF, see Configuring URPF in the *S7700 Smart Routing Switch Configuration Guide - Security*.

- Disabling MAC address learning

  After MAC address learning is disabled, the S7700 does not learn source MAC addresses of the packets matching traffic classification rules.

  On a stable network where MAC addresses of packets seldom change, disabling MAC address learning can reduce the size of the MAC address table and improve device

performance. Unauthorized users may change MAC addresses frequently to attack a network. To prevent MAC address overflow and protect the network from such attacks, disable MAC address learning.

The S7700 can dynamically add, modify, and delete actions in traffic behaviors.

Traffic Policy

A traffic policy is a QoS policy configured by binding traffic classifiers to traffic behaviors. You can associate a traffic classifier with a traffic behavior in a traffic policy.

You can dynamically add, modify, or delete the traffic classifier and traffic behavior in a traffic policy.

CAUTION

Dynamically updating the traffic classifiers and behaviors in a traffic policy makes the traffic policy ineffective for a short time. Confirm the operation before you use this command.

1.3 Configuring Priority Mapping Based on Simple Traffic Classification

Priority mapping based on simple traffic classification maps packet priorities to PHBs and colors to provide differentiated services.

1.3.1 Establishing the Configuration Task

Before configuring priority mapping based on simple traffic classification, familiarize yourself with the applicable environment, complete the pre-configuration tasks, and obtain the data required for the configuration. This will help you complete the configuration task quickly and accurately.

Applicable Environment

When packets come from an upstream device, you can classify them according to the precedence fields in the packets, such as the 802.1p priority or DSCP priority. In a DiffServ domain, you need to define the mapping from packet priorities to PHBs and colors to classify packets. After the DiffServ domain is bound to an inbound interface, the QoS mechanism performs congestion management and congestion avoidance according to packet PHBs and colors on an inbound interface.

When packets are sent to a downstream device, you can classify them based on packet PHBs and colors. In a DiffServ domain, define the mapping from packet PHBs and colors to priorities to classify packets. After the DiffServ domain is bound to an outbound interface, a downstream device provides QoS services based on packet priorities.

Pre-configuration Tasks

Before configuring priority mapping based on simple traffic classification, complete the following task:

- Adding an interface that packets pass to a specified VLAN

Data Preparation

To configure priority mapping based on simple traffic classification, you need the following data.

No.	Data
1	Name of a DiffServ domain
2	802.1p priorities of incoming or outgoing packets in a VLAN, DSCP priorities, PHBs, and colors of incoming or outgoing IP packets
3	Type and number of the interface bound to a DiffServ domain

1.3.2 Configuring an Interface to Trust the Priority of Packets

After an interface is configured to trust packet priorities, the S7700 performs PHB mapping according to the specified priority.

Context

The S7700 provides the following priority trust modes:

- Trusting 802.1p priorities

  If packets carry a VLAN tag, the S7700 searches for the mapping table of 802.1p priorities and internal priorities and marks internal priorities for the packets based on the 802.1p priorities of packets. If packets do not carry a VLAN tag, the S7700 uses the default 802.1p priority of an interface and searches for the mapping table of 802.1p priorities and internal priorities based on the default 802.1p priority of an interface.
- Trusting DSCP priorities

  The S7700 searches for the mapping table of DSCP priorities and internal priorities and marks internal priorities for the packets based on DSCP priorities of packets.

To set the same trust priority on multiple interfaces, perform the configuration on the port group.

NOTE

Internal priorities are represented by CoS and colors defined in the DiffServ model.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Or, run:

port-group *port-group-name*

The port group view is displayed.

NOTE

- The interface type can be Ethernet, GE, XGE, or Eth-Trunk.
- Create a port group before performing this task. For details on how to create a port group, see (Optional) Configuring the Interface Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.
- The packet priority trusted by an interface cannot be configured by using a port group on W series boards.

Step 3 Run:

trust { **8021p** { **inner** | **outer** } | **dscp** }

The interface is configured to trust packet priorities.

By default, an interface trusts 802.1p priorities in outer VLAN tags of packets.

NOTE

- On an S-series board:
  - If you run the **trust 8021p inner** command on an inbound interface, the S7700 maps 802.1p priorities in outer VLAN tags to PHBs and colors.
  - If you run the **trust 8021p outer** command on an inbound interface, the S7700 maps 802.1p priorities in outer VLAN tags to PHBs and colors. If a packet does not carry a tag, the packet enters a queue based on the default 802.1p priority of an inbound interface.
- On an E-series board and an F-series board:
  - On an inbound interface, the S7700 maps packet priorities to PHBs and colors.
  - On an outbound interface, if the **trust 8021p inner** command is used, the packet priorities are not mapped to the 802.1p priority in the inner tag if the packets carry double tags. This is because the 802.1p priority in the outer tag mapped to PHBs and colors is added to the 802.1p priority field of the outer tag.
- W board support only the **trust dscp** command and do not trust any packet priority by default.

For more information about boards, see Board Classification in the *S7700 Smart Routing Switch Hardware Description*.

----End

1.3.3 (Optional) Setting the Default 802.1p Priority of an Interface

After the default 802.1p priority of an interface is set, the S7700 performs PHB mapping for the received untagged packets based on the default 802.1p priority if the interface is configured to trust 802.1p priorities of packets.

Context

If an interface receives untagged packets, it needs to add the default VLAN ID and 802.1p priority to the packets before forwarding them.

If an interface is configured to trust 802.1p priorities, the S7700 uses the default 802.1p priority of the interface when the interface receives untagged packets.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
port priority priority-value
```

The default 802.1p priority of the interface is set.

By default, the 802.1p priority of an interface is 0.

NOTE

The default 802.1p priority of an interface cannot be set on X40SFC boards.

----End

1.3.4 Creating a DiffServ Domain and Configuring Priority Mapping

When the S7700 functions as a border node between a DiffServ domain and other networks, configure mappings between internal priorities (PHBs and colors) and external priorities (such as 802.1p priorities, DSCP priorities, and EXP priorities).

Context

A DiffServ domain is composed of a group of interconnected DiffServ nodes that use the same service policy and PHBs.

In the DiffServ domain, the S7700 maps priorities of incoming packets to PHBs and colors. The S7700 performs congestion management for packets based on packet PHBs and congestion avoidance for packets based on packet colors. If packets are sent to a downstream device, the S7700 binds a DiffServ domain to the outbound interface. In the DiffServ domain, the S7700 maps packet PHBs and colors to priorities so that the downstream device provides QoS services based on packet priorities.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
diffserv domain { default | ds-domain-name }
```

A DiffServ domain is created and the DiffServ domain view is displayed.

The default DiffServ domain defines the mappings from priorities of packets to PHBs and colors. You can change the mappings defined in the default DiffServ domain, but cannot delete the default DiffServ domain.

In addition to the default DiffServ domain, a maximum of seven domains can be created on an S7700.

Step 3 Run the following commands as required.

- To map 802.1p priorities of packets in a VLAN to PHBs and colors, run the following command on the inbound interface:

  8021p-inbound *8021p-value* **phb** *service-class* [*color*]

- To map PHBs and colors of packets to 802.1p priorities of VLAN packets, run the following command on the outbound interface:

  8021p-outbound *service-class color* **map** *8021p-value*

- To map DSCP priorities of IP packets to PHBs and colors, run the following command on the inbound interface:

  ip-dscp-inbound *dscp-value* **phb** *service-class* [*color*]

- To map PHBs and colors to DSCP priorities of IP packets, run the following command on the outbound interface:

  ip-dscp-outbound *service-class color* **map** *dscp-value*

- To map EXP priorities of MPLS packets to PHBs and colors, run the following command on the inbound interface:

  mpls-exp-inbound *exp-value* **phb** *service-class* [*color*]

- To map PHBs and colors of MPLS packets to EXP priorities, run the following command on the outbound interface:

  mpls-exp-outbound *service-class color* **map** *exp-value*

For details about the mapping, see the following commands:

- 8021p-inbound defines the mapping from the default 802.1p priorities to PHBs and colors.
- 8021p-outbound defines the mapping from PHBs and colors to 802.1p priorities.
- ip-dscp-inbound defines the mapping from DSCP priorities to PHBs and colors.
- ip-dscp-outbound defines the mapping from PHBs and colors to DSCP priorities.
- mpls-exp-inbound defines the mapping from EXP priorities to PHBs and colors.
- mpls-exp-outbound defines the mapping from PHBs and colors to EXP priorities.

NOTE

By default, the MPLS function is disabled on the S7700. To use the MPLS function of the S7700, purchase the license from Huawei local office.

The G24SA, G24CA and X12SA boards do not support MPLS.

S-series boards do not support priority mapping in the outbound direction on an interface.

For details about S-series boards, see Board Classification in the *S7700 Smart Routing Switch Hardware Description*.

If the mpls-exp-inbound and mpls-exp-outbound commands are executed on the F48T, G48S, G48T, G24TFA, G48TBC, or G48SBC board, a maximum of four DiffServ domains can be configured.

----End

1.3.5 Applying a DiffServ Domain

After the DiffServ domain is applied to a specified interface, the S7700 performs PHB mapping for packets passing through the interface.

Context

To map priorities of packets coming from an upstream device to PHBs and colors based on the mappings defined in a DiffServ domain, bind the DiffServ domain to the inbound interface. The system maps packet priorities to PHBs and colors based on the mappings defined in the DiffServ domain.

To map PHBs to priorities of packets sent to a downstream device based on the mappings defined in a DiffServ domain, bind the DiffServ domain to the outbound interface. The system then maps PHBs and colors to priorities of packets based on the mappings defined in the DiffServ domain.

To bind multiple interfaces to the same DiffServ domain, perform the configuration on the port group to reduce the workload.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Or, run:

```
port-group port-group-name
```

The port group view is displayed.

NOTE

- The interface type can be Ethernet, GE, XGE, or Eth-Trunk.
- Create a port group before performing this task. For details on how to create a port group, see (Optional) Configuring the Interface Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

Step 3 Run:

```
trust upstream { ds-domain-name | default | none }
```

The interface is bound to a DiffServ domain.

If the **trust upstream none** command is run on an interface, the S7700 does not perform priority mapping for incoming and outgoing packets on the interface.

To delete the DiffServ domain that is bound to an interface, you must first run the **undo trust upstream** command to delete the bound DiffServ domain. Then run the **trust upstream** command to reconfigure the DiffServ domain.

For details about the mapping, see the following commands:

- 8021p-inbound defines the mapping from the default 802.1p priorities to PHBs and colors.
- 8021p-outbound defines the mapping from PHBs and colors to 802.1p priorities.
- ip-dscp-inbound defines the mapping from DSCP priorities to PHBs and colors.
- ip-dscp-outbound defines the mapping from PHBs and colors to DSCP priorities.
- mpls-exp-inbound defines the mapping from EXP priorities to PHBs and colors.

- mpls-exp-outbound defines the mapping from PHBs and colors to EXP priorities.

NOTE

WN board do not support the **trust upstream none** command.

Step 4 (Optional) Run:

```
undo qos phb marking enable
```

PHB mapping of outgoing packets is disabled.

By default, PHB mapping is enabled on an outbound interface.

----End

1.3.6 Checking the Configuration

After priority mapping based on simple traffic classification is configured, you can view the mapping between packet priorities and PHBs.

Prerequisites

The configurations of priority mapping based on simple traffic classification are complete.

Procedure

- Run the **display diffserv domain** [**all** | **name** *ds-domain-name*] command to check the configuration of the DiffServ domain.

----End

1.4 Creating a Traffic Policy Based on Complex Traffic Classification

After the traffic policy based on complex traffic classification is configured, the S7700 classifies packets according to the priority of packets and quintuple information. Then the S7700 takes different traffic actions for packets matching classification conditions, such as permit/deny, re-marking, and redirection.

1.4.1 Establishing the Configuration Task

Before configuring the traffic policy based on complex traffic classification, familiarize yourself with the applicable environment, complete the pre-configuration tasks, and obtain the required data. This helps you complete the configuration task quickly and accurately.

Applicable Environment

At the ingress of a network, the S7700 functions as a border node. To limit the incoming traffic on a network, the S7700 can provide differentiated services for various services according to the DSCP field, protocol type, IP address, port number, fragmentation type, and time range of packets. In this case, you need to create a traffic policy based on complex traffic classification.

Generally, complex traffic classification is configured on a border node, and simple traffic classification is configured on a core node.

Pre-configuration Tasks

Before creating a traffic policy based on complex traffic classification, complete the following tasks:

- Configuring the physical parameters of interfaces
- Setting link layer attributes of interfaces
- Configuring routing protocols to ensure the connectivity of the network
- Configuring ACLs if ACLs are used as matching rules for traffic classification

Data Preparation

To create a traffic policy based on complex traffic classification, you need the following data.

No.	Data
1	Name of the traffic classifier and matching rules of the traffic classifier
2	Name of the traffic behavior and related parameters
3	Name of the traffic policy
4	Interface that the traffic policy is applied to and ID of the VLAN

1.4.2 Configuring Complex Traffic Classification

The S7700 can classify traffic according to the ACL, and the Layer 2 information and Layer 3 information in packets.

Creating a Traffic Classifier Based on Layer 2 Information

After traffic classification based on Layer 2 information is configured, the S7700 classifies packets based on the Layer 2 information including the 802.1p priority, VLAN ID, source/destination MAC address, incoming/outgoing interface, and Layer 2 protocol type.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
traffic classifier classifier-name [ operator { and | or } ] [ precedence precedence-value ]
```

A traffic classifier based on Layer 2 information is created and the traffic classifier view is displayed.

The **and** parameter indicates that the relationship between rules in a traffic classifier is "AND". That is, the packets match a traffic classifier only when the packets match all non-ACL rules and an ACL rule in the traffic classifier. The **or** parameter indicates that the relationship between

rules in a traffic classifier is "OR". That is, the packets match a traffic classifier when the packets match a rule in the traffic classifier.

By default, the relationship between rules in a traffic classifier is OR.

CAUTION

A traffic classifier allows a maximum of 40000 traffic classification rules.

Step 3 Run the following commands as required.

- To define matching rules based on the 802.1p priority in the inner VLAN tag of QinQ packets, run:

  if-match cvlan-8021p { *8021p-value* } &<1-8>

- To define matching rules based on the 802.1p priority of packets in a VLAN, run:

  if-match 8021p { *8021p-value* } &<1-8>

- To define matching rules based on the VLAN ID in the inner VLAN tag or the VLAN IDs in inner and outer tags of QinQ packets, run:
- To define matching rules based on the outer VLAN ID or VLAN IDs of inner and outer tags of QinQ packets, run:

  if-match vlan-id *start-vlan-id* [**to** *end-vlan-id*] [**cvlan-id** *cvlan-id*]

- To define matching rules based on discarded packets, run:

  if-match discard

- To define matching rules based on double tags of QinQ packets, run:

  if-match double-tag

- To define matching rules based on the EXP priority of MPLS packets, run:

  if-match mpls-exp *exp-value* &<1-8>

- To define matching rules based on the destination MAC address, run:

  if-match destination-mac *mac-address* [
  [**mac-address-mask**] *mac-address-mask*]

- To define matching rules based on the source MAC address, run:

  if-match source-mac *mac-address* [[**mac-address-mask**] *mac-address-mask*]

- To define matching rules based on the incoming interface, run:

  if-match inbound-interface *interface-type interface-number*

- To define matching rules based on the outgoing interface, run:

  if-match outbound-interface *interface-type interface-number*

- To define matching rules based on the protocol field in the Ethernet frame header, run:

  if-match l2-protocol{ **arp** | **ip** | **mpls** | **rarp** | *protocol-value* }

- To define matching rules based on all the packets, run:

  if-match any

----End

Creating a Traffic Classifier Based on Layer 3 Information

After traffic classification based on Layer 3 information is configured, the S7700 classifies packets according to Layer 3 information in packets.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
traffic classifier classifier-name [ operator { and | or } ] [ precedence precedence-value ]
```

A traffic classifier based on Layer 3 information is created and the traffic classifier view is displayed.

The **and** parameter indicates that the relationship between rules in a traffic classifier is AND. That is, the packets match a traffic classifier only when the packets match all non-ACL rules and an ACL rule in the traffic classifier. The **or** parameter indicates that the relationship between rules in a traffic classifier is OR. That is, the packets match a traffic classifier when the packets match a rule in the traffic classifier.

By default, the relationship between rules in a traffic classifier is OR.

CAUTION

A traffic classifier allows a maximum of 40000 traffic classification rules.

Step 3 Run the following commands as required.

- To define matching rules based on the DSCP priority of IP packets, run:
  ```
  if-match [ ipv6 ] dscp dscp-value &<1-8>
  ```
- To define matching rules based on the IP priority of IP packets, run:
  ```
  if-match ip-precedence ip-precedence-value &<1-8>
  ```

  NOTE

  In a traffic classifier where the relationship between rules is AND, the **if-match dscp** and **if-match ip-precedence** commands cannot be used simultaneously.
- To define matching rules based on the Layer 3 protocol type, run:
  ```
  if-match protocol { ip | ipv6 }
  ```
- To define matching rules based on the SYN Flag field of TCP packets, run:
  ```
  if-match tcp syn-flag { syn-flag-value | ack | fin | psh | rst | syn | urg }
  ```
- To define matching rules based on the next IPv6 packet header type, run:
  ```
  if-match ipv6 next-header header-number first-next-header
  ```

  NOTE

  G24SA, X12SA and G24CA boards do not support the routes whose prefix length ranges from 64 to 128.

----End

Creating a Traffic Classifier Based on an ACL

After traffic classification based on an ACL is configured, the S7700 classifies packets based on the ACL.

Context

The S7700 can use an ACL to classify packets based on the IP quintuple.

The S7700 supports basic ACLs, Layer 2 ACLs, user-defined ACLs and advanced ACLs:

- Basic ACLs are used to classify data packets based on the source IP address, fragmentation flag, and time segment of packets.
- Advanced ACLs are used to classify and define data packets based on the source IP address, destination IP address, source port number, destination port number, fragmentation flag, time segment, and protocol type of packets.
- Layer 2 ACLs are used to classify data packets based on the source MAC address and destination MAC address of packets.
- User-defined ACLs process data packets according to the rules defined by users.

The S7700 supports basic ACL6s and advanced ACL6s for IPv6 packets:

- A basic ACL6 can use the source IP address, fragmentation flag, and effective time range as the elements of rules.
- An advanced ACL6 can use the source IP address and destination IP address of data packets, protocol type supported by IP, features of the protocol such as the source port number and destination port number, ICMPv6 protocol, and ICMPv6 Code as the elements of rules.

Procedure

- Creating a traffic classifier based on a basic ACL
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     acl [**ipv6**] *basic-acl-number* [**match-order** { **auto** | **config** }]

     A basic ACL is created and the ACL view is displayed.

     Or, run:

     acl [**ipv6**] **name** *acl-name* **basic** [**match-order** { **auto** | **config** }]

     A named ACL is created and ACL view is displayed.

     NOTE

     The *basic-acl-number* parameter specifies the number of a basic ACL. The value is an integer that ranges from 2000 to 2999.
  3. (Optional) Run:
     step *step-value*

     The step value between ACL rule IDs is set.
  4. Run:
     rule [*rule-id*] { **deny** | **permit** } [**fragment** | **logging** | **source** { *source-ipv6-address prefix-length* | *source-ipv6-address/prefix-length* | *source-ipv6-address* **postfix** *postfix-length* | **any** } | **time-range** *time-name*] *

     A basic ACL4 rule is created.

     Or, run:

rule [*rule-id*] { **deny** | **permit** } [**fragment** | **logging** | **source** { *source-ipv6-address prefix-length* | *source-ipv6-address/prefix-length* | *source-ipv6-address* **postfix** *postfix-length* | **any** } | **time-range** *time-name*] *

A basic ACL6 rule is created.

5. Run:
   quit

   Return to the system view.
6. Run:
   traffic classifier *classifier-name* [**operator** { **and** | **or** }] [**precedence** *precedence-value*]

   A traffic classifier is created and the traffic classifier view is displayed.

   The **and** parameter indicates that the relationship between rules in a traffic classifier is AND. That is, packets match a traffic classifier only when the packets match all non-ACL rules and an ACL rule in the traffic classifier. The **or** parameter indicates that the relationship between rules in a traffic classifier is OR. That is, packets match a traffic classifier when the packets match a rule in the traffic classifier.

   By default, the relationship between rules in a traffic classifier is OR.

CAUTION

A traffic classifier allows a maximum of 40000 traffic classification rules.

7. Run:
   if-match [**ipv6**] **acl** *basic-acl-number*

   A traffic classifier based on a basic ACL is created.

- Creating a traffic classifier based on an advanced ACL
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     :acl *advance-acl-number* [**match-order** { **auto** | **config** }]

     A advance ACL is created and the ACL view is displayed.

     Or, run:

     acl [**ipv6**] **name** *acl-name* **link** [**match-order** { **auto** | **config** }]

     A named ACL is created and ACL view is displayed.

     NOTE

     The *advance-acl-number* parameter specifies the number of a basic ACL. The value is an integer that ranges from 3000 to 3999.
  3. (Optional) Run:
     step *step-value*

     The step value between ACL rule IDs is set.

4. Run the following commands as required.
   - When the parameter *protocol* is specified as the Internet Control Message Protocol (ICMP), the command format is as follows:
     - **rule** [*rule-id*] { **deny** | **permit** } { *protocol-number* | **icmp** } [**destination** { *destination-address destination-wildcard* | **any** } | **dscp** *dscp* | **fragment** | **icmp-type** { *icmp-name* | *icmp-type icmp-code* } | **precedence** *precedence* | **source** { *source-address source-wildcard* | **any** } | **time-range** *time-name* | **tos** *tos* | **ttl-expired**]*
   - When the parameter *protocol* is specified as the Transmission Control Protocol (TCP), the command format is as follows:
     - **rule** [*rule-id*] { **deny** | **permit** } { *protocol-number* | **tcp** } [**destination** { *destination-address destination-wildcard* | **any** } | **destination-port** { **eq** | **gt** | **lt** | **range** } *port* | **dscp** *dscp* | **fragment** | **precedence** *precedence* | **source** { *source-address source-wildcard* | **any** } | **source-port** { **eq** | **gt** | **lt** | **range** } *port* | **tcp-flag** { **ack** | **fin** | **psh** | **rst** | **syn** | **urg** }* | **time-range** *time-name* | **tos** *tos* | **ttl-expired**]*
   - When the parameter *protocol* is specified as the User Datagram Protocol (UDP), the command format is as follows:
     - **rule** [*rule-id*] { **deny** | **permit** } { *protocol-number* | **udp** } [**destination** { *destination-address destination-wildcard* | **any** } | **destination-port** { **eq** | **gt** | **lt** | **range** } *port* | **dscp** *dscp* | **fragment** | **precedence** *precedence* | **source** { *source-address source-wildcard* | **any** } | **source-port** { **eq** | **gt** | **lt** | **range** } *port* | **time-range** *time-name* | **tos** *tos* | **ttl-expired**]*
   - When the parameter *protocol* is specified as another protocol rather than TCP, UDP, or ICMP, the command format is as follows:
     - **rule** [*rule-id*] { **deny** | **permit** } { *protocol-number* | **gre** | **igmp** | **ip** | **ipinip** | **ospf** } [**destination** { *destination-address destination-wildcard* | **any** } | **dscp** *dscp* | **fragment** | **precedence** *precedence* | **source** { *source-address source-wildcard* | **any** } | **time-range** *time-name* | **tos** *tos* | **ttl-expired**]*
   - When **protocol** is set to TCP, the command format of an advanced ACL6 rule is as follows:

     rule [*rule-id*] { **deny** | **permit** } { **tcp** | *protocol-number* } [**destination** { *destination-ipv6-address prefix-length* | *destination-ipv6-address/prefix-length* | **postfix** *postfix-length* | **any** } | **destination-port** { **eq** | **gt** | **lt** | **range** } *port* | **dscp** *dscp* | **fragment** | **logging** | **precedence** *precedence* | **source** { *source-ipv6-address prefix-length* | *source-ipv6-address/prefix-length* | *source-ipv6-address* **postfix** *postfix-length* | **any** } | **source-port** { **eq** | **gt** | **lt** | **range** } *port* | **time-range***time-name* | **tos** *tos*] *
   - When **protocol** is set to UDP, the command format of an advanced ACL6 rule is as follows:

     rule [*rule-id*] { **deny** | **permit** } { **udp** | *protocol-number* } [**destination** { *destination-ipv6-address prefix-length* | *destination-ipv6-address/prefix-length* | **postfix** *postfix-length* | **any** } | **destination-port** { **eq** | **gt** | **lt** | **range** } *port* | **dscp** *dscp* | **fragment** | **logging** | **precedence** *precedence* | **source** { *source-ipv6-address prefix-length* | *source-ipv6-address/prefix-length* | *source-ipv6-address* **postfix** *postfix-length* | **any** } | **source-port** { **eq** | **gt** | **lt** | **range** } *port* | **time-range** *time-name* | **tos** *tos*] *
   - When **protocol** is set to ICMPv6, the command format of an advanced ACL6 rule is as follows:

rule [*rule-id*] { **deny** | **permit** } { **icmpv6** | *protocol-number* } [**destination** { *destination-ipv6-address prefix-length* | *destination-ipv6-address/prefix-length* | **postfix** *postfix-length* | **any** } | **dscp** *dscp* | **fragment** | **icmp6-type** { *icmp6-type-name* | *icmp6-type icmp6-code* } | **logging** | **precedence** *precedence* | **source** { *source-ipv6-address prefix-length* | *source-ipv6-address/prefix-length* | *source-ipv6-address* **postfix** *postfix-length* | **any** } | **time-range** *time-name* | **tos** *tos*] *

- When **protocol** is set to other protocols, the command format of an advanced ACL6 rule is as follows:

  rule [*rule-id*] { **deny** | **permit** } { *protocol-number* | **gre** | **ipv6** | **ospf** } [**destination** { *destination-ipv6-address prefix-length* | *destination-ipv6-address/prefix-length* | *destination-ipv6-address* **postfix** *postfix-length* | **any** } | **dscp** *dscp* | **fragment** | **logging** | **precedence** *precedence* | **source** { *source-ipv6-address prefix-length* | *source-ipv6-address/prefix-length* | *source-ipv6-address* **postfix** *postfix-length* | **any** } | **time-range** *time-name* | **tos** *tos*] *

5. Run:
   quit

   Return to the system view.
6. Run:
   traffic classifier *classifier-name* [**operator** { **and** | **or** }] [**precedence** *precedence-value*]

   A traffic classifier is created and the traffic classifier view is displayed.

   The **and** parameter indicates that the relationship between rules in a traffic classifier is AND. That is, packets match a traffic classifier only when the packets match all non-ACL rules and an ACL rule in the traffic classifier. The **or** parameter indicates that the relationship between rules in a traffic classifier is OR. That is, packets match a traffic classifier when the packets match a rule in the traffic classifier.

   By default, the relationship between rules in a traffic classifier is OR.
7. Run:
   if-match [**ipv6**] **acl** *advanced-acl-number*

   A traffic classifier based on an advanced ACL is created.

- Creating a traffic classifier based on a Layer 2 ACL
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     acl *l2-acl-number* [**match-order** { **auto** | **config** }]

     A Layer 2 ACL is created and the ACL view is displayed.

     Or, run:

     acl name *acl-name* **user** [**match-order** { **auto** | **config** }]

     A named ACL is created and ACL view is displayed.

     NOTE

     The *l2-defined-acl-number* parameter specifies the number of a user-defined ACL. The value is an integer that ranges from 4000 to 4999.

3. (Optional) Run:
   step *step-value*

   The step value between ACL rule IDs is set.
4. Run:
   rule [*rule-id*] { **deny** | **permit** } [{ **ether-ii** | **802.3** | **snap** } | **l2-protocol** *type-value* [*type-mask*] | **destination-mac** *dest-mac-address* [*dest-mac-mask*] | **source-mac** *source-mac-address* [*source-mac-mask*] | **vlan-id** *vlan-id* [*vlan-id-mask*] | **8021p** *802.1p-value* | **cvlan-id** *cvlan-id* [*cvlan-id-mask*] | **cvlan-8021p** *802.1p-value* | **double-tag**]* [**time-range** *time-range-name*]

   A Layer 2 ACL rule is created.
5. Run:
   quit

   Return to the system view.
6. Run:
   traffic classifier *classifier-name* [**operator** { **and** | **or** }] [**precedence** *precedence-value*]

   A traffic classifier is created and the traffic classifier view is displayed.

   The **and** parameter indicates that the relationship between rules in a traffic classifier is AND. That is, packets match a traffic classifier only when the packets match all non-ACL rules and an ACL rule in the traffic classifier. The **or** parameter indicates that the relationship between rules in a traffic classifier is OR. That is, packets match a traffic classifier when the packets match a rule in the traffic classifier.

   By default, the relationship between rules in a traffic classifier is OR.
7. Run:
   if-match acl *l2-acl-number*

   A traffic classifier based on a Layer 2 ACL is created.

- Creating a traffic classifier based on a user-defined ACL
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     acl [**number**] *user-defined-acl-number* [**match-order** { **auto** | **config** }]

     A user-defined ACL is created and the user-defined ACL view is displayed.

     NOTE

     user-defined-acl-number specified the number of a user-defined ACL. The value is an integer that ranges from 5000 to 5999.
  3. (Optional) Run:
     step *step-value*

     The step value between ACL rule IDs is set.
  4. Run the following command:
     rule [*rule-id*] { **deny** | **permit** } [[**l2-head** | **ipv4-head** | **ipv6-head** | **l4-head**] { *rule-string rule-mask offset* }] [**time-range** *time-range-name*]

The user-defined ACL is configured.

5. Run:
   quit

   Return to the system view.

6. Run:
   traffic classifier *classifier-name* [**operator** { **and** | **or** }] [**precedence** *precedence-value*]

   A traffic classifier is created and the traffic classifier view is displayed.

   and indicates the relationship between rules is AND. That is, packets must match all the non-ACL rules and one of the ACL rules of the traffic classifier. **or** indicates the relationship between rules is OR. That is, packets need to match only one rule of the traffic classifier.

   By default, the relationship between rules in a traffic classifier is OR.

7. Run:
   if-match acl *user-defined-acl-number*

   A traffic classifier based on a user-defined ACL is configured.

   You can use only the **if-match acl** *user-defined-acl-number* command in a traffic classifier where the relationship between rules is AND or configure other matching rules. When the **if-match acl** *user-defined-acl-number* command is used and other matching rules are configured, the **if-match acl** *user-defined-acl-number* command can only be used with the commands of **if-match vlan-id**, **if-match inbound-interface**, and **if-match outbound-interface**.

----End

1.4.3 Configuring a Traffic Behavior

The S7700 supports the actions of permit/deny, re-marking, redirection, traffic policing, flow mirroring, and traffic statistics, which can be configured as required.

Configuring the Deny or Permit Action

By configuring the deny or permit action, the S7700 rejects or permits packets matching traffic classification rules to control the network traffic.

Procedure

Step 1 Run:
system-view

The system view is displayed.

Step 2 Run:
traffic behavior *behavior-name*

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run the following commands as required.

- Run:

permit

The permit action is configured.

- Run:

  deny

  The deny action is configured.

NOTE

- If the **deny** action is configured, the packets matching a traffic classifier are discarded. The packets are still discarded even if other actions except for the traffic statistics action are configured.
- If the **permit** action is configured, the packets matching a traffic classifier are processed in order.

----End

Configuring the Re-marking Action

The re-marking action allows the S7700 to re-mark priorities of packets matching traffic classification rules, such as the 802.1p priority of VLAN packets, EXP priority of MPLS packets, and DSCP value of IP packets.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

traffic behavior *behavior-name*

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run the following commands as required.

- Run:

  remark 8021p [*8021p-value* | **inner-8021p**]

  The 802.1p priority of the packets matching the traffic classification is re-marked.

  NOTE

  If **inner-8021p** is specified, the 802.1p priority in the inner tag of packets is re-marked to the outer tag.

- Run:

  remark cvlan-id *cvlan-id*

  The VLAN ID in the inner VLAN tag of the QinQ packets matching the traffic classification is re-marked.

- Run:

  remark vlan-id *vlan-id*

  The VLAN ID in the outer VLAN tag of the packets in a VLAN matching the traffic classification is re-marked.

- Run:

  remark destination-mac *mac-address*

  The destination MAC address of the packets matching the traffic classification is re-marked.

NOTE

In a traffic behavior, the **remark destination-mac** command cannot be used with the following commands simultaneously:

- **redirect ip-nexthop**
- **redirect ip-multihop**

If the traffic policy applied to the outbound direction of an interface contains a VLAN re-marking action, the outbound VLAN on the interface must work in tagged mode.

- Run:

  remark dscp { *dscp-name* | *dscp-value* }

  The DSCP priority of the packets matching the traffic classification is re-marked.

- Run:

  remark local-precedence { *local-precedence-name* | *local-precedence-value* } [*color*]

  The local priority of the packets matching the traffic classification is re-marked.

  In a traffic behavior, the **remark 8021p** command and the **remark local-precedence** command cannot be used together.

NOTE

The **remark cvlan-id** and **remark vlan-id** commands are used to implement the flow-based VLAN mapping function. For description and configuration of flow-based VLAN mapping, see VLAN Mapping Configuration in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

CAUTION

After the **remark vlan-id**, **remark 8021p**, **remark cvlan-id**, and **nest top-most vlan-id** commands are used, the system modifies VLAN tags of packets according to the configuration. These actions are called VLAN-based actions.

If the ACL6 rule corresponding to the VLAN-based action is based on the following information, the VLAN-based action does not take effect:

- Source address and prefix of IPv6 packets
- Destination address and prefix of IPv6 packets
- Type and message code of ICMPv6 packets

----End

Configuring the Redirection Action

The redirection action redirects packets matching the traffic classification rule to the CPU of the LPU, the specified interface, the specified next hop address, or the label switching path (LSP).

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

traffic behavior *behavior-name*

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run the following commands as required.

- Run:

  redirect cpu

  The packets matching the traffic classification are redirected to the CPU.

  CAUTION

  After the **redirect cpu** command is used, the packets matching the traffic classification rule are redirected to the CPU, causing CPU performance to deteriorate. Exercise caution when you run the **redirect cpu** command.

- Run:

  redirect [**vpn-instance** *vpn-instance-name*] **ip-nexthop** *ip-address* &<1-4>

  The packets matching the traffic classification are redirected to the next hop.

  If multiple next hop IP addresses are configured, the S7700 redirects packets in active/standby mode. A maximum of four next hop IP addresses can be configured in a traffic behavior. The S7700 determines the primary path and backup paths according to the sequence in which next hop IP addresses were configured. The next hop IP address that was configured first has the highest priority and this next hop is used as the primary path. Other next hops are used as backup paths. When the primary path is Down, the backup path with the highest priority is used as the primary path.

  NOTE

  The policy-based routing function can be implemented by configuring redirection.

- Run:

  redirect ip-multihop { **nexthop** *ip-address* } &<2-4>

  The packets matching the traffic classification are redirected to one of the multiple next hops.

  If multiple next hops are specified, the S7700 redirects packets through the equal-cost routes that work in load balancing mode. That is, the S7700 selects a next hop by using the Hash algorithm based on the source IP addresses of the packets, regardless of the traffic volume. If the source IP addresses of the packets are the same, the S7700 forwards the packets to the same next hop regardless of the traffic volume.

  When redirecting packets to multiple next hops, the S7700 can quickly switch the link to an available outbound interface by using the Hash algorithm if the outbound interface corresponding to the current next hop becomes Down or the route changes suddenly.

  If no ARP entry corresponding to the next hop address is matched on the S7700, the **redirect ip-multihop** command can be run successfully. The S7700 forwards the packets to the original destination. The redirection function, however, is invalid until there is the corresponding ARP entry on the device.

- Run:

  redirect interface *interface-type interface-number*

  The packets matching the traffic classification are redirected to a specified interface.

  NOTE

  If traffic is redirected to an interface in Down state, traffic is lost on the interface and are not switched to the original forwarding path.

- Run:
  redirect lsp public *dest-address* { *nexthop-address* | **interface** *interface-type interface-number* | **secondary** }

  The packets matching the traffic classification are redirected to the public LSP.
- Run:
  redirect multi-trunk { **eth-trunk** *trunk-id* } &<1-4>

  The packets matching the traffic classifier are redirected to one or more Eth-Trunks.

  If the inbound interface is the high-speed interface (for example, XGE interface) but the outbound interface is the low-speed interface (for example, GE interface), packets need to be redirected to physical interfaces of multiple Eth-Trunks. The traffic can be load balanced and packet loss is prevented. For details, see Configuring Traffic Distribution Based on the Eth-Trunk.

NOTE

In a traffic behavior, the **remark destination-mac** command cannot be used with the following commands simultaneously:

- **redirect ip-nexthop**
- **redirect ip-multihop**

----End

Configuring Traffic Policing

Traffic policing discards the packets that exceed the rate limit or re-marks colors or CoS of these packets.

Procedure

Step 1 Run:
system-view

The system view is displayed.

Step 2 Run:
traffic behavior *behavior-name*

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run:
car cir *cir-value* [**pir** *pir-value*] [**cbs** *cbs-value* **pbs** *pbs-value*] [**share**] [**mode** { **color-blind** | **color-aware** }] [**green** { **discard** | **pass** [**service-class** *class* **color** *color*] } | **yellow** { **discard** | **pass** [**service-class** *class* **color** *color*] } | **red** { **discard** | **pass** [**service-class** *class* **color** *color*] }]*

The CAR action is configured.

Step 4 (Optional) Run:
car *car-name* **share**

The aggregate CAR action is configured.

NOTE

Before configuring aggregate CAR, run the **qos car** command to configure a CAR profile.

----End

Disabling URPF

Context

After the URPF function is enabled on an interface, the S7700 performs the URPF check on all traffic passing through the interface. To prevent the packets of a certain type from being discarded, you can disable the URPF check for these packets. For example, if the S7700 is configured to trust all the packets from a server, the S7700 does not check these packets.

NOTE

S-series boards do not support the URPF function.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
traffic behavior behavior-name
```

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run:

```
ip urpf disable
```

The URPF function is disabled.

By default, the URPF function is enabled in a traffic behavior.

After URPF is disabled, associate the traffic behavior containing URPF disabling and a traffic classifier with a traffic policy. When the traffic policy is applied globally or applied to a board, an interface, or a VLAN, the system does not perform URPF check on the flows matching the traffic classification rule.

NOTE

For details on URPF, see Configuring URPF in the *S7700 Smart Routing Switch Configuration Guide - Security*.

----End

Configuring Flow Mirroring

The flow mirroring action mirrors all the packets matching traffic classification rules to the observing interface or the CPU of the LPU.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
traffic behavior behavior-name
```

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run the following commands as required.

- Run:
  mirroring to observe-port *observe-port-index*

  All the flows that match a traffic classifier are mirrored to an observing interface.

  NOTE

  Run either the **observe-port (local mirroring)** or the **observe-port (remote mirroring)** command to create an observing interface before you configure the mirroring of flows.

- Run:
  mirroring to cpu

  All the flows that match a traffic classifier are mirrored to the CPU of the LPU.

NOTE

For details about flow mirroring, see Mirroring Configuration in the *S7700 Smart Routing Switch Configuration Guide - Device Management*.

----End

Configuring the Action of Adding Outer VLAN Tags to Packets

This action adds outer VLAN tags to packets matching traffic classification rules.

Procedure

Step 1 Run:
system-view

The system view is displayed.

Step 2 Run:
traffic behavior *behavior-name*

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run:
nest top-most vlan-id *vlan-id*

The action of adding an outer VLAN tag is configured.

NOTE

The **nest top-most vlan-id** command is used to implement the flow-based QinQ function. For description and configuration of flow-based QinQ, see QinQ Configuration in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

----End

Configuring Traffic Statistics

The traffic statistics action collects traffic statistics on packets matching traffic classification rules.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

traffic behavior *behavior-name*

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run:

statistic enable

The traffic statistics function is enabled.

NOTE

To collect statistics about packets matching a classifier, enable the traffic statistics function in the bound traffic behavior view.

----End

Disabling MAC Address Learning

After MAC address learning is disabled, MAC addresses of the packets that match traffic classification rules are not learned. This improves device efficiency and protects device security.

Context

When a network is running stably and the MAC address of packets is fixed, a device does not need to learn MAC addresses of other packets. You can apply a traffic policy and disable MAC address learning in all the traffic classifiers bound to the traffic policy. This saves MAC addresses are saved and improves device performance.

Unauthorized users may change MAC addresses frequently to attack a network. To prevent MAC address overflow and protect the network from such attacks, apply a traffic policy and disable MAC address learning in all the traffic classifiers bound to the traffic policy.

Procedure

Step 1 Run:

system-view

The system view is displayed.

Step 2 Run:

traffic behavior *behavior-name*

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run:

mac-address learning disable

MAC address learning is disabled.

After MAC address learning is disabled, MAC addresses of the packets that match the traffic classifier are not learned. The MAC addresses of the packets that do not match the traffic classifier are still learned by default.

NOTE

The **mac-address learning disable** command is only supported by the .

Disable MAC address learning in the following situations:

- To disable MAC address learning on an interface, in a port group, or in a VLAN, run the **mac-address learning disable** command in the corresponding view.
- To disable MAC address learning for the packets matching a specified traffic classifier, run the **mac-address learning disable** command in the traffic behavior view.

----End

1.4.4 Configuring a Traffic Policy

You can associate a traffic classifier with a traffic behavior in a traffic policy.

Context

When creating a traffic policy on the , specify the matching order of traffic classifiers in the traffic policy. The matching order includes the automatic order and configuration order:

- If the automatic order is used, traffic classifiers are matched based on their priorities. The priority order is: Layer 2 and Layer 3 information > Layer 2 information > Layer 3 information. The traffic classifier with the highest priority is matched first.
- If the configuration order is used, traffic classifiers are matched in the sequence in which they were bound to the traffic policy. The traffic classifier that was bound to the traffic policy first is matched first.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
traffic policy policy-name [ match-order { auto | config } ]
```

A traffic policy is created and the traffic policy view is displayed.

Step 3 Run:

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

A traffic classifier is bound to a traffic behavior in the traffic policy.

----End

1.4.5 Applying the Traffic Policy

The configured traffic policy takes effect only after being applied to the system, a slot, an interface, or a VLAN.

Context

NOTE

An LPU may not support a traffic policy; therefore, applying the traffic policy in the system or in a VLAN on the LPU fails. Run the **display traffic-policy applied-record** [*policy-name*] command to view the LPU where the traffic policy takes effect.

On the X40SFC, if ports among ports 1 to 20 and ports 21 to 40 are added to the same Eth-Trunk or VLAN, when the **car** command is used in the outbound direction of the Eth-Trunk or VLAN, the rate of outgoing traffic in the Eth-Trunk or VLAN is two times the configured CIR value.

Procedure

- Applying a traffic policy to the system or a slot
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     traffic-policy policy-name global { inbound | outbound } [ slot slot-id ]
     ```
     A traffic policy is applied to the system or a slot in the inbound or outbound direction.

     Only one traffic policy can be applied to the system in the inbound or outbound direction.

     Only one traffic policy can be applied to a slot in the inbound or outbound direction. A traffic policy cannot be applied to the system and a slot simultaneously.

     - After a traffic policy is applied to the system, the system performs traffic policing for all the packets that match traffic classification rules in the inbound or outbound direction.
     - After a traffic policy is applied to a slot, the system performs traffic policing for all the incoming or outgoing packets that match traffic classification rules on the slot.
- Applying a traffic policy to an interface
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     interface interface-type interface-number [.subnumber ]
     ```
     The interface view is displayed.
  3. Run:
     ```
     traffic-policy policy-name { inbound | outbound }
     ```
     A traffic policy is applied to the interface in the inbound or outbound direction.

     Only one traffic policy can be applied to an interface in the inbound or outbound direction.

     After a traffic policy is applied, the system performs traffic policing for the packets that pass through this interface and match traffic classification rules in the inbound or outbound direction.

NOTE

It is recommended that you should not use the traffic policy containing the re-marking of the 802.p priority, the inner VLAN tag of QinQ packets, and the VLAN ID of packets in a VLAN on the untagged interface in the outbound direction; otherwise, the information carried in the packets may be incorrect.

On a sub-interface, a traffic policy can be applied only to the inbound direction.

On X40SFC, when an interface among interfaces 1-20 and an interface among interfaces 21-40 are added to the same Eth-Trunk or VLAN, the outgoing traffic rate of the Eth-Trunk or VLAN is limited by **car**. The outgoing traffic rate is 2 times the CAR value.

- Applying a traffic policy to a VLAN
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     vlan vlan-id
     ```
     The VLAN view is displayed.
  3. Run:
     ```
     traffic-policy policy-name { inbound | outbound }
     ```
     A traffic policy is applied to the VLAN in the inbound or outbound direction.

     Only one traffic policy can be applied to a VLAN in the inbound or outbound direction.

     After a traffic policy is applied, the system performs traffic policing for the packets that belong to a VLAN and match traffic classification rules in the inbound or outbound direction.

----End

Follow-up Procedure

NOTE

After a traffic policy is applied, you do not need to re-apply the traffic policy when you add, modify, or delete a matching rule of the traffic classifier bound to the traffic policy, an action of the traffic behavior bound to the traffic policy, or a pair of the traffic classifier and traffic behavior bound to the traffic policy.

1.4.6 Checking the Configuration

After a traffic policy based on complex traffic classification is configured, you can view the configuration of the traffic classifier, traffic behavior, and traffic policy.

Prerequisites

The configurations of the traffic policy based on complex traffic classification are complete.

Procedure

- Run the **display acl** { *acl-number* | **all** } command to check the ACL rules.
- Run the **display traffic classifier user-defined** [*classifier-name*] command to check the traffic classifier on the S7700.
- Run the **display traffic behavior user-defined** [*behavior-name*] command to check the traffic behavior configuration.

- Run the **display traffic policy user-defined** [*policy-name* [**classifier** *classifier-name*]] command to check the traffic policy information.
- Run the **display traffic-policy applied-record** [*policy-name*] command to check the applied traffic policy.

----End

1.5 Maintaining Class-based QoS

If the traffic statistics function is enabled, you can view and clear the flow-based traffic statistics.

1.5.1 Displaying the Flow-based Traffic Statistics

You can use the **display traffic policy statistics** command to view the traffic statistics matching the specified traffic classification rule.

Context

To view the flow-based traffic statistics, a traffic policy must exist and contain the traffic statistics action.

Procedure

- Run the **display traffic policy statistics** { **global** [**slot** *slot-id*] | **interface** *interface-type interface-number* | **vlan** *vlan-id* } { **inbound** | **outbound** } [**verbose** { **classifier-base** | **rule-base** } [**class** *classifier-name*]] command to check the flow-based traffic statistics.

----End

1.5.2 Clearing the Flow-based Traffic Statistics

You can use the **reset** command to clear the flow-based traffic statistics.

Context

CAUTION

The flow-based traffic statistics cannot be restored after being cleared. Exercise caution when you run the command.

Procedure

- Run the **reset traffic policy statistics** { **global** [**slot** *slot-id*] | **interface** *interface-type interface-number* | **vlan** *vlan-id* } { **inbound** | **outbound** } command in the user view to clear the flow-based traffic statistics.

----End

1.6 Configuration Examples

This section provides several configuration examples of class-based QoS.

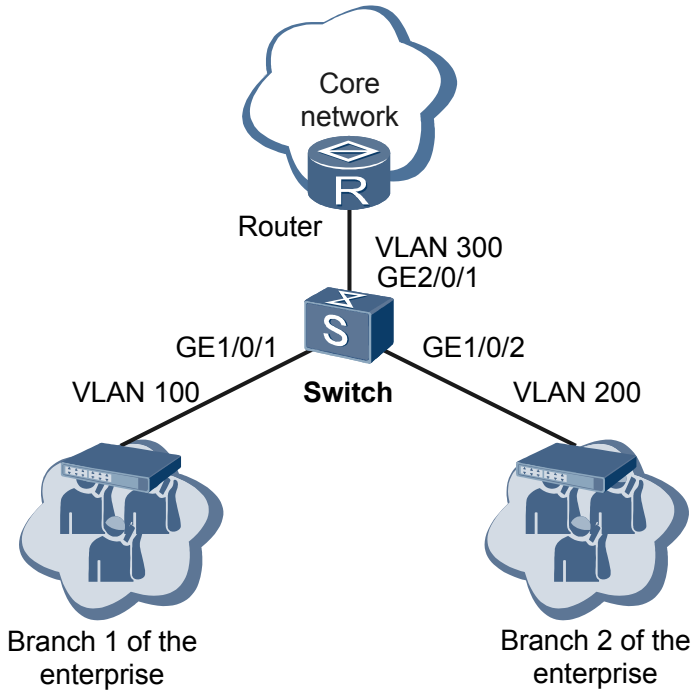
1.6.1 Example for Configuring Priority Mapping Based on Simple Traffic Classification

After priority mapping based on simple traffic classification is configured, the S7700 maps 802.1p priorities of packets to different CoS to provide differentiated services.

Networking Requirements

As shown in **Figure 1-1**, the Switch is connected to the router through GE 2/0/1; Branch 1 and Branch 2 of the enterprise access the network through the Switch and router. Branch 1 and Branch 2 of the enterprise belong to VLANs 100 and 200. Branch 1 requires better QoS guarantee; therefore, the priority of data packets from Branch 1 is mapped to 4 and the priority of data packets from Branch 2 is mapped to 2. By doing this, Switch provides differentiated services.

Figure 1-1 Networking diagram of priority mapping based on simple traffic classification



Configuration Roadmap

The configuration roadmap is as follows:

1. Create VLANs and configure interfaces so that Branch 1 and Branch 2 of the enterprise can access the network through the Switch.

2. Create DiffServ domains and map 802.1p priorities to PHBs and colors.
3. Bind the DiffServ domain to inbound interfaces GE1/0/1 and GE 1/0/2 on the Switch.

Data Preparation

To complete the configuration, you need the following data:

- Names of DiffServ domains
- 802.1p priorities of packets from Branch 1 and Branch 2 of the enterprise
- CoS of Branch 1 and Branch 2 of the enterprise

Procedure

Step 1 Create VLANs and configure interfaces.

Create VLANs 100, 200, and 300.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 100 200 300
```

Configure the type of GE1/0/1 , GE 1/0/2, and GE 2/0/1 as trunk, add GE1/0/1 to VLAN 100, add GE 1/0/2 to VLAN 200, and add GE 2/0/1 to VLAN 100, VLAN 200, and VLAN 300.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] port link-type trunk
[Switch-GigabitEthernet1/0/1] port trunk allow-pass vlan 100
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] port link-type trunk
[Switch-GigabitEthernet1/0/2] port trunk allow-pass vlan 200
[Switch-GigabitEthernet1/0/2] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 100 200 300
[Switch-GigabitEthernet2/0/1] quit
```

Create VLANIF 300 and assign interface IP address uo192.168.1.1/24 to VLANIF 300.

```
[Switch] interface vlanif 300
[Switch-Vlanif300] ip address 192.168.1.1 24
```

NOTE

Assign IP address 192.168.1.2/24 to the interface connecting the router and the Switch.

Step 2 Create and configure DiffServ domains.

Create DiffServ domains **ds1** and **ds2** and map 802.1p priorities of packets from Branch 1 and Branch 2 of the enterprise to PHBs and colors.

```
[Switch] diffserv domain ds1
[Switch-dsdomain-ds1] 8021p-inbound 0 phb af4 green
[Switch-dsdomain-ds1] quit
[Switch] diffserv domain ds2
[Switch-dsdomain-ds2] 8021p-inbound 0 phb af2 green
[Switch-dsdomain-ds2] quit
```

Step 3 Bind DiffServ domains to interfaces.

Bind DiffServ domains **ds1** and **ds2** to GE1/0/1 , and GE 1/0/2 respectively.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] trust upstream ds1
```

```
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] trust upstream ds2
[Switch-GigabitEthernet1/0/2] quit
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
vlan batch 100 200 300
#
diffserv domain ds1
  8021p-inbound 0 phb af4 green
#
diffserv domain ds2
  8021p-inbound 0 phb af2 green
#
interface Vlanif300
 ip address 192.168.1.1 255.255.255.0
#
interface GigabitEthernet1/0/1
 port link-type trunk
 port trunk allow-pass vlan 100
 trust upstream ds1
#
interface GigabitEthernet1/0/2
 port link-type trunk
 port trunk allow-pass vlan 200
 trust upstream ds2
#
interface GigabitEthernet2/0/1
 port link-type trunk
 port trunk allow-pass vlan 100 200 300
#
return
```

1.6.2 Example for Re-marking the Priorities Based on Complex Traffic Classification

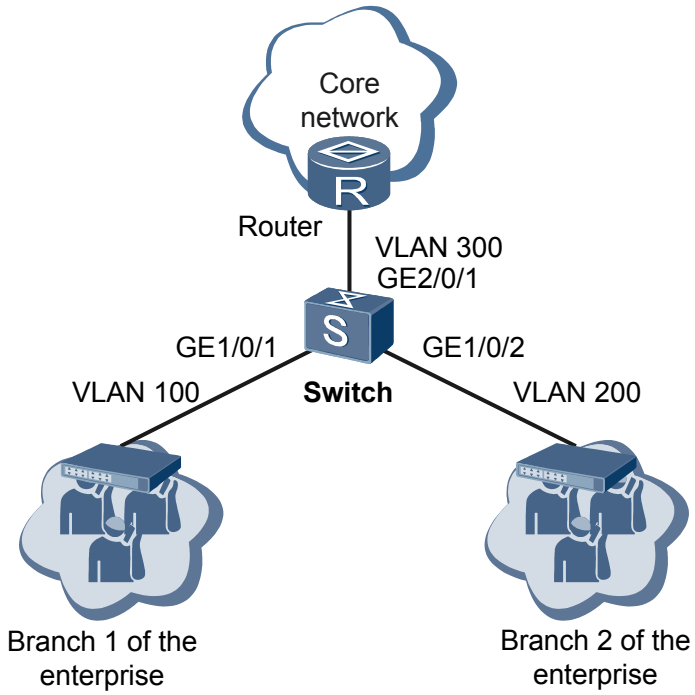
After priority re-marking based on complex traffic classification is configured, the S7700 adds the same outer VLAN ID to packets with different VLAN IDs. In addition, the S7700 re-marks different 802.1p priorities of packets with different VLAN IDs to provide differentiated services.

Networking Requirements

The Switch is connected to the router through GE2/0/1; Branch 1 and Branch 2 of the enterprise can access the network through the Switch and router. See **Figure 1-2**.

Data services of Branch 1 and Branch 2 of the enterprise come from VLANs 100 and 200. When the data service packets of Branch 1 and Branch 2 of the enterprise pass the Switch, the Switch needs to add the outer VLAN tag with the VLAN 300 to the packets so that these packets are identified as data services on the core network. In addition, Branch 1 requires better QoS guarantee; therefore, the priority of data packets to Branch 1 is mapped to 4 and the priority of data packets to Branch 2 is mapped to 2. By doing this, differentiated services are provided.

Figure 1-2 Networking diagram of priority re-marking based on complex traffic classification



Configuration Roadmap

The configuration roadmap is as follows:

1. Create VLANs and configure interfaces so that Branch 1 and Branch 2 of the enterprise can access the network through the Switch.
2. Create traffic classifiers based on the VLAN ID in the inner VLAN tag on the Switch.
3. Create traffic behaviors on the Switch and re-mark 802.1p priorities of packets.
4. Create a traffic policy on the Switch, bind traffic behaviors to traffic classifiers in the traffic policy, and apply the traffic policy to the interface at the inbound direction.

Data Preparation

To complete the configuration, you need the following data:

- Re-marked priorities of packets with different VLAN IDs in the inner VLAN tags
- Type, direction, and number of the interface that a traffic policy needs to be applied to

Procedure

Step 1 Create VLANs and configure interfaces.

Create VLANs 100, 200, and 300 on the Switch and configure the interfaces so that the Switch adds the outer VLAN tag with the VLAN ID as 300 to the packets sent from GE 1/0/1 and GE 1/0/2 and GE 2/0/1 can forward packets in VLAN 300.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 100 200 300
```

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] port hybrid pvid vlan 100
[Switch-GigabitEthernet1/0/1] port hybrid untagged vlan 100 300
[Switch-GigabitEthernet1/0/1] port vlan-stacking vlan 100 stack-vlan 300
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] port hybrid pvid vlan 200
[Switch-GigabitEthernet1/0/1] port hybrid untagged vlan 200 300
[Switch-GigabitEthernet1/0/2] port vlan-stacking vlan 200 stack-vlan 300
[Switch-GigabitEthernet1/0/2] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 300
[Switch-GigabitEthernet2/0/1] quit
```

Create VLANIF 300 and assign IP address 192.168.1.1/24 to VLANIF 300.

```
[Switch] interface vlanif 300
[Switch-Vlanif300] ip address 192.168.1.1 24
[Switch-Vlanif300] quit
```

Step 2 Create traffic classifiers.

Create traffic classifiers **c1** to **c2** on the Switch to classify incoming packets based on the VLAN ID in the inner VLAN tag.

```
[Switch] traffic classifier c1 operator and
[Switch-classifier-c1] if-match cvlan-id 100
[Switch-classifier-c1] quit
[Switch] traffic classifier c2 operator and
[Switch-classifier-c2] if-match cvlan-id 200
[Switch-classifier-c2] quit
```

Step 3 Create traffic behaviors.

Create traffic behaviors **b1** to **b2** on the Switch to re-mark priorities of user packets.

```
[Switch] traffic behavior b1
[Switch-behavior-b1] remark 8021p 4
[Switch-behavior-b1] quit
[Switch] traffic behavior b2
[Switch-behavior-b2] remark 8021p 2
[Switch-behavior-b2] quit
```

Step 4 Create a traffic policy and apply it to an interface.

Create traffic policy **p1** on the Switch, bind traffic classifiers to traffic behaviors in the traffic policy, and apply the traffic policy to GE 1/0/1 and GE 1/0/2 in the inbound direction to re-mark priorities of packets coming from the user side.

```
[Switch] traffic policy p1
[Switch-trafficpolicy-p1] classifier c1 behavior b1
[Switch-trafficpolicy-p1] classifier c2 behavior b2
[Switch-trafficpolicy-p1] quit
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] traffic-policy p1 inbound
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] traffic-policy p1 inbound
[Switch-GigabitEthernet1/0/2] quit
```

Step 5 Verify the configuration.

Check the configuration of traffic classifiers.

```
<Switch> display traffic classifier user-defined
   User Defined Classifier Information:
   Classifier: c2
   Precedence: 10
```

```
    Operator: AND
    Rule(s) : if-match cvlan-id 200

   Classifier: c1
   Precedence: 5
    Operator: AND
    Rule(s) : if-match cvlan-id 100

Total classifier number is 2
```

Check the configuration of the traffic policy.

```
<Switch> display traffic policy user-defined p1
  User Defined Traffic Policy Information:
  Policy: p1
   Classifier: c1
    Operator: AND
     Behavior: b1
      Remark:
        Remark 8021p 4
   Classifier: c2
    Operator: AND
     Behavior: b2
      Remark:
        Remark 8021p 2
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 100 200 300
#
traffic classifier c2 operator and precedence 5
 if-match cvlan-id 200
traffic classifier c1 operator and precedence 10
 if-match cvlan-id 100
#
traffic behavior b2
 remark 8021p 2
traffic behavior b1
 remark 8021p 4
#
traffic policy p1
 classifier c1 behavior b1
 classifier c2 behavior b2
#
interface Vlanif300
 ip address 192.168.1.1 255.255.255.0
#
interface GigabitEthernet1/0/1
 port hybrid pvid vlan
100
 port hybrid untagged vlan 100 300
 port vlan-stacking vlan 100 stack-vlan 300
 traffic-policy p1 inbound
#
interface GigabitEthernet1/0/2
 port hybrid pvid vlan
200
 port hybrid untagged vlan 200 300
 port vlan-stacking vlan 200 stack-vlan 300
 traffic-policy p1 inbound
#
interface GigabitEthernet2/0/1
 port link-type trunk
```

```
 port trunk allow-pass vlan 300
#
return
```

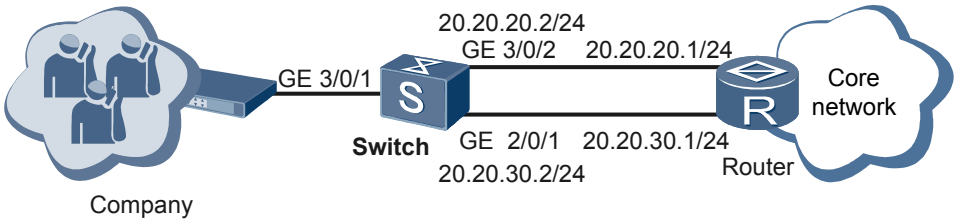
1.6.3 Example for Configuring Policy-based Routing

After packet redirection based on complex traffic classification is configured, the S7700 redirects packets with different IP priorities to different interfaces so that the S7700 provides different bandwidth services.

Networking Requirements

The Layer 2 switch of a company is connected to the ISP device through the Switch; one is a 1-Gbit/s link with the gateway as 20.20.20.1/24 and the other is a 10-Gbit/s link with the gateway as 20.20.30.1/24. The company requires that the 10 Gbit/s links send only the packets with priorities as 4, 5, 6, and 7 and 1 Gbit/s links send packets of lower priorities to the ISP. See **Figure 1-3**.

Figure 1-3 Policy-based routing networking



Configuration Roadmap

The configuration roadmap is as follows:

1. Create VLANs and configure interfaces so that the Switch can ping the ISP device.
2. Create ACL rules to match the packets with priorities as 4, 5, 6, and 7 and priorities as 0, 1, 2, and 3.
3. Create traffic classifiers to match the preceding ACL rules.
4. Create traffic behaviors to redirect matching packets to 20.20.20.1/24 and 20.20.30.1/24.
5. Create a traffic policy, bind traffic classifiers to traffic behaviors in the traffic policy, and apply the traffic policy to an interface.

Data Preparation

To complete the configuration, you need the following data:

- VLAN 20 and VLAN 30 that all of GE1/0/1, GE1/0/2 and GE2/0/1 are added to
- ACL rules 3001 and 3002
- Traffic classifiers **c1** and **c2**
- Traffic behaviors **b1** and **b2**
- Traffic policy **p1**

Procedure

Step 1 Create VLANs and configure interfaces.

Create VLANs 20 and 30.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 20 30
```

Configure the type of GE 1/0/1, GE 1/0/2 and GE 2/0/1 to trunk, and add all of GE 1/0/1, GE 1/0/2 and GE 2/0/1 to VLAN 20 and VLAN 30.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] port link-type trunk
[Switch-GigabitEthernet1/0/1] port trunk allow-pass vlan 20 30
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] port link-type trunk
[Switch-GigabitEthernet1/0/2] port trunk allow-pass vlan 20 30
[Switch-GigabitEthernet1/0/2] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 20 30
[Switch-GigabitEthernet2/0/1] quit
```

Create VLANIF 20 and VLANIF 30 and assign IP addresses to them.

```
[Switch] interface vlanif 20
[Switch-Vlanif20] ip address 20.20.20.2 24
[Switch-Vlanif20] quit
[Switch] interface vlanif 30
[Switch-Vlanif30] ip address 20.20.30.2 24
[Switch-Vlanif30] quit
```

NOTE

Assign network segment addresses 20.20.20.1/24 and 20.20.30.1/24 to the interfaces connecting the router and Switch. The details are not mentioned here.

Step 2 Create ACL rules.

Create advanced ACL rules 3001 and 3002 on the Switch to permit the packets with priorities as 4, 5, 6, and 7 and priorities as 0, 1, 2, and 3 to pass through.

```
[Switch] acl 3001
[Switch-acl-adv-3001] rule permit ip precedence 0
[Switch-acl-adv-3001] rule permit ip precedence 1
[Switch-acl-adv-3001] rule permit ip precedence 2
[Switch-acl-adv-3001] rule permit ip precedence 3
[Switch-acl-adv-3001] quit
[Switch] acl 3002
[Switch-acl-adv-3002] rule permit ip precedence 4
[Switch-acl-adv-3002] rule permit ip precedence 5
[Switch-acl-adv-3002] rule permit ip precedence 6
[Switch-acl-adv-3002] rule permit ip precedence 7
[Switch-acl-adv-3002] quit
```

Step 3 Create traffic classifiers.

Create traffic classifiers **c1** and **c2** on the Switch with matching rules as ACL 3001 and ACL 3002.

```
[Switch] traffic classifier c1
[Switch-classifier-c1] if-match acl 3001
[Switch-classifier-c1] quit
[Switch] traffic classifier c2
```

```
[Switch-classifier-c2] if-match acl 3002
[Switch-classifier-c2] quit
```

Step 4 Create traffic behaviors.

Create traffic behaviors **b1** and **b2** on the Switch to redirect packets to network segments 20.20.20.1/24 and 20.20.30.1/24.

```
[Switch] traffic behavior b1
[Switch-behavior-b1] redirect ip-nexthop 20.20.20.1
[Switch-behavior-b1] quit
[Switch] traffic behavior b2
[Switch-behavior-b2] redirect ip-nexthop 20.20.30.1
[Switch-behavior-b2] quit
```

Step 5 Create a traffic policy and apply it to an interface.

Create traffic policy **p1** on the Switch and bind traffic classifiers to traffic behaviors in the traffic policy.

```
[Switch] traffic policy p1
[Switch-trafficpolicy-p1] classifier c1 behavior b1
[Switch-trafficpolicy-p1] classifier c2 behavior b2
[Switch-trafficpolicy-p1] quit
```

Apply traffic policy **p1** to GE 1/0/1.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] traffic-policy p1 inbound
[Switch-GigabitEthernet1/0/1] quit
```

Step 6 Verify the configuration.

Check the configuration of ACL rules.

```
[Switch] display acl 3001
Advanced ACL 3001, 4 rules
Acl's step is 5
 rule 5 permit ip precedence routine
 rule 10 permit ip precedence priority
 rule 15 permit ip precedence immediate
 rule 20 permit ip precedence flash
[Switch] display acl 3002
Advanced ACL 3002, 4 rules
Acl's step is 5
 rule 5 permit ip precedence flash-override
 rule 10 permit ip precedence critical
 rule 15 permit ip precedence internet
 rule 20 permit ip precedence network
```

Check the configuration of traffic classifiers.

```
[Switch] display traffic classifier user-defined
  User Defined Classifier Information:
   Classifier: c2
   Precedence: 10
    Operator: OR
    Rule(s) : if-match  acl 3002

   Classifier: c2
   Precedence: 5
    Operator: OR
    Rule(s) : if-match acl 3001


Total classifier number is 2
```

View the configuration of the traffic policy.

```
<Switch> display traffic policy user-defined p1
  User Defined Traffic Policy Information:
  Policy: p1
   Classifier: c1
    Operator: OR
     Behavior: b1
      Redirect: no forced
        Redirect ip-nexthop
        20.20.20.1
   Classifier: c2
    Operator: OR
     Behavior: b2
      Redirect: no forced
        Redirect ip-nexthop
        20.20.30.1
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 20 30
#
acl number 3001
 rule 5 permit ip precedence routine
 rule 10 permit ip precedence priority
 rule 15 permit ip precedence immediate
 rule 20 permit ip precedence flash
#
acl number 3002
 rule 5 permit ip precedence flash-override
 rule 10 permit ip precedence critical
 rule 15 permit ip precedence internet
 rule 20 permit ip precedence network
#
traffic classifier c1 operator or precedence 5
 if-match acl 3001
traffic classifier c2 operator or precedence 10
 if-match acl 3002
#
traffic behavior b1
 redirect ip-nexthop 20.20.20.1
traffic behavior b2
 redirect ip-nexthop 20.20.30.1
#
traffic policy p1
 classifier c1 behavior b1
 classifier c2 behavior b2
#
interface Vlanif20
 ip address 20.20.20.2 255.255.255.0
#
interface Vlanif30
 ip address 20.20.30.2 255.255.255.0
#
interface GigabitEthernet1/0/1
 port link-type trunk
 port trunk allow-pass vlan 20 30
 traffic-policy p1 inbound
#
interface GigabitEthernet1/0/2
 port link-type trunk
 port trunk allow-pass vlan 20 30
#
interface GigabitEthernet2/0/1
 port link-type trunk
```

```
 port trunk allow-pass vlan 20 30
#
return
```

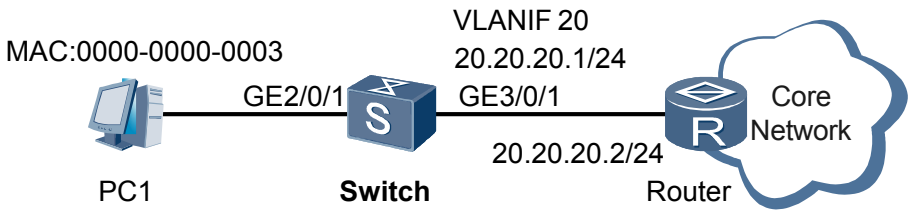
1.6.4 Example for Configuring Traffic Statistics Based on Complex Traffic Classification

After traffic statistics based on complex traffic classification is configured, the S7700 collect traffic statistics on packets with the specified source MAC address.

Networking Requirements

As shown in **Figure 1-4**, PC1 with the MAC address of 0000-0000-0003 is connected to other devices through GE1/0/1 on the Switch. The Switch is required to collect the statistics on the packets with the source MAC address of 0000-0000-0003.

Figure 1-4 Networking diagram for configuring traffic statistics based on complex traffic classification



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure interfaces so that the Switch is connected to PC1 and the router.
2. Create an ACL to match the packets with the source MAC address as 0000-0000-0003.
3. Create a traffic classifier to match the ACL.
4. Create a traffic behavior to take the statistics on the matching packets.
5. Create a traffic policy, bind the traffic classifier to the traffic behavior in the traffic policy, and apply the traffic policy to GE1/0/1 in the inbound direction.

Data Preparation

To complete the configuration, you need the following data:

- VLAN 20
- ACL 4000
- Traffic classifier **c1**
- Traffic behavior **b1**
- Traffic policy **p1**

Procedure

Step 1 Create a VLAN and configure interfaces.

Create VLAN 20.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan 20
[Switch-vlan20] quit
```

Configure the type of GE1/0/1 as access and GE2/0/1 as trunk, and add GE1/0/1 and GE2/0/1 to VLAN 20.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] port link-type access
[Switch-GigabitEthernet1/0/1] port default vlan 20
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 20
[Switch-GigabitEthernet2/0/1] quit
```

Create VLANIF 20 and assign IP address 20.20.20.1/24 to it.

```
[Switch] interface vlanif 20
[Switch-Vlanif20] ip address 20.20.20.1 24
[Switch-Vlanif20] quit
```

NOTE

Assign network segment address 20.20.20.2/24 to the interface connecting the router and Switch. The details are not mentioned here.

Step 2 Create an ACL.

Create Layer 2 ACL 4000 on the Switch to match the packets with the source MAC address as 0000-0000-0003.

```
[Switch] acl 4000
[Switch-acl-L2-4000] rule permit source-mac 0000-0000-0003 ffff-ffff-ffff
[Switch-acl-L2-4000] quit
```

Step 3 Create a traffic classifier.

Create traffic classifier **c1** on the Switch with ACL 4000 as the matching rule.

```
[Switch] traffic classifier c1
[Switch-classifier-c1] if-match acl 4000
[Switch-classifier-c1] quit
```

Step 4 Create a traffic behavior.

Create traffic behavior **b1** on the Switch and configure the traffic statistics action.

```
[Switch] traffic behavior b1
[Switch-behavior-b1] statistic enable
[Switch-behavior-b1] quit
```

Step 5 Create a traffic policy and apply it to an interface.

Create traffic policy **p1** on the Switch and bind the traffic classifier to the traffic behavior in the traffic policy.

```
[Switch] traffic policy p1
[Switch-trafficpolicy-p1] classifier c1 behavior b1
[Switch-trafficpolicy-p1] quit
```

Apply traffic policy **p1** to GE1/0/1.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] traffic-policy p1 inbound
[Switch-GigabitEthernet1/0/1] quit
[Switch] quit
```

Step 6 Verify the configuration.

Check the configuration of the ACL.

```
<Switch> display acl 4000
L2 ACL 4000, 1 rule
Acl's step is 5
 rule 5 permit source-mac 0000-0000-0003
```

Check the configuration of the traffic classifier.

```
<Switch> display traffic classifier user-defined
  User Defined Classifier Information:
   Classifier: c1
    Operator: OR
    Rule(s) : if-match 5 acl 4000
Total classifier number is 1
```

View the configuration of the traffic policy.

```
<Switch> display traffic policy user-defined p1
  User Defined Traffic Policy Information:
  Policy: p1
   Classifier: c1
   Precedence: 5
    Operator: OR
     Behavior: b1
      Statistic: enable
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 20
#
acl number 4000
 rule 5 permit source-mac 0000-0000-0003
#
traffic classifier c1 operator or precedence 5
 if-match acl 4000
#
traffic behavior b1
 statistic enable
#
traffic policy p1
 classifier c1 behavior b1
#
interface Vlanif20
 ip address 20.20.20.1 255.255.255.0
#
interface GigabitEthernet1/0/1
 port link-type access
 port default vlan 20
 traffic-policy p1 inbound
#
interface GigabitEthernet2/0/1
 port link-type trunk
 port trunk allow-pass vlan 20
```

```
#
return
```

1.6.5 Example for Filtering Packets Based on Complex Traffic Classification

By configuring packet filtering based on complex traffic classification, the S7700 only permits the packets with the specified source MAC address to pass through.

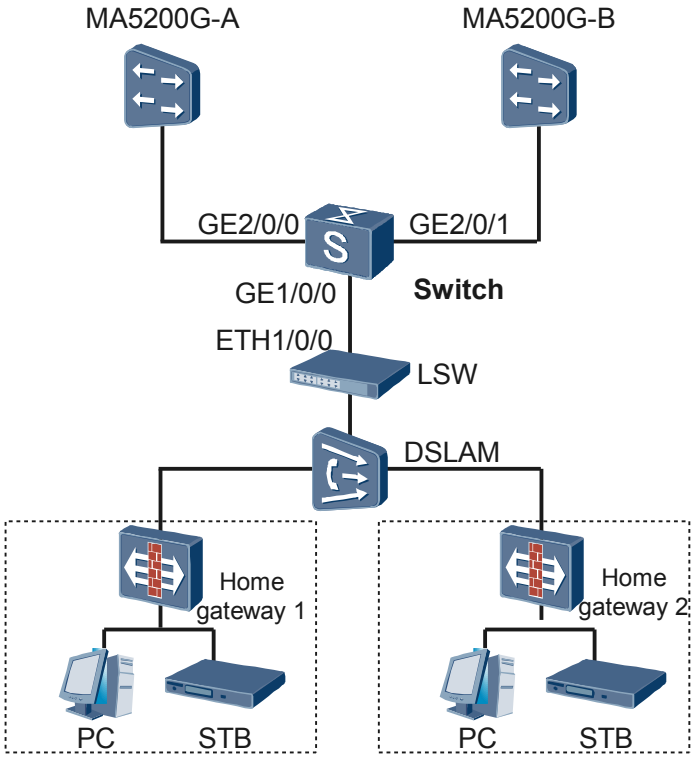
Networking Requirements

As shown in **Figure 1-5**, the PC and STB are connected to upstream BRAS devices MA5200G-A and MA5200G-B through the home gateway, DSLAM, LSW, and Switch. The PC and STB access the network by obtaining IP addresses through BRAS devices. It is required that the IP address of the PC is obtained from MA5200G-A and the IP address of the STB is obtained from MA5200G-B.

In **Figure 1-5**:

- The PC is connected to the network through PPPoE dialup and the STB is connected to the network through triggering of DHCP messages. The DHCP messages are sent when the PC is started.
- The MAC address (00e0-8e00-0000/ffff-ff00-0000) of the PC cannot be controlled, and the MAC address of the STB (00c0-8c00-0000/ffff-ff00-0000) can be controlled.
- The MAC addresses of home gateway 1 and home gateway 2 are 00d0-f800-0000/ffff-ff00-0000 and 00d0-d000-0000/ffff-ff00-0000.
- The DSLAM adds VLAN 100 to PPPoE packets and VLAN 200 to DHCP messages.

Figure 1-5 Networking diagram for filtering packets based on complex traffic classification



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure each interface to connect the switches.
2. Configure ACL rules to match the MAC address of the STB.
3. Configure the traffic classifier to match ACL rules and VLAN.
4. Configure the traffic behavior to allow packets that matching rules.
5. Configure the traffic policy in which the traffic classifier is bound to the traffic behavior and apply the traffic policy to GE 1/0/0 at the inbound direction.

Data Preparation

To complete the configuration, you need the following data:

- VLAN IDs 100 and 200
- ACL IDs 4000 and 4001
- Name of the traffic classifier **tc1** and **tc2**
- Name of the traffic behavior **tb**
- Name of the traffic policy **tp**

Procedure

Step 1 Create a VLAN and configure each interface.

Create VLAN 100 and VLAN 200.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 100 200
```

Set the type of GE 1/0/0, GE 2/0/0, and GE 2/0/1 to trunk and add them to VLAN 100 and VLAN 200.

```
[Switch] interface gigabitethernet 1/0/0
[Switch-GigabitEthernet1/0/0] port link-type trunk
[Switch-GigabitEthernet1/0/0] port trunk allow-pass vlan 100 200
[Switch-GigabitEthernet1/0/0] quit
[Switch] interface gigabitethernet 2/0/0
[Switch-GigabitEthernet2/0/0] port link-type trunk
[Switch-GigabitEthernet2/0/0] port trunk allow-pass vlan 100
[Switch-GigabitEthernet2/0/0] quit
[Switch] interface gigabitethernet 2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 200
[Switch-GigabitEthernet2/0/1] quit
```

Step 2 Configure ACL rules.

Create Layer 2 ACL 4000 on the Switch, which allows only the packets with the source MAC address being the MAC address of the STB to pass through.

```
[Switch] acl 4000
[Switch-acl-L2-4000] rule permit source-mac 00e0-8e00-0000 ffff-ff00-0000
[Switch-acl-L2-4000] rule permit source-mac 00c0-8c00-0000 ffff-ff00-0000
[Switch-acl-L2-4000] rule deny
[Switch-acl-L2-4000] quit
```

Create Layer 2 ACL 4001 on the Switch, which allows only the packets with the source MAC address being the MAC address of the home gateway to pass through.

```
[Switch] acl 4001
[Switch-acl-L2-4001] rule permit source-mac 00d0-f800-0000 ffff-ff00-0000
[Switch-acl-L2-4001] rule permit source-mac 00d0-d000-0000 ffff-ff00-0000
[Switch-acl-L2-4001] rule deny
[Switch-acl-L2-4001] quit
```

Step 3 Configure traffic classifiers.

Configure the traffic classifier **tc1** on the Switch to match ACL 4000 and VLAN 200.

```
[Switch] traffic classifier tc1 operator and
[Switch-classifier-tc1] if-match acl 4000
[Switch-classifier-tc1] if-match vlan-id 200
[Switch-classifier-tc1] quit
```

Configure the traffic classifier **tc2** on the Switch to match ACL 4001 and VLAN 200.

```
[Switch] traffic classifier tc2 operator and
[Switch-classifier-tc2] if-match acl 4001
[Switch-classifier-tc2] if-match vlan-id 200
[Switch-classifier-tc2] quit
```

Step 4 Configure the traffic behavior.

Configure the traffic behavior **tb** on the Switch and set the **permit** mode.

```
[Switch] traffic behavior tb
[Switch-behavior-tb] permit
[Switch-behavior-tb] quit
```

Step 5 Apply the traffic policy to the interface.

Create the traffic policy **tp** on the Switch, in which the traffic classifier is bound to the traffic behavior.

```
[Switch] traffic policy tp
[Switch-trafficpolicy-tp] classifier tc1 behavior tb
[Switch-trafficpolicy-tp] classifier tc2 behavior tb
[Switch-trafficpolicy-tp] quit
```

Apply the traffic policy **tp** to GE 1/0/0.

```
[Switch] interface gigabitethernet 1/0/0
[Switch-GigabitEthernet1/0/0] traffic-policy tp inbound
[Switch-GigabitEthernet1/0/0] quit
[Switch] quit
```

Step 6 Verify the configuration.

Check the configuration of ACL rules.

```
<Switch> display acl 4000
L2 ACL 4000, 3 rules
Acl's step is 5
 rule 5 permit source-mac 00e0-8e00-0000 ffff-ff00-0000
 rule 10 permit source-mac 00c0-8c00-0000 ffff-ff00-0000
 rule 15 deny
```

Check the configuration of the traffic classifier.

```
<Switch> display traffic classifier user-defined
  User Defined Classifier Information:
   Classifier: tc2
    Precedence: 10
    Operator: AND
    Rule(s) : if-match acl 4001
```

```
            if-match vlan-id 200

   Classifier: tc1
    Precedence: 5
    Operator: AND
    Rule(s) : if-match acl 4000
              if-match vlan-id 200

Total classifier number is 2
```

Check the configuration of the traffic policy.

```
<Switch> display traffic policy user-defined tp
  User Defined Traffic Policy Information:
  Policy: tp
   Classifier: tc1
    Operator: AND
     Behavior: tb
      Permit
   Classifier: tc2
    Operator: AND
     Behavior: tb
      Permit
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 100 200
#
acl number 4000
 rule 5 permit source-mac 00e0-8e00-0000 ffff-ff00-0000
 rule 10 permit source-mac 00c0-8c00-0000 ffff-ff00-0000
 rule 15 deny
#
acl number 4001
 rule 5 permit source-mac 00d0-f800-0000 ffff-ff00-0000
 rule 10 permit source-mac 00d0-d000-0000 ffff-ff00-0000
 rule 15 deny
#
traffic classifier tc1 operator and precedence 5
 if-match acl 4000
 if-match vlan-id 200
traffic classifier tc2 operator and precedence 10
 if-match acl 4001
 if-match vlan-id 200
#
traffic behavior tb
 permit
#
traffic policy tp
 classifier tc1 behavior tb
 classifier tc2 behavior tb
#
interface GigabitEthernet1/0/0
 port link-type trunk
 port trunk allow-pass vlan 100 200
 traffic-policy tp inbound
#
interface GigabitEthernet2/0/0
 port link-type trunk
 port trunk allow-pass vlan 100
#
interface GigabitEthernet2/0/1
 port link-type trunk
```

```
 port trunk allow-pass vlan 200
#
return
```

2 Traffic Policing and Traffic Shaping Configuration

About This Chapter

This document describes basic concepts of traffic policing and traffic shaping. It also describes the configuration method of the Interface-based Traffic Policing, the configuration method of traffic policing based on a traffic classifier, and provides traffic shaping, and provides configuration examples.

2.1 Traffic Policing and Traffic Shaping Overview

This section describes the basic concepts of traffic policing and traffic shaping and the differences between traffic policing and traffic shaping.

2.1.1 Traffic Policing

To make full use of limited network resources, perform traffic policing for special service flows to adapt to the allocated network resources.

Traffic policing monitors the rate limit to limit the traffic and resource usage. It then discards the excess traffic to limit traffic within a proper range and to protect network resources.

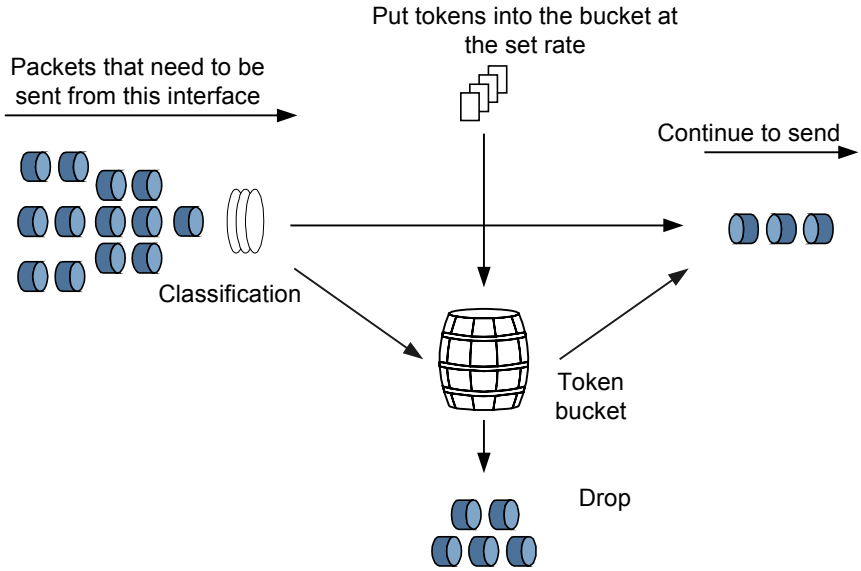
Token Bucket and Traffic Measurement

When the traffic exceeds the rate limit, the S7700 uses traffic control policies. Generally, the S7700 uses a token bucket to measure the volume of traffic.

A token bucket is considered as a container that stores a certain number of tokens. The S7700 puts tokens at the configured rate (one token bucket can forward one bit of data) in a token bucket. When the token bucket is full, the excess tokens overflow and the number of tokens no longer increases.

When measuring the traffic in a token bucket, the S7700 forwards packets based on the number of tokens in the token bucket. If there are sufficient tokens in the token bucket to forward packets, the traffic rate is within the rate limit. Otherwise, the traffic rate exceeds the rate limit.

Figure 2-1 Using a token bucket to measure the traffic



The S7700 supports the single token bucket and dual token buckets.

- Single token bucket

  The single token bucket technology uses the following parameters:

  - Committed burst size (CBS): indicates the maximum volume of traffic that bursts in bucket C, in bytes.
  - Committed information rate (CIR): indicates the rate of tokens that are put into bucket C, that is, the average traffic rate allowed by bucket C, in kbit/s.

  If there are sufficient tokens in the bucket, packets are forwarded. At the same time, the number of tokens in the bucket decreases based on the length of the packets. If there are no tokens in the bucket, packets are discarded.

- Dual token buckets

  The dual token bucket technology uses the following parameters in addition to the CIR and CBS:

  - Peak burst size (PBS): indicates the maximum volume of traffic that bursts and exceeds the CBS in bucket P, in bytes.
  - Peak information rate (PIR): indicates the rate of tokens that are put into bucket P, that is, the average traffic rate allowed by bucket P, in kbit/s.

  For the dual token buckets:

  - The service traffic that is less than the CIR value is colored green and is allowed to pass through.
  - The service traffic that exceeds the PIR value is colored red and is discarded.
  - The service traffic that ranges from the CIR value to the PIR value is colored yellow and is discarded when congestion occurs.

Traffic Policing Features Supported by the S7700

The S7700 supports the following traffic policing features:

- Interface-based traffic policing.

  Interface-based traffic policing controls all incoming traffic on an interface regardless of packet types. It discards the excess traffic, limits traffic within a proper range, and protects network resources and carriers' interests.

- Traffic policing based on a traffic classifier

  Traffic policing based on a traffic classifier limits the rate of the traffic matching a traffic classifier. The S7700 limits the rate of incoming traffic. It discards the traffic that exceeds the rate limit, limits traffic within an appropriate range, and protects network resources and carriers' interests. Traffic policing based on a traffic classifier uses dual token buckets.

  After traffic policing based on a traffic classifier is configured on an S7700, CAR can be performed twice for upstream flows. The S7700 first applies CAR to the upstream flows that match a traffic classifier, and then it aggregates all the upstream flows and applies CAR to limit the aggregated flows. The upstream flows refer to the incoming service flows matching a traffic classifier that is bound to a traffic behavior containing aggregate CAR.

  NOTE

  - Aggregate CAR supports only the single token bucket.
  - Traffic policing based on a traffic classifier on the S7700 implements interface-based and flow-based rate limiting in both directions. The matching rule is set to **if-match any**.

- CPCAR

  On a network, a large number of packets, including valid packets and malicious attack packets, are delivered to the CPU. The malicious attack packets affect other services or

even interrupt the system. Transmitting excess valid packets also leads to high CPU usage, which deteriorates CPU performance and interrupts services.

To protect the CPU, limit the packets sent to the CPU. CPCAR performs traffic policing on certain packets sent to the CPU, for example, host packets, blacklisted service flows, and user-defined flows. It limits the transmission rate of the packets, protects the CPU, and ensures that the system keeps running.

When the traffic of host packets is heavy or a large number of malicious host packets are sent, the system performance may deteriorate. To ensure system running, configure CPCAR on the S7700 to limit the traffic of host packets.

The S7700 can limit the traffic of packets that are sent to the CPU of the main control board or LPU.

CPCAR uses the single token bucket.

For details about host packets, blacklisted flows, user-defined flows, and CPCAR configuration, see Local Attack Defense Configuration in the *S7700 Smart Routing Switch Configuration Guide - Security*.

2.1.2 Traffic Shaping

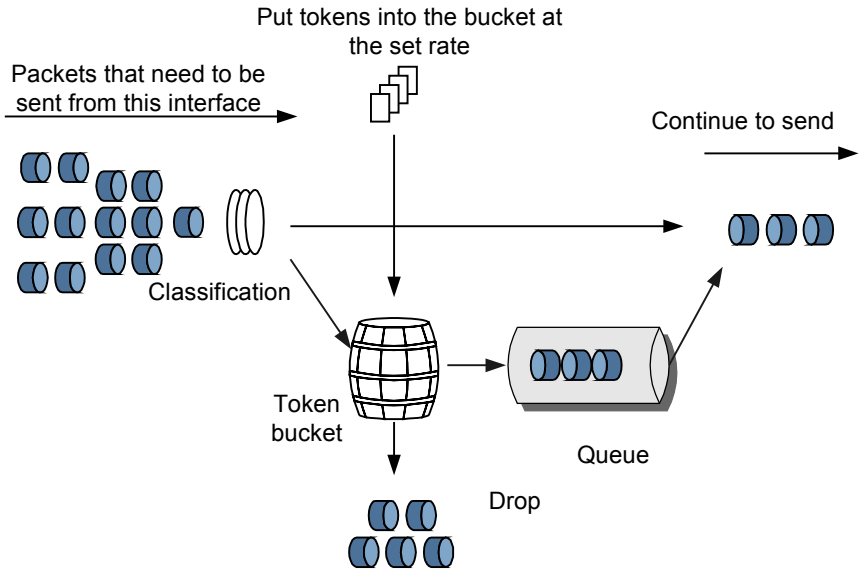
Traffic shaping controls the rate of packets so that packets are sent at an even rate. It adapts the transmission rate of packets to the downstream devices to prevent unnecessary packet loss and congestion.

Traffic shaping also limits traffic and resources by monitoring the traffic rate. In traffic shaping, the S7700 also uses token buckets to measure the traffic.

Difference Between Traffic Shaping and Traffic Policing

The main difference between traffic shaping and traffic policing is that the S7700 caches the packets discarded in traffic policing. These packets are stored in a buffer or a queue, as shown in **Figure 2-2**. When there are sufficient tokens in a token bucket, those cached packets are sent out at an average rate.

Figure 2-2 Networking diagram of traffic shaping



The delay may be increased just because the traffic shaping technology puts the packets into a buffer or a queue. The traffic policing technology, however, does not cause a delay.

Traffic Shaping Features Supported by the S7700

The S7700 supports the following traffic shaping features:

- Traffic shaping on an interface

  The S7700 performs traffic shaping for all the packets that pass through an interface.
- Traffic shaping in an interface queue

  The S7700 performs traffic shaping for the packets of a certain type that pass through an interface based on simple traffic classification. In this manner, traffic shaping based on voice, data, and video services is implemented.

2.2 Configuring Interface-based Traffic Policing

After interface-based traffic policing is configured, the S7700 performs traffic policing on all service traffic on the interface.

2.2.1 Establishing the Configuration Task

Before configuring interface-based traffic policing, familiarize yourself with the applicable environment, complete the pre-configuration tasks, and obtain the data required for the configuration. This will help you complete the configuration task quickly and accurately.

Applicable Environment

If the service traffic sent by users is not limited, a network is congested because a large number of users send bursts of data in the same period. To make full use of limited network resources and provide better services for more users, limit user service traffic.

If traffic on the management interface is heavy because of malicious attacks or network exceptions, the CPU usage becomes high and services are interrupted. To ensure that the system works properly, limit the traffic on the management interface.

Interface-based traffic policing takes effect for all the incoming service traffic.

Pre-configuration Tasks

Before configuring interface-based traffic policing, complete the following tasks:

- Setting physical parameters of interfaces
- Setting link layer attributes of interfaces to ensure that these interfaces work properly
- Assigning IP addresses to the interfaces and configuring routing protocols to ensure that routes are reachable

Data Preparation

To configure interface-based traffic policing, you need the following data.

No.	Data
1	CIR, PIR, CBS, and PBS values on a common interface
2	Rate limit on a management interface

2.2.2 Limiting the Rate of Traffic on the Outbound Interface

To limit the rate of the traffic entering an interface of the S7700, apply a CAR profile to the interface.

Context

NOTE

The following operations are performed on a common interface but not on the management interface.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
qos car car-name cir cir-value [ cbs cbs-value [ pbs pbs-value ] | pir pir-value [ cbs cbs-value pbs pbs-value ] ]
```

A CAR profile is created.

By default, no CAR profile is created.

Step 3 Run:

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

The interface view is displayed.

Or, run:

```
port-group port-group-name
```

The port group view is displayed.

NOTE

You can configure interface-based traffic policing on Ethernet, GE, XGE, and Eth-Trunk interfaces.

To configure the same CAR profile on multiple interfaces, configure the CAR profile on the port group to reduce the workload.

Create a port group before performing this task. For details on how to create a port group, see Configuring the Interface Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

Step 4 Run:

```
qos car inbound car-name
```

The CAR profile is applied on an interface.

After a CAR profile is applied on an interface, the S7700 implements traffic policing for all the service traffic received on the interface.

----End

2.2.3 Configuring the Rate Limit on the Management Interface

Traffic policing on the management interface limits the traffic received from the management interface to improve system performance.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
interface Ethernet 0/0/0
```

The management interface view is displayed.

Step 3 Run:

```
qos lr pps packets
```

The rate limit is set.

NOTE

The rate limit of traffic on the management interface cannot be less than 100; otherwise, FTP and Telnet functions may fail to work.

----End

2.2.4 Checking the Configuration

After interface-based traffic policing is configured, you can view the name, index, and parameters of the CAR profiles and the number of times the CAR profile was applied.

Prerequisites

The configurations of interface-based traffic policing are complete.

Procedure

- Run the **display qos car** { **all** | **name** *car-name* } command to check the CAR profile configuration.
- Run the **display qos configuration interface** [*interface-type interface-number*] command to check all the QoS configurations on an interface.

----End

2.3 Configuring Traffic Policing Based on a Traffic Classifier

After traffic policing based on a traffic classifier is configured, the S7700 policies the traffic matching traffic classification rules.

2.3.1 Establishing the Configuration Task

Before configuring traffic policing based on a traffic classifier, familiarize yourself with the applicable environment, complete the pre-configuration tasks, and obtain the data required for the configuration. This will help you complete the configuration task quickly and accurately.

Applicable Environment

If the service traffic sent by users is not limited, a network is congested because a large number of users send bursts of data in the same period. To make full use of limited network resources and provide better services for more users, limit user service traffic.

Traffic policing based on a traffic classifier can be used to control the service traffic of a certain type.

Pre-configuration Tasks

Before configuring traffic policing based on a traffic classifier, complete the following tasks:

- Setting physical parameters of interfaces
- Setting link layer attributes of interfaces to ensure that these interfaces work properly
- Assigning IP addresses to the interfaces and configuring routing protocols to ensure that routes are reachable

Data Preparation

To configure traffic policing based on a traffic classifier, you need the following data.

No.	Data
1	Name of the traffic classifier and related parameters
2	Name of the traffic behavior and CAR parameters: CIR, (optional) CBS, (optional) PIR, (optional) PBS, (optional) color, (optional) coloring mode, and (optional) CoS
3	Name of the traffic policy, and object and inbound or outbound direction to which traffic policing based on a traffic classifier is applied

2.3.2 Configuring Complex Traffic Classification

The S7700 can classify traffic according to the ACL, Layer 2 information in packets, and Layer 3 information in packets.

Select proper traffic classification rules and configure complex traffic classification as required. For details, see **1.4.2 Configuring Complex Traffic Classification**.

2.3.3 Configuring a Traffic Policing Action on the

You can configure traffic policing actions, set the CIR, PIR, CBS, and PBS values, and configure actions for packets with different PHBs and colors.

Context

Level-2 CAR is supported by the S7700. After the system applies the CAR to the service flows matching a traffic classifier in a traffic policy, it aggregates all the service flows matching the traffic classifier bound to the aggregate CAR action in the same traffic policy and applies the CAR to the flows. This is also called hierarchical traffic policing.

Hierarchical traffic policing implements traffic statistics multiplexing and service control. For example, hierarchical traffic policing limits the services of level-1 and level-2 users. It also limits the traffic of level-1 user groups and level-2 groups user.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
traffic behavior behavior-name
```

A traffic behavior is created and the traffic behavior view is displayed.

Step 3 Run:

```
car cir cir-value [ pir pir-value ] [ cbs cbs-value pbs pbs-value ] [ share ]
[ mode { color-blind | color-aware } ] [ green { discard | pass [ service-class
class color color ] } ] [ yellow { discard | pass [ service-class class color
color ] } ] [ red { discard | pass [ service-class class color color ] } ] *
```

A CAR action is configured.

Apart from being defined in a DiffServ domain, packet colors can also be defined in traffic policing, as follows:

- When the burst size of a packet is less than the CBS value, the packet is colored green.
- When the burst size of a packet is greater than or equal to the CBS value but smaller than the PBS value, the packet is colored yellow.
- When the burst size of a packet is greater than or equal to the PBS value, the packet is colored red.

When you bind a DiffServ domain to an inbound interface and configure traffic policing in which the CBS and PBS values are set:

- If the coloring mode of traffic policing is set to color-blind, packet coloring depends on the coloring of a packet defined in traffic policing, regardless of packet coloring defined in a DiffServ domain.
- If the coloring mode of traffic policing is set to color-aware, packet coloring complies with the following rules when the coloring rules of a packet defined in traffic policing and in a DiffServ domain conflict:

Table 2-1 Rules for making the color of a packet effective

Color of a Packet Defined in a DiffServ Domain	Color of a Packet Defined in Traffic Policing	Final Color of a Packet
Green	Yellow	Yellow
Green	Red	Red
Yellow	Green	Yellow
Yellow	Red	Red
Red	Green	Red
Red	Yellow	Red

Step 4 Run:

```
quit
```

Exit from the traffic behavior view.

Step 5 (Optional) Conduct the following steps to configure aggregate CAR.

1. Run:

   ```
   system-view
   ```

   The system view is displayed.

2. Run:

   ```
   qos car car-name cir cir-value [ cbs cbs-value [ pbs pbs-value ] | pir pir-value [ cbs cbs-value pbs pbs-value ] ]
   ```

   A CAR profile is created.

   NOTE

   Aggregate CAR supports only the single token bucket.

3. Run:

   ```
   traffic behavior behavior-name
   ```

   A traffic behavior is created and the traffic behavior view is displayed.

4. Run:

   ```
   car car-name share
   ```

   An aggregate CAR action is configured.

   NOTE

   S series boards do not support share CAR.

   Aggregate CAR is valid for only incoming packets.

   After aggregate CAR is configured, the rules in a traffic classifier bound to a traffic behavior share a CAR index. The system aggregates the traffic and implements the CAR for the traffic. If the traffic classifier contains both Layer 2 information-based rules and Layer 3 information-based rules, the **car share** command does not take effect.

   If traffic policing is configured on an interface, the rate limit is only applied to the interface. If traffic policing is configured globally, the rate limit is applied to all interfaces.

----End

2.3.4 Creating a Traffic Policy

You can associate a traffic classifier with a traffic behavior in a traffic policy.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
traffic policy policy-name [ match-order { auto | config } ]
```

A traffic policy is created and the traffic policy view is displayed.

After a traffic policy is applied, you cannot use the **traffic policy** command to modify the matching order of traffic classifiers in the traffic policy. To modify the matching order, delete the traffic policy, and re-create a traffic policy and specify the matching order.

Step 3 Run:

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

A traffic classifier is bound to a traffic behavior in the traffic policy.

----End

2.3.5 Applying the Traffic Policy

The configured traffic policy takes effect only after being applied to the system or a slot, an interface, or a VLAN.

Procedure

- Applying a traffic policy to the system or a slot
  1. Run:

     ```
     system-view
     ```

     The system view is displayed.
  2. Run:

     ```
     traffic-policy policy-name global { inbound | outbound } [ slot slot-id ]
     ```

     A traffic policy is applied to the system or a slot in the inbound or outbound direction.

     Only one traffic policy can be applied to the system in the inbound or outbound direction.

     Only one traffic policy can be applied to a slot in the inbound or outbound direction. A traffic policy cannot be applied to the system and a slot simultaneously.

     - After a traffic policy is applied to the system, the system performs traffic policing for all the incoming or outgoing packets that match traffic classification rules.
     - After a traffic policy is applied to a slot, the system performs traffic policing for all the incoming or outgoing packets that match traffic classification rules on the slot.
- Applying a traffic policy to an interface

1. Run:
   system-view

   The system view is displayed.
2. Run:
   interface *interface-type interface-number*

   The interface view is displayed.
3. Run:
   traffic-policy *policy-name* { **inbound** | **outbound** }

   A traffic policy is applied to the interface in the inbound or outbound direction.

   Only one traffic policy can be applied to an interface in the inbound or outbound direction.

   After a traffic policy is applied, the system performs traffic policing for the packets that pass through this interface and match a traffic classifier in the inbound or outbound direction.

   On a sub-interface, a traffic policy can be applied only to the inbound direction.

- Applying a traffic policy to a VLAN
  1. Run:
     system-view

     The system view is displayed.
  2. Run:
     vlan *vlan-id*

     The VLAN view is displayed.
  3. Run:
     traffic-policy *policy-name* { **inbound** | **outbound** }

     A traffic policy is applied to the VLAN in the inbound or outbound direction.

     Only one traffic policy can be applied to a VLAN in the inbound or outbound direction.

     After a traffic policy is applied, the system performs traffic policing for the packets that belong to a VLAN and match a traffic classifier in the inbound or outbound direction.

----End

2.3.6 Checking the Configuration

After traffic policing based on a traffic classifier is configured, you can view the traffic statistics or CAR statistics.

Context

The configurations of traffic policing based on a traffic classifier are complete.

Procedure

- Run the **display traffic behavior user-defined** [*behavior-name*] command to check the traffic behavior configuration.

- Run the **display traffic classifier user-defined** [*classifier-name*] command to check the traffic classifier configuration.
- Run the **display traffic policy user-defined** [*policy-name* [**classifier** *classifier-name*]] command to check the traffic policy configuration.
- Run the **display qos car** { **all** | **name** *car-name* } command to check the CAR profile configuration.
- Run the **display qos configuration interface** [*interface-type interface-number*] command to check all the QoS configurations on the interface.

----End

2.4 Configuring Traffic Shaping

After traffic shaping is configured, the S7700 shapes packets matching traffic classification rules so that packets are sent out at an even rate.

2.4.1 Establishing the Configuration Task

Before configuring traffic shaping, familiarize yourself with the applicable environment, complete the pre-configuration tasks, and obtain the required data. This helps you complete the configuration task quickly and accurately.

Applicable Environment

If the bandwidth of upstream and downstream networks is different, you can configure traffic shaping on the outgoing interface connecting the upstream network and downstream network. In this manner, the rate of packets sent to the downstream network meets the requirements of the bandwidth of the downstream network. This can prevent congestion and packet loss on the network to a certain degree.

The S7700 supports traffic shaping on an interface and in an interface queue. You can configure traffic shaping as required. If traffic shaping of these two types is configured, ensure that the CIR for traffic shaping on an interface is greater than or equal to the sum of CIRs for traffic shaping in an interface queue. Otherwise, traffic shaping fails. For example, traffic of lower priorities preempts the bandwidth of traffic of higher priorities.

Pre-configuration Tasks

Before configuring traffic shaping, complete the following tasks:

- Setting physical parameters of interfaces
- Setting link layer attributes of interfaces to ensure normal operation of the interfaces
- Assigning IP addresses to the interfaces and configuring routing protocols to ensure that routes are reachable

Data Preparation

To configure traffic shaping, you need the following data.

No.	Data
1	Rate for traffic shaping on an interface
2	(Optional) Rate for traffic shaping in an interface queue, including the CIR and PIR
3	Interface on which traffic shaping is applied or index of the queue

2.4.2 Configuring Traffic Shaping on an Interface

You can configure traffic shaping on an interface to limit the rate of data sent by the interface.

Context

Use this procedure to perform traffic shaping for all the downstream packets on an interface.

If you need to set the same traffic shaping rate on multiple interfaces, you can perform the configuration on the port group to reduce the workload.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Or run the **port-group** *port-group-name* command to display the port group view.

NOTE

Create a port group before performing this task. For details on how to create a port group, see Configuring the Interface Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

Step 3 Run:

```
qos lr cir cir-value [ cbs cbs-value ] [ outbound ]
```

The rate for traffic shaping on an interface is set.

By default, the CIR for traffic shaping on an interface is the maximum bandwidth of the interface. For example, the CIR for traffic shaping on an Ethernet interface is 100000 kbit/s; the CIR for traffic shaping on a GE interface is 1000000 kbit/s ; the CIR for traffic shaping on a 10GE interface is 10000000 kbit/s.

NOTE

- If this command is run repeatedly on the same interface, the latest configuration overrides the previous configuration.
- On an MPLS Traffic Engineering (TE) tunnel, the CIR for traffic shaping set through the **qos lr** command must be greater than the maximum reservable bandwidth of the MPLS TE tunnel if the maximum link bandwidth and maximum reservable bandwidth of the MPLS TE tunnel are set; otherwise, the maximum reservable bandwidth of the MPLS TE tunnel cannot be provided.
- If traffic shaping in an interface queue is configured on the same interface, the CIR for traffic shaping on an interface must be greater than or equal to the sum of CIRs for traffic shaping in an interface queue. Otherwise, traffic shaping fails. For example, traffic of lower priorities preempts the bandwidth of traffic of higher priorities.

----End

2.4.3 (Optional) Setting the Length of the Interface Queue

This section describes how to set the length of the interface queue and the length of the specified priority queue.

Context

The length of the interface priority queue is automatically managed by the system. You can set the length of the interface priority queue as required by using the **qos queue** command.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
qos queue queue-index length length-value
```

The length of the interface priority queue is set.

NOTE

The queue length on the G48SBC, G48TBC or LE2D2X08SED0 boards cannot be changed. The two boards support the maximum queue length by default.

----End

2.4.4 Configuring Traffic Shaping in an Interface Queue

This section describes how to configure traffic shaping, enable traffic shaping in an interface queue, and set traffic shaping parameters.

Context

Use this procedure to perform traffic shaping for packets of a certain type of services on an interface.

Before configuring traffic shaping in an interface queue, map priorities of packets to PHBs based on simple traffic classification or re-mark the internal priorities based on complex traffic classification. Different services can enter different interface queues.

To set the same queue shaping rate on multiple interfaces, perform the configuration on the port group to reduce the workload.

NOTE

For details about priority mapping based on simple traffic classification, see **Configuring Priority Mapping Based on Simple Traffic Classification**

For details about internal priority re-marking based on complex traffic classification, see **Creating a Traffic Policy Based on Complex Traffic Classification**.

Traffic shaping cannot be configured by using a port group on W series boards.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Or, run:

```
port-group port-group-name
```

The port group view is displayed.

NOTE

Create a port group before performing this task. For details on how to create a port group, see Configuring the Interface Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

Step 3 Run:

```
qos queue queue-index shaping cir cir-value pir pir-value [ cbs cbs-value pbs pbs-value ]
```

The rate for traffic shaping in an interface queue is set.

By default, the rate for traffic shaping in an interface queue is the maximum bandwidth of the interface.

----End

2.4.5 Checking the Configuration

After traffic shaping is configured, you can view the rate limit on an interface or in an interface queue.

Procedure

- Run the **display qos queue statistics interface** *interface-type interface-number* command to check the rate limit of the interface queue.

NOTE

S-series boards do not support the **display qos queue statistics** command. For details about boards, see Board Classification in the *S7700 Smart Routing Switch - Hardware Description*.

- Run the **display qos configuration interface** *interface-type interface-number* command to check all the QoS configurations on the interface.

----End

2.5 Maintaining Traffic Policing and Traffic Shaping

This section describes how to maintain traffic policing and traffic shaping.

2.5.1 Displaying the Traffic Statistics

If the traffic statistics action is configured, you can run display commands to view the traffic statistics.

Context

To view the flow-based traffic statistics, a traffic policy must exist and contain the traffic statistics action.

Procedure

- Run the **display traffic policy statistics** { **global** [**slot** *slot-id*] | **interface** *interface-type interface-number* | **vlan** *vlan-id* } { **inbound** | **outbound** } [**verbose** { **classifier-base** | **rule-base** } [**class** *classifier-name*]] command to check the flow-based traffic statistics.
- Run the **display qos car statistics interface** *interface-type interface-number* **inbound** command to check the statistics on forwarded and discarded packets on a specified interface configured with interface-based traffic policing.
- Run the **display qos queue statistics interface** *interface-type interface-number* command to check the queue-based traffic statistics on the interface.

  NOTE

  - The **display qos queue statistics** command cannot be used on S-series and WAN boards.
  - The statistics on bytes cannot be collected on S-series and WAN boards.
  - For details about S-series and WAN boards, see Board Classification in the *S7700 Smart Routing Switch - Hardware Description*.

----End

2.5.2 Checking the Usage of the Queue

You can use display commands to view the Usage of the Queue.

Context

To obtain the usage of queues, you can run the following command in any view.

Procedure

- Run the **display qos queue length interface** *interface-type interface-number* command to view the usage of priority queues on the interface.

  The command output on different types of boards is different:

  - S-series boards

    The system allocates the minimum buffer length to each queue on the interface. When a queue uses up its buffer, it obtains the shared buffer on the interface. By default, the used length of a queue is the sum of the buffer length of the queue (N) and the shared buffer length of the interface (M).

    After the queue length is set:

    - If the queue length is smaller than M, the used queue length is M.
    - If the queue length is larger than M, the used queue length is the configured value that is converted by the system. For example, when the queue length is set to 61442, the displayed value is 61568.

  - E-series boards

    By default, the used queue length is the number of bytes of packets buffered in the queue.

    After the queue length is set, the used queue length is the configured value that is converted by the system. For example, when the queue length is set to 61442, the displayed value is 61568.

  NOTE

  G48SBC boards, G48TBC boards and LE2D2X08SED0 boards do not support the **display qos queue length** command.

  For details about boards, see Board Classification in the *S7700 Smart Routing Switch Hardware Description*.

----End

2.5.3 Clearing the Traffic Statistics

You can use the reset commands to clear the traffic statistics.

Context

CAUTION

The traffic policing statistics cannot be restored after being cleared. Exercise caution when you run the command.

Procedure

- Run the **reset traffic policy statistics** { **global** | **interface** *interface-type interface-number* | **vlan** *vlan-id* } { **inbound** | **outbound** } command to clear the flow-based traffic statistics.

- Run the **reset qos car statistics interface** *interface-type interface-number* **inbound** command to clear the statistics on forwarded and discarded packets on a specified interface configured with interface-based traffic policing.
- Run the **reset qos queue statistics interface** *interface-type interface-number* command to clear the queue-based traffic statistics on the interface.

  NOTE

  The **reset qos queue statistics** command cannot be used on S-series boards.

  For details about boards, see Board Classification in the *S7700 Smart Routing Switch - Hardware Description*.

----End

2.6 Configuration Examples

This section provides several configuration examples of traffic policing and traffic shaping.

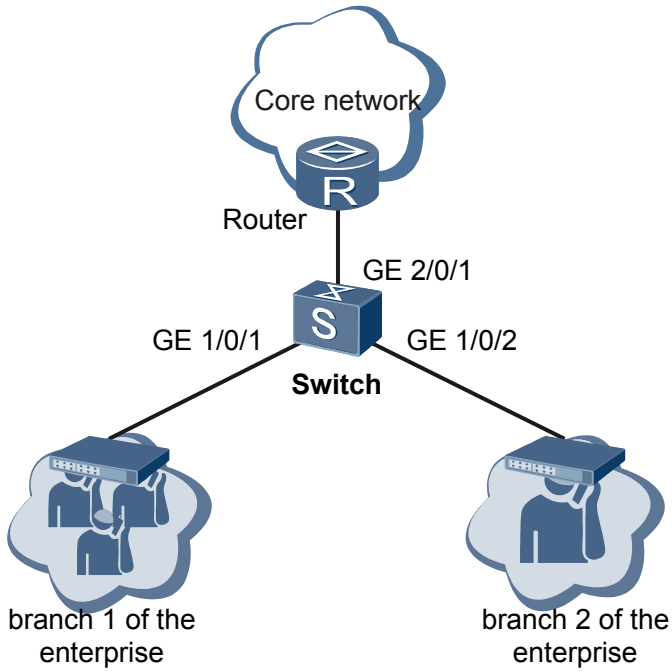
2.6.1 Example for Configuring Interface-based Traffic Policing

You can configure interface-based traffic policing so that the Switch can provide different bandwidth services for users.

Networking Requirements

As shown in **Figure 2-3**, the Switch is connected to the router by using GE 2/0/1; Branch 1 and Branch 2 of the enterprise access the Switch by using GE 1/0/1 and GE 1/0/2 and access the network by using the Switch and router. It is required that the fixed bandwidth of Branch 1 be 8 Mbit/s and the maximum bandwidth be 10 Mbit/s, and the fixed bandwidth of Branch 2 be 5 Mbit/s and the maximum bandwidth be 8 Mbit/s.

Figure 2-3 Networking diagram of interface-based traffic policing



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure each interface of the Switch so that users can access the network.
2. Create CAR profiles and set the CIR and PIR values.
3. Apply CAR profiles on GE 1/0/1 and GE 1/0/2 of the Switch in the inbound direction.

Data Preparation

To complete the configuration, you need the following data:

- IP address of the upstream interface on the Switch: 192.168.1.1/24
- VLAN that Branch 1 belong to: VLAN 100
- VLAN that Branch 2 belong to: VLAN 200
- CIR and PIR values for Branch 1: 8000 kbit/s and 10000 kbit/s
- CIR and PIR values for Branch 2: 5000 kbit/s and 8000 kbit/s

Procedure

Step 1 Create VLANs and configure each interface of the Switch.

Create VLANs 100, 200, and 300, and add GE 1/0/1, GE 1/0/2, and GE 1/0/1 to VLANs 100, 200, and 300.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 100 200 300
```

Configure the type of GE 1/0/1, GE 1/0/2, and GE 2/0/1 as trunk and permit packets from VLANs 100, 200, and 300 to pass through.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] port link-type trunk
[Switch-GigabitEthernet1/0/1] port trunk allow-pass vlan 100
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] port link-type trunk
[Switch-GigabitEthernet1/0/2] port trunk allow-pass vlan 200
[Switch-GigabitEthernet1/0/2] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 100 200 300
[Switch-GigabitEthernet2/0/1] quit
```

Create VLANIF 300 and assign the network segment address 192.168.1.1/24 to VLANIF 300.

```
[Switch] interface vlanif 300
[Switch-Vlanif300] ip address 192.168.1.1 24
```

NOTE

Assign IP address 192.168.1.2/24 to the interface connecting the router and the Switch.

Step 2 Configure CAR profiles.

Create CAR profiles **qoscar1** and **qoscar2** on the Switch to limit the traffic of enterprise and individual users.

```
[Switch] qos car qoscar1 cir 8000 pir 10000
[Switch] qos  car qoscar2 cir 5000 pir 8000
```

Step 3 Apply CAR profiles.

Apply CAR profiles **qoscar1** and **qoscar2** in the inbound direction of GE 1/0/1 and GE 1/0/2 on the Switch to limit incoming traffic of enterprise and individual users.

```
[Switch] interface gigabitEthernet1/0/1
[Switch-GigabitEthernet1/0/1] qos car inbound qoscar1
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] qos car inbound qoscar2
[Switch-GigabitEthernet1/0/2] quit
[Switch] quit
```

Step 4 Verify the configuration.

Check the configuration of CAR profiles.

```
<Switch> display qos car all
  ------------------------------------------------------------------
   CAR Name     : qoscar1
   CAR Index    : 0
    car cir 8000 (Kbps) pir 10000 (Kbps) cbs 1000000 (byte) pbs 1250000 (byte)
  ------------------------------------------------------------------
   CAR Name     : qoscar2
   CAR Index    : 1
    car cir 5000 (Kbps) pir 8000 (Kbps) cbs 625000 (byte) pbs 1000000 (byte)
```

Send traffic to GE 1/0/1 and GE 1/0/2 at the rate of 6000 kbit/s, 9000 kbit/s, and 11000 kbit/s, and run the **display qos car statistics** command to check the QoS CAR statistics. If the configuration is successful, you can obtain the following results:

- When packets are sent to GE 1/0/1 and GE 1/0/2 at the rate of 6000 kbit/s, all the packets are forwarded.
- When packets are sent to GE 1/0/1 and GE 1/0/2 at the rate of 9000 kbit/s, all the packets on GE 1/0/1 are forwarded but certain packets on GE 1/0/2 are discarded.
- When packets are sent to GE 1/0/1 and GE 1/0/2 at the rate of 11000 kbit/s, certain packets on GE 1/0/1 and GE 1/0/2 are discarded.

Run the **display qos car statistics** command on GE 1/0/1 in the inbound direction, and the following information is displayed:

```
<Switch> display qos car statistics interface gigabitethernet 1/0/1 inbound
 Board : 1
 Passed packets:                          0
 Passed bytes:                            -
 Discard packets:                         0
 Discard bytes:                           -
 Board : 10
 Passed packets:                          0
 Passed bytes:                            0
 Discard packets:                         0
 Discard bytes:                           0
 Board : 11
 Passed packets:                          0
 Passed bytes:                            -
 Discard packets:                         0
 Discard bytes:                           -
 Discard bytes:                           1048
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 100 200 300
#
 qos car qoscar1 cir 8000 pir 10000 cbs 1000000 pbs 1250000
 qos car qoscar2 cir 5000 pir 8000 cbs 625000 pbs 1000000
#
interface Vlanif300
 ip address 192.168.1.1 255.255.255.0
#
interface GigabitEthernet1/0/1
 port link-type trunk
 port trunk allow-pass vlan 100
 qos car inbound qoscar1
#
interface GigabitEthernet1/0/2
 port link-type trunk
 port trunk allow-pass vlan 200
 qos car inbound qoscar2
#
interface GigabitEthernet2/0/1
 port link-type trunk
 port trunk allow-pass vlan 100 200 300
#
return
```

2.6.2 Example for Configuring Traffic Policing Based on a Traffic Classifier

The Switch provides different bandwidth by configuring traffic policing based on a traffic classifier and setting different CAR parameters.

Networking Requirements

The Switch is connected to the router by using GE 2/0/1; enterprise users can access the network by using the Switch and the router. In **Table 2-2**:

- Voice services belong to VLAN 120.
- Video services belong to VLAN 110.
- Data services belong to VLAN 100.

On the Switch, traffic policing needs to be performed on packets of different services to limit traffic within a proper range and ensure bandwidth of each service.

DSCP priorities carried in service packets sent from the user side cannot be trusted and services require different QoS in practice. Therefore, you need to re-mark DSCP priorities of different service packets on the Switch so that the downstream router can process packets based on priorities.

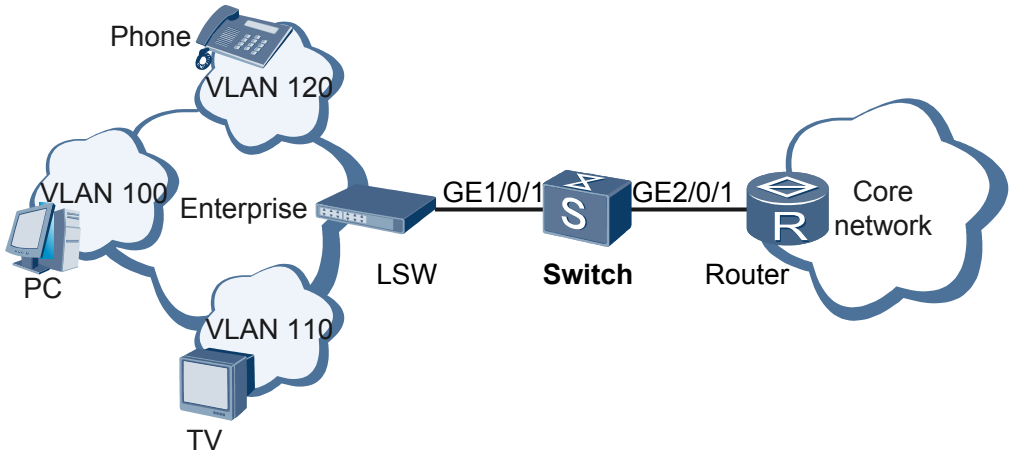
The requirements are as follows:

Table 2-2 QoS provided by the Switch for upstream traffic

Traffic Type	CIR (Mbit/s)	PIR (Mbit/s)	DSCP Priority
Voice	2	10	46

Traffic Type	CIR (Mbit/s)	PIR (Mbit/s)	DSCP Priority
Video	4	10	30
Data	4	10	14

Figure 2-4 Network diagram for configuring traffic policing based on a traffic classifier



Configuration Roadmap

The configuration roadmap is as follows:

1. Create VLANs and configure interfaces so that enterprise can access the network by using the Switch.
2. Create traffic classifiers based on the VLAN ID on the Switch.
3. Create traffic behaviors on the Switch to limit the traffic received from the enterprise and re-mark DSCP priorities of packets.
4. Create a traffic policy on the Switch, bind traffic behaviors to traffic classifiers in the traffic policy, and apply the traffic policy to the interface between the enterprise and the Switch.

Data Preparation

To complete the configuration, you need the following data:

- Names of traffic classifiers matching service flows
- Re-marked priorities of packets with different VLAN IDs
- Parameters for packets with different VLAN IDs: CIR and PIR values
- Type and number of the interface to which a traffic policy needs to be applied

Procedure

Step 1 Create VLANs and configure interfaces.

Create VLAN 100, VLAN 110, and VLAN 120 on the Switch.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 100 110 120
```

Configure the access types of GE 1/0/1 and GE2/0/1 to trunk, add GE 1/0/1 and GE2/0/1 to VLAN 100, VLAN 110, and VLAN 120.

```
[Switch] interface gigabitethernet 1/0/1
[Switch-GigabitEthernet1/0/1] port link-type trunk
[Switch-GigabitEthernet1/0/1] port trunk allow-pass vlan 100 110 120
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet 2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 100 110 120
[Switch-GigabitEthernet2/0/1] quit
```

Step 2 Create traffic classifiers.

Create traffic classifiers **c1** to **c3** on the Switch to match different service flows from the enterprise based on VLAN IDs.

```
[Switch] traffic classifier c1 operator and
[Switch-classifier-c1] if-match vlan-id 120
[Switch-classifier-c1] quit
[Switch] traffic classifier c2 operator and
[Switch-classifier-c2] if-match vlan-id 110
[Switch-classifier-c2] quit
[Switch] traffic classifier c3 operator and
[Switch-classifier-c3] if-match vlan-id 100
[Switch-classifier-c3] quit
```

Step 3 Create traffic behaviors.

Create traffic behaviors **b1** to **b3** on the Switch to limit different service flows and re-mark priorities.

```
[Switch] traffic behavior b1
[Switch-behavior-b1] car cir 2000 pir 10000 green pass
[Switch-behavior-b1] remark dscp 46
[Switch-behavior-b1] statistic enable
[Switch-behavior-b1] quit
[Switch] traffic behavior b2
[Switch-behavior-b2] car cir 4000 pir 10000 green pass
[Switch-behavior-b2] remark dscp 30
[Switch-behavior-b2] statistic enable
[Switch-behavior-b2] quit
[Switch] traffic behavior b3
[Switch-behavior-b3] car cir 4000 pir 10000 green pass
[Switch-behavior-b3] remark dscp 14
[Switch-behavior-b3] statistic enable
[Switch-behavior-b3] quit
```

Step 4 Create a traffic policy and apply it on the interface.

Create traffic policy **p1** on the Switch, bind traffic classifiers to traffic behaviors in the traffic policy, and apply the traffic policy to GE1/0/1 in the inbound direction to limit the packets received from the user side and re-mark priorities of these packets.

```
[Switch] traffic policy p1
[Switch-trafficpolicy-p1] classifier c1 behavior b1
[Switch-trafficpolicy-p1] classifier c2 behavior b2
[Switch-trafficpolicy-p1] classifier c3 behavior b3
[Switch-trafficpolicy-p1] quit
[Switch] interface gigabitethernet 1/0/1
[Switch-GigabitEthernet1/0/1] traffic-policy p1 inbound
[Switch-GigabitEthernet1/0/1] quit
```

Step 5 Verify the configuration.

Check the configuration of the traffic classifier.

```
[Switch] display traffic classifier user-defined
  User Defined Classifier Information:
   Classifier: c2
    Precedence: 10
    Operator: AND
    Rule(s) : if-match vlan-id 110

   Classifier: c3
    Precedence: 15
    Operator: AND
    Rule(s) : if-match vlan-id 100

   Classifier: c1
    Precedence: 5
    Operator: AND
    Rule(s) : if-match vlan-id 120

Total classifier number is 3
```

Check the configuration of the traffic policy. Here, the configuration of the traffic policy **p1** is displayed.

```
[Switch] display traffic policy user-defined p1
  User Defined Traffic Policy Information:
  Policy: p1
   Classifier: c1
    Operator: AND
     Behavior: b1
      Committed Access Rate:
        CIR 2000 (Kbps), PIR 10000 (Kbps), CBS 250000 (byte), PBS 1250000 (byte)
        Color Mode: color Blind
        Conform Action: pass
        Yellow  Action: pass
        Exceed  Action: discard
      Marking:
        Remark DSCP ef
      Statistic: enable
   Classifier: c2
    Operator: AND
     Behavior: b2
      Marking:
        Remark DSCP af33
      Statistic: enable
      Committed Access Rate:
        CIR 4000 (Kbps), PIR 10000 (Kbps), CBS 500000 (byte), PBS 1250000 (byte)
        Color Mode: color Blind
        Conform Action: pass
        Yellow  Action: pass
        Exceed  Action: discard
       Marking:
        Remark DSCP af33
      Statistic: enable
   Classifier: c3
    Operator: AND
     Behavior: b3
      Committed Access Rate:
        CIR 4000 (Kbps), PIR 10000 (Kbps), CBS 500000 (byte), PBS 1250000 (byte)
        Color Mode: color Blind
        Conform Action: pass
        Yellow  Action: pass
        Exceed  Action: discard
      Marking:
        Remark DSCP af13
      Statistic: enable
```

Check the configuration of the traffic policy applied on an interface. Here, the configuration of the traffic policy applied to GE1/0/1 is displayed.

```
[Switch] display traffic policy statistics interface gigabitethernet 1/0/1 inbound

 Interface: GigabitEthernet1/0/1
 Traffic policy inbound: p1
 Rule number: 3
 Current status: OK!
-------------------------------------------------------------------------
 Board : 1
Item                                Packets                       Bytes
-------------------------------------------------------------------------
Matched                                  10                           10000
  +--Passed                              8                            8000
  +--Dropped                             2                            2000
    +--Filter                            2                            2000
    +--URPF                              -                            -
    +--CAR                               2                            2000
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 100 110 120
#
traffic classifier c1 operator and precedence 5
 if-match vlan-id 120
traffic classifier c2 operator and precedence 10
 if-match vlan-id 110
traffic classifier c3 operator and precedence 15
 if-match vlan-id 100
#
traffic behavior b2
 car cir 4000 pir 10000 cbs 500000 pbs 1250000 mode color-blind green pass
yellow pass red discard
 remark dscp af33
 statistic enable
traffic behavior b3
 car cir 4000 pir 10000 cbs 500000 pbs 1250000 mode color-blind green pass
yellow pass red discard
 remark dscp af13
 statistic enable
traffic behavior b1
 car cir 2000 pir 10000 cbs 250000 pbs 1250000 mode color-blind green pass
yellow pass red discard
 remark dscp ef
 statistic enable
#
traffic policy p1
 classifier c1 behavior b1
 classifier c2 behavior b2
 classifier c3 behavior b3
#
interface GigabitEthernet1/0/1
 port link-type trunk
 port trunk allow-pass vlan 100 110 120
 traffic-policy p1 inbound
#
interface GigabitEthernet2/0/1
 port link-type trunk
 port trunk allow-pass vlan 100 110 120
#
return
```

2.6.3 Example for Configuring Hierarchical Traffic Policing

Hierarchical traffic policing sets different CAR parameters for different users and services to provide differentiated broadband services.

Networking Requirements

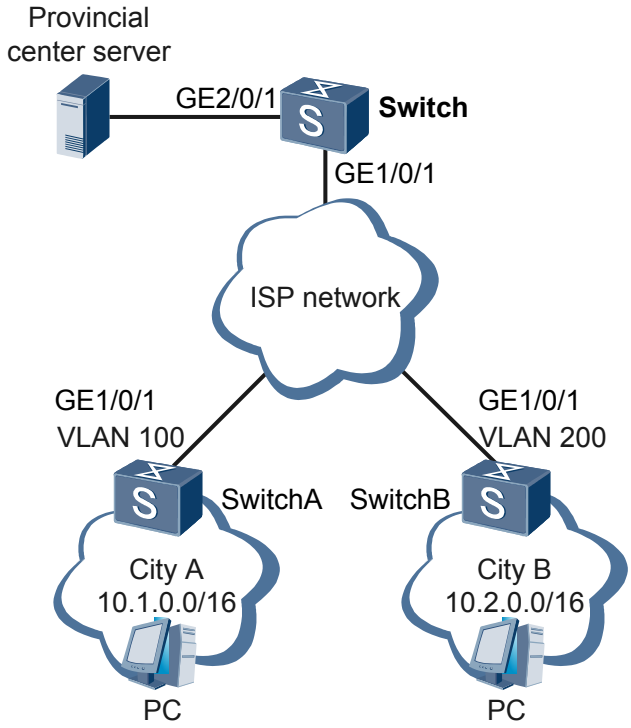
As shown in **Figure 2-5**, data exchange between the provincial center and city A, and between the provincial center and city B is implemented by leasing the carrier network. The carrier allocates 2 Mbit/s bandwidth for each city. The provincial center is connected to the carrier network through the Switch and traffic needs to be controlled on the Switch to ensure that:

- The rate limit of the traffic sent from the provincial center to each city is 2 Mbit/s.
- The Switch processes voice, video, and data services based on priorities. It sends traffic with higher priorities first and allocates certain bandwidth to traffic with lower priorities.
- With bandwidth guarantee, bandwidth is allocated randomly.

Table 2-3 Downstream traffic control on the Switch

City	EF Traffic	AF31 Traffic	AF11 Traffic	BE Traffic
City A	700 kbit/s	400 kbit/s	500 kbit/s	200 kbit/s
City B	800 kbit/s	500 kbit/s	300 kbit/s	100 kbit/s

Figure 2-5 Hierarchical traffic policing



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure CAR profiles to limit the traffic sent to city A and city B within 2 Mbit/s.
2. Configure ACLs to permit the traffic sent to city A and city B to pass through.
3. Configure traffic classifiers to match traffic priorities and the ACLs.
4. Configure traffic behaviors to allocate ensured bandwidth to each type of traffic and limit the total traffic.
5. Configure a traffic policy, bind the configured traffic behaviors and traffic classifiers to the traffic policy, and apply the traffic policy to the interface connecting the provincial center server and the Switch.

Data Preparation

To complete the configuration, you need the following data:

- Total bandwidth of the traffic sent to city A and city B and CAR profile names
- Numbers of ACLs matching the traffic sent to city A and city B and network segment IP addresses
- Priorities of traffic
- CIR values
- Traffic policy name and type and number of the interface to which the traffic policy is applied

Procedure

Step 1 Configure CAR profiles.

Create and configure a CAR profile to limit the traffic sent to city A within 2 Mbit/s.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] qos car city_a cir 2000
```

Create and configure a CAR profile to limit the traffic sent to city B within 2 Mbit/s.

```
[Switch] qos car city_b cir 2000
```

Step 2 Configure ACLs.

Configure ACL 3000 to permit the TCP packets destined for city A on the network segment 10.1.0.0/16 to pass through.

```
[Switch] acl 3000
[Switch-acl-adv-3000] rule 5 permit tcp destination 10.1.0.0 0.0.255.255
```

Configure ACL 3001 to permit the TCP packets destined for city B on the network segment 10.2.0.0/16 to pass through.

```
[Switch] acl 3001
[Switch-acl-adv-3001] rule 5 permit tcp destination 10.2.0.0 0.0.255.255
```

Step 3 Configure traffic classifiers.

Create traffic classifiers **city_a_ef**, **city_a_af31**, **city_a_af11**, and **city_a_be** for traffic sent to city A. These traffic classifiers match traffic with DSCP priorities EF, AF31, AF11, and 0 and ACL 3000. Create a traffic classifier **city_a_default** for other traffic to match ACL 3000.

```
[Switch] traffic classifier city_a_ef operator and
[Switch-classifier-city_a_ef] if-match dscp ef
[Switch-classifier-city_a_ef] if-match acl 3000
[Switch-classifier-city_a_ef] quit
[Switch] traffic classifier city_a_af31 operator and
[Switch-classifier-city_a_af31] if-match dscp af31
[Switch-classifier-city_a_af31] if-match acl 3000
[Switch-classifier-city_a_af31] quit
[Switch] traffic classifier city_a_af11 operator and
[Switch-classifier-city_a_af11] if-match dscp af11
[Switch-classifier-city_a_af11] if-match acl 3000
[Switch-classifier-city_a_af11] quit
[Switch] traffic classifier city_a_be operator and
[Switch-classifier-city_a_be] if-match dscp 0
[Switch-classifier-city_a_be] if-match acl 3000
[Switch-classifier-city_a_be] quit
[Switch] traffic classifier city_a_default operator and
[Switch-classifier-city_a_default] if-match acl 3000
[Switch-classifier-city_a_default] quit
```

Create traffic classifiers **city_b_ef**, **city_b_af31**, **city_b_af11**, and **city_b_be** for traffic sent to city B. These traffic classifiers match traffic with DSCP priorities EF, AF31, AF11, and 0 and ACL 3001. Create a traffic classifier **city_b_default** for other traffic to match ACL 3001.

```
[Switch] traffic classifier city_b_ef operator and
[Switch-classifier-city_b_ef] if-match dscp ef
[Switch-classifier-city_b_ef] if-match acl 3001
[Switch-classifier-city_b_ef] quit
[Switch] traffic classifier city_b_af31 operator and
[Switch-classifier-city_b_af31] if-match dscp af31
[Switch-classifier-city_b_af31] if-match acl 3001
[Switch-classifier-city_b_af31] quit
[Switch] traffic classifier city_b_af11 operator and
[Switch-classifier-city_b_af11] if-match dscp af11
[Switch-classifier-city_b_af11] if-match acl 3001
[Switch-classifier-city_b_af11] quit
[Switch] traffic classifier city_b_be operator and
[Switch-classifier-city_b_be] if-match dscp 0
[Switch-classifier-city_b_be] if-match acl 3001
[Switch-classifier-city_b_be] quit
[Switch] traffic classifier city_b_default operator and
[Switch-classifier-city_b_default] if-match acl 3001
[Switch-classifier-city_b_default] quit
```

Step 4 Configure traffic behaviors.

Create traffic behaviors **city_a_ef**, **city_a_af31**, **city_a_af11**, and **city_a_be** to allocate CIR values 700 kbit/s, 400 kbit/s, 500 kbit/s, and 200 kbit/s to traffic with DSCP priorities EF, AF31, AF11, and 0. Create a traffic behavior **city_a_default** to allocate certain bandwidth to other traffic.

```
[Switch] traffic behavior city_a_ef
[Switch-behavior-city_a_ef] car cir 700 pir 2000
[Switch-behavior-city_a_ef] car city_a share
[Switch-behavior-city_a_ef] statistic enable
[Switch-behavior-city_a_ef] quit
[Switch] traffic behavior city_a_af31
[Switch-behavior-city_a_af31] car cir 400 pir 2000
[Switch-behavior-city_a_af31] car city_a share
[Switch-behavior-city_a_af31] statistic enable
[Switch-behavior-city_a_af31] quit
[Switch] traffic behavior city_a_af11
[Switch-behavior-city_a_af11] car cir 500 pir 2000
[Switch-behavior-city_a_af11] car city_a share
[Switch-behavior-city_a_af11] statistic enable
[Switch-behavior-city_a_af11] quit
[Switch] traffic behavior city_a_be
[Switch-behavior-city_a_be] car cir 200 pir 2000
```

```
[Switch-behavior-city_a_be] car city_a share
[Switch-behavior-city_a_be] statistic enable
[Switch-behavior-city_a_be] quit
[Switch] traffic behavior city_a_default
[Switch-behavior-city_a_default] car cir 64 pir 2000
[Switch-behavior-city_a_default] car city_a share
[Switch-behavior-city_a_default] statistic enable
[Switch-behavior-city_a_default] quit
```

Create traffic behaviors **city_b_ef**, **city_b_af31**, **city_b_af11**, and **city_b_be** to allocate CIR values 800 kbit/s, 500 kbit/s, 300 kbit/s, and 100 kbit/s to traffic with DSCP priorities EF, AF31, AF11, and 0. Create a traffic behavior **city_b_default** to allocate certain bandwidth to other traffic.

```
[Switch] traffic behavior city_b_ef
[Switch-behavior-city_b_ef] car cir 800 pir 2000
[Switch-behavior-city_b_ef] car city_b share
[Switch-behavior-city_b_ef] statistic enable
[Switch-behavior-city_b_ef] quit
[Switch] traffic behavior city_b_af31
[Switch-behavior-city_b_af31] car cir 500 pir 2000
[Switch-behavior-city_b_af31] car city_b share
[Switch-behavior-city_b_af31] statistic enable
[Switch-behavior-city_b_af31] quit
[Switch] traffic behavior city_b_af11
[Switch-behavior-city_b_af11] car cir 300 pir 2000
[Switch-behavior-city_b_af11] car city_b share
[Switch-behavior-city_b_af11] statistic enable
[Switch-behavior-city_b_af11] quit
[Switch] traffic behavior city_b_be
[Switch-behavior-city_b_be] car cir 100 pir 2000
[Switch-behavior-city_b_be] car city_b share
[Switch-behavior-city_b_be] statistic enable
[Switch-behavior-city_b_be] quit
[Switch] traffic behavior city_b_default
[Switch-behavior-city_b_default] car cir 64 pir 2000
[Switch-behavior-city_b_default] car city_b share
[Switch-behavior-city_b_default] statistic enable
[Switch-behavior-city_b_default] quit
```

Step 5 Configure a traffic policy.

Create and configure a traffic policy **city_control**, bind configured traffic classifiers and traffic behaviors to the traffic policy, and apply the traffic policy to the interface connecting the provincial center server and the Switch.

```
[Switch] traffic policy city_control
[Switch-trafficpolicy-city_control] classifier city_a_ef behavior city_a_ef
[Switch-trafficpolicy-city_control] classifier city_a_af31 behavior city_a_af31
[Switch-trafficpolicy-city_control] classifier city_a_af11 behavior city_a_af11
[Switch-trafficpolicy-city_control] classifier city_a_be behavior city_a_be
[Switch-trafficpolicy-city_control] classifier city_a_default behavior
city_a_default
[Switch-trafficpolicy-city_control] classifier city_b_ef behavior city_b_ef
[Switch-trafficpolicy-city_control] classifier city_b_af31 behavior city_b_af31
[Switch-trafficpolicy-city_control] classifier city_b_af11 behavior city_b_af11
[Switch-trafficpolicy-city_control] classifier city_b_be behavior city_b_be
[Switch-trafficpolicy-city_control] classifier city_b_default behavior
city_b_default
[Switch-trafficpolicy-city_control] quit
[Switch] interface gigabitethernet 2/0/1
[Switch-GigabitEthernet2/0/1] traffic-policy city_control inbound
[Switch-GigabitEthernet2/0/1] quit
```

Step 6 Verify the configuration.

- Verify that the traffic sent from the Switch to each city is within 2 Mbit/s.

Use a tester to simulate traffic with priorities EF, AF31, AF11, and BE and traffic with other priorities. Send each type of traffic from GE 2/0/1 of the Switch to city A and city B at a rate of 100 Mbit/s. Observe the received traffic on GE0/0/1 of SwitchA and SwitchB. You can see that the traffic rate is 2 Mbit/s.

- Verify that the traffic with other priorities has certain bandwidth.

  Use a tester to simulate traffic with DSCP priorities EF and AF21. Send each type of traffic from GE 2/0/1 of the Switch to city A at a rate of 100 Mbit/s. Observe the received traffic on GE0/0/1 of SwitchA. The bandwidth of traffic with DSCP priority EF is higher than 700 kbit/s, the bandwidth of traffic with DSCP priority AF21 is higher than 64 kbit/s, and the bandwidth sum of the two types of traffic is 2 Mbit/s.

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 qos car city_a cir 2000 cbs 376000
 qos car city_b cir 2000 cbs 376000
#
acl number 3000
 rule 5 permit tcp destination 10.1.0.0 0.0.255.255
acl number 3001
 rule 5 permit tcp destination 10.2.0.0 0.0.255.255
#
traffic classifier city_a_af11 operator and precedence 15
 if-match acl 3000
 if-match dscp af11
traffic classifier city_a_af31 operator and precedence 10
 if-match dscp af31
 if-match acl 3000
traffic classifier city_a_be operator and precedence 20
 if-match dscp default
 if-match acl 3000
traffic classifier city_a_default operator and precedence 25
 if-match acl 3000
traffic classifier city_a_ef operator and precedence 5
 if-match dscp ef
 if-match acl 3000
traffic classifier city_b_af11 operator and precedence 40
 if-match dscp af11
 if-match acl 3001
traffic classifier city_b_af31 operator and precedence 35
 if-match dscp af31
 if-match acl 3001
traffic classifier city_b_be operator and precedence 45
 if-match dscp default
 if-match acl 3001
traffic classifier city_b_default operator and precedence 50
 if-match acl 3001
traffic classifier city_b_ef operator and precedence 30
 if-match dscp ef
 if-match acl 3001
#
traffic behavior city_a_af11
 car cir 500 pir 2000 cbs 62500 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_a share
 statistic enable
traffic behavior city_a_af31
 car cir 400 pir 2000 cbs 50000 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_a share
```

```
 statistic enable
traffic behavior city_a_be
 car cir 200 pir 2000 cbs 25000 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_a share
 statistic enable
traffic behavior city_a_default
 car cir 64 pir 2000 cbs 10000 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_a share
 statistic enable
traffic behavior city_a_ef
 car cir 700 pir 2000 cbs 87500 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_a share
 statistic enable
traffic behavior city_b_af11
 car cir 300 pir 2000 cbs 37500 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_b share
 statistic enable
traffic behavior city_b_af31
 car cir 500 pir 2000 cbs 62500 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_b share
 statistic enable
traffic behavior city_b_be
 car cir 100 pir 2000 cbs 12500 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_b share
 statistic enable
traffic behavior city_b_default
 car cir 64 pir 2000 cbs 10000 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_b share
 statistic enable
traffic behavior city_b_ef
 car cir 800 pir 2000 cbs 100000 pbs 250000 mode color-blind green pass yellow
pass red discard
 car city_b share
 statistic enable
#
traffic policy city_control
 classifier city_a_ef behavior city_a_ef
 classifier city_a_af31 behavior city_a_af31
 classifier city_a_af11 behavior city_a_af11
 classifier city_a_be behavior city_a_be
 classifier city_a_default behavior city_a_default
 classifier city_b_ef behavior city_b_ef
 classifier city_b_af31 behavior city_b_af31
 classifier city_b_af11 behavior city_b_af11
 classifier city_b_be behavior city_b_be
 classifier city_b_default behavior city_b_default
#
interface GigabitEthernet1/0/1
 port link-type trunk
 port trunk allow-pass vlan 100 200
#
interface GigabitEthernet2/0/1
 port link-type trunk
 port trunk allow-pass vlan 100 200
 traffic-policy city_control inbound
#
return
```

2.6.4 Example for Configuring Traffic Shaping

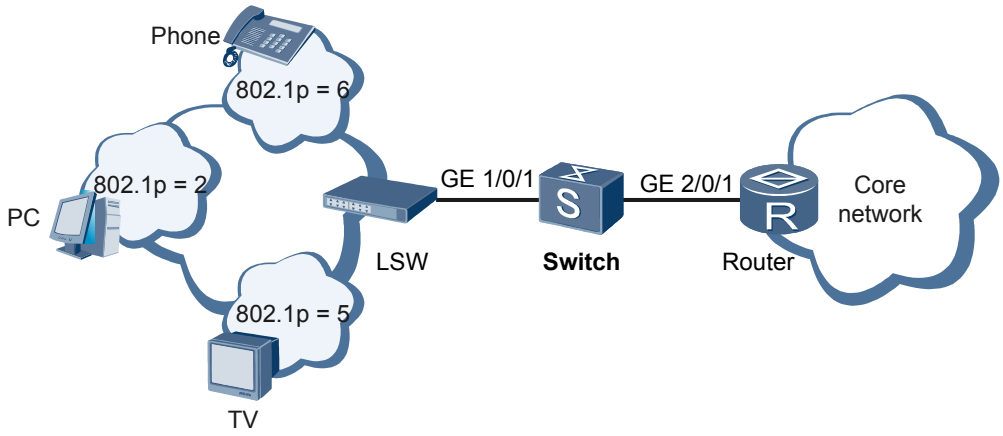
You can configure traffic shaping and set different traffic shaping rates for different types of packets to reduce the jitter and ensure bandwidth of various services.

Networking Requirements

The Switch is connected to GE 2/0/1 and the router; the 802.1p priorities of voice, video, and data services from the Internet are 6, 5, and 2 respectively, and these services can reach users through the router and Switch, as shown in **Figure 2-6**. The rate of the traffic from the network side is greater than the rate of the LSW interface; therefore, a jitter may occur in the outbound direction of GE 1/0/1. To reduce the jitter and ensure the bandwidth of various services, the requirements are as follows:

- The CIR on the interface is 10000 kbit/s.
- The CIR and PIR for the voice service are 3000 kbit/s and 5000 kbit/s respectively.
- The CIR and PIR for the video service are 5000 kbit/s and 8000 kbit/s respectively.
- The CIR and PIR for the data service are 2000 kbit/s and 3000 kbit/s respectively.

Figure 2-6 Networking diagram for configuring traffic shaping



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure priority mapping based on simple traffic classification to map priorities of packets to PHBs.
2. Configure traffic shaping on an interface to limit the bandwidth of the interface.
3. Configure traffic shaping in an interface queue to limit the CIRs of voice, video, and data services.

Data Preparation

To complete the configuration, you need the following data:

- 802.1p priorities being 6, 5, and 2 mapped to PHBs

- Rate for traffic shaping on an interface
- Rate for traffic shaping in each interface queue

Procedure

Step 1 Create VLANs and configure interfaces.

Create VLAN 10.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 10
```

Configure the type of GE 1/0/1 and GE 2/0/1 as trunk, and then add GE1/0/1 and GE2/0/1 to VLAN 10.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] port link-type trunk
[Switch-GigabitEthernet1/0/1] port trunk allow-pass vlan 10
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 10
[Switch-GigabitEthernet2/0/1] quit
```

Create VLANIF 10 and assign network segment address 10.10.10.1/24 to VLANIF 10.

```
[Switch] interface vlanif 10
[Switch-Vlanif10] ip address 10.10.10.1 255.255.255.0
[Switch-Vlanif10] quit
```

NOTE

Assign IP address 10.10.10.2/24 to the interface connecting the router and Switch.

Step 2 Configure priority mapping based on simple traffic classification.

Create DiffServ domain **ds1** in which 802.1p priorities being 6, 5, and 2 are mapped to PHBs CS7, EF, and AF2.

```
[Switch] diffserv domain ds1
[Switch-dsdomain-ds1] 8021p-inbound 6 phb cs7
[Switch-dsdomain-ds1] 8021p-inbound 5 phb ef
[Switch-dsdomain-ds1] 8021p-inbound 2 phb af2
[Switch-dsdomain-ds1] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] trust upstream ds1
[Switch-GigabitEthernet2/0/1] quit
```

Step 3 Configure traffic shaping on an interface.

Configure traffic shaping on an interface of the Switch and set the CIR to 10000 kbit/s.

```
[Switch] interface gigabitethernet 1/0/1
[Switch-GigabitEthernet1/0/1] qos lr cir 10000 outbound
```

Step 4 Configure traffic shaping in an interface queue.

Configure traffic shaping in the interface queues on the Switch, and then set the CIR and PIR of the voice service to 3000 kbit/s and 5000 kbit/s, the CIR and PIR of the video service to 5000 kbit/s and 8000 kbit/s, and the CIR and PIR of the data service to 2000 kbit/s and 3000 kbit/s.

```
[Switch-GigabitEthernet1/0/1] qos queue 7 shaping cir 3000 pir 5000
[Switch-GigabitEthernet1/0/1] qos queue 5 shaping cir 5000 pir 8000
[Switch-GigabitEthernet1/0/1] qos queue 2 shaping cir 2000 pir 3000
```

```
[Switch-GigabitEthernet1/0/1] quit
[Switch] quit
```

Step 5 Verify the configuration.

Check the configuration of DiffServ domain **ds1**.

```
<Switch> display diffserv domain name ds1
diffserv domain name:ds1
 8021p-inbound 0 phb be green
 8021p-inbound 1 phb af1 green
 8021p-inbound 2 phb af2 green
 8021p-inbound 3 phb af3 green
 8021p-inbound 4 phb af4 green
 8021p-inbound 5 phb ef green
 8021p-inbound 6 phb cs7 green
 8021p-inbound 7 phb cs7 green
 8021p-outbound be green map 0
 ......
```

If the configuration succeeds, the committed bandwidth for the packets transmitted by GE1/0/1 is 10000 kbit/s; the transmission rate of the voice service ranges from 3000 kbit/s to 5000 kbit/s; the transmission rate of the video service ranges from 5000 kbit/s to 8000 kbit/s; the transmission rate of the data service ranges from 2000 kbit/s to 3000 kbit/s.

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 10
#
diffserv domain ds1
 8021p-inbound 6 phb cs7 green
#
interface Vlanif10
 ip address 10.10.10.1 255.255.255.0
#
interface GigabitEthernet1/0/1
 port link-type trunk
 port trunk allow-pass vlan 10
 qos lr cir 10000 cbs 1250000 outbound
 qos queue 2 shaping cir 2000 pir 3000
 qos queue 5 shaping cir 5000 pir 8000
 qos queue 7 shaping cir 3000 pir 5000
#
interface GigabitEthernet2/0/1
 port link-type trunk
 port trunk allow-pass vlan 10
 trust upstream ds1
#
return
```

3 Congestion Avoidance and Congestion Management Configuration

About This Chapter

This chapter describes the basic concepts of congestion avoidance and congestion management. It also describes configuration methods and provides configuration examples of congestion avoidance and congestion management.

3.1 Overview of Congestion Avoidance and Congestion Management

This section describes the basic concepts of congestion avoidance and congestion management.

3.1.1 Congestion Avoidance

Congestion avoidance is a flow control mechanism. A system configured with congestion avoidance monitors network resource usage such as queues and memory buffers. When congestion occurs or aggravates, the system discards packets.

Congestion avoidance mechanisms include tail drop, Random Early Detection (RED), and Weighted Random Early Detection (WRED). The S7700 performs congestion avoidance based on WRED.

Tail Drop

The traditional packet drop policy uses tail drop. The tail drop policy processes all the packets uniformly, regardless of their class of service (CoS). When congestion occurs, packets at the end of a queue are discarded until the congestion problem is solved.

The tail drop policy causes global TCP synchronization. When packets from multiple TCP connections are discarded in a queue, these TCP connections enter the congestion avoidance and slow start state simultaneously, which is called global TCP synchronization. This causes traffic reduction and leads to traffic peak. As the process repeats, it causes the volume of network traffic to change from heavy to light and affects the link usage.

RED

The RED mechanism randomly discards packets so that the S7700 reduces the transmission speeds of multiple TCP connections at different periods of time. This prevents global TCP synchronization.

RED sets the upper threshold and lower threshold for the length of each queue and processes packets as follows:

- When the queue length is shorter than the lower threshold, no packet is discarded.
- When the queue length exceeds the upper threshold, all the received packets are discarded.
- When the queue length ranges from the lower threshold to the upper threshold, incoming packets are dropped randomly. The system sets a random number for each incoming packet, and compares it with the packet drop probability of the current queue. If the random number is larger than the drop probability, the packet is dropped. The longer the queue, the higher the drop probability.

WRED

The WRED mechanism also prevents global TCP synchronization by randomly discarding packets. The random number generated by WRED is based on the priority. WRED distinguishes the drop policy based on colors of packets, so the drop probability of packets with higher priorities is low.

3.1.2 Congestion Management

When intermittent congestion occurs on the network, delay-sensitive services require higher QoS than others. In this case, congestion management is required. The bandwidth needs to be increased if a network is always congested.

Congestion management uses the queue scheduling technologies. Currently, the S7700 adopts the following queue scheduling modes:

- **PQ Scheduling**
- **WRR Scheduling**
- **DRR Scheduling**
- **PQ+WRR/PQ+DRR Scheduling**

PQ Scheduling

Priority Queuing (PQ) scheduling is a queuing technology by which packets are scheduled based on the priorities of queues in a strict manner. The packets of lower priorities can be scheduled only after packets of higher priorities are scheduled.

In PQ scheduling mode, packets of delay-sensitive core services are put into a high priority queue and packets of other non-core services are put into a low priority queue. This ensures that core services are sent first.

The disadvantage of PQ scheduling is that the packets of lower priorities are not processed if there are a large number of packets of higher priorities, when congestion occurs.

WRR Scheduling

WRR refers to Weighted Round Robin. WRR schedules packets of queues in a polling manner, ensuring that packets in each queue are sent at a certain time.

Assume that there are eight output queues on an interface. WRR sets weights for the eight queues, that is, w7, w6, w5, w4, w3, w2, w1, and w0. The weight indicates a percentage of obtaining resources. For example, the weights of queues on a 100-Mbit/s interface are set to 50, 50, 30, 30, 10, 10, 10, and 10, corresponding to w7, w6, w5, w4, w3, w2, w1, and w0. In this case, the lowest priority queue can obtain bandwidth of at least 5 Mbit/s. This avoids the disadvantage of PQ scheduling.

The advantage of WRR is as follows: Although packets in multiple queues are processed in a polling manner, the time allocated to each queue is not fixed. If a queue is null, packets of the next queue are scheduled. This ensures better usage of bandwidth.

The disadvantages of WRR are as follows:

- WRR allocates bandwidth according to the number of packets. When the average length of packets in each queue is the same or known, you can obtain the required bandwidth by setting the weight of WRR. However, you cannot obtain the required bandwidth by setting the weight of WRR when the average length of packets in each queue changes.
- The packets of short-delay services such as voice services cannot be scheduled in time.

DRR Scheduling

The principle of Deficit Round Robin (DRR) is similar to the principle of WRR.

The difference is that WRR schedules packets according to the number of packets, but DRR schedules packets according to the length of packets. If the packet length exceeds the scheduling capability of a queue, DRR allows the deficit weight to ensure that packets of a long length are scheduled. When packets are scheduled in a polling manner again, this queue is not scheduled until the weight becomes positive. Then, this queue participates in DRR scheduling.

DRR scheduling offsets the disadvantage of PQ scheduling and one disadvantage of WRR scheduling, that is, bandwidth cannot be obtained according to the proportion.

The packets of short-delay services such as voice services cannot be scheduled in time in DRR mode.

PQ+WRR/PQ+DRR Scheduling

PQ scheduling, WRR scheduling, and DRR scheduling have the following advantages and disadvantages:

- If only PQ scheduling is used, packets of lower priorities cannot obtain the bandwidth for a long time.
- If only WRR or DRR scheduling is used, delay-sensitive services such as voice service cannot be scheduled first.
- PQ+WRR or PQ+DRR scheduling can use the advantages of both PQ and WRR or DRR scheduling and offset their disadvantages.

Through PQ+WRR or PQ+DRR scheduling, important protocol packets and delay-sensitive service packets are put in a PQ queue and specified bandwidth is allocated to this queue. Other packets are put into a WRR or DRR queue according to their priorities and scheduled in a polling manner according to the weight of the queue.

3.2 Configuring Congestion Avoidance

After congestion avoidance is configured, the S7700 processes packets of different colors based on the WRED configuration.

3.2.1 Establishing the Configuration Task

Before configuring congestion avoidance, familiarize yourself with the applicable environment, complete the pre-configuration tasks, and obtain the required data. This will help you complete the configuration task quickly and accurately.

Applicable Environment

To prevent congestion and solve the problem of global TCP synchronization, you can configure WRED to adjust the traffic on a network and remove the overload of the traffic on a network.

Pre-configuration Tasks

Before configuring congestion avoidance, complete the following tasks on the incoming interface:

- Configuring priority mapping based on simple traffic classification to map priorities of packets to PHBs and colors

- Configuring traffic policing based on complex traffic classification and the remarking action

NOTE

Before configuring congestion avoidance, you need to perform either of the preceding tasks to color packets as the basis of congestion avoidance.

Data Preparation

To configure congestion avoidance, you need the following data.

No.	Data
1	Upper threshold, lower threshold, and maximum drop percent of WRED

3.2.2 (Optional) Setting the Length of the Interface Queue

This section describes how to set the length of the interface queue and the length of the specified priority queue.

Context

The length of the interface priority queue is automatically managed by the system. You can set the length of the interface priority queue as required by using the **qos queue** command.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
qos queue queue-index length length-value
```

The length of the interface priority queue is set.

NOTE

The queue length on the G48SBC, G48TBC or LE2D2X08SED0 boards cannot be changed. The two boards support the maximum queue length by default.

----End

3.2.3 (Optional) Configuring the CFI Field as the Internal Drop Priority

After the CFI field is configured as the internal drop priority, if the rate of packets exceeds the CIR, the S7700 sets the value of the CFI field in packets to 1. When congestion occurs, the S7700 first discards the packets with the CFI field being 1.

Context

The Canonical Format Indicator (CFI) field in a VLAN tag is also called the Drop Eligible Indicator (DEI), and is used to mark the drop priority of packets in certain situations. When the rate of packets exceeds the CIR, the S7700 sets the DEI field of the packets to 1. That is, these packets have a high drop priority. If congestion occurs, subsequent devices first discard packets with the DEI field being 1.

If you need to set the CFI field as the internal drop priority on multiple interfaces, you can perform the configuration on the port group.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Or run the **port-group** *port-group-name* command to enter the port group view.

NOTE

- The interface type can be Ethernet, GE, XGE, or Eth-Trunk.
- Create a port group before performing this task. For details on how to create a port group, see Configuring the Interface Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

Step 3 Run:

```
dei enable
```

The CFI field is configured as the internal drop priority.

By default, the CFI field is not configured as the internal drop priority.

----End

3.2.4 Creating a WRED Drop Profile

This section describes how to create a WRED drop profile, and set the upper threshold, lower threshold, and maximum drop percent of the WRED drop profile for packets of different colors.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

```
drop-profile drop-profile-name
```

A drop profile is created and the drop profile view is displayed.

There is a default WRED drop profile. You cannot delete the default WRED drop profile, but can modify the values of the parameters.

Step 3 Run:

```
color { green | non-tcp | red | yellow } low-limit low-limit-percentage high-limit high-limit-percentage discard-percentage discard-percentage
```

WRED parameters are set.

By default, the upper threshold, lower threshold, and maximum drop percent of a WRED drop profile are 100.

NOTE

The WRED algorithm for non-TCP packets cannot be used on S series boards and G48SBC boards.

For more information about boards, see Board Classification in the *S7700 Smart Routing Switch Hardware Description*.

----End

3.2.5 Applying the WRED Drop Profile

The configured WRED drop profile takes effect only after being applied. You can apply the WRED drop profile to an interface or a queue.

Context

You can apply a WRED drop profile to an interface or in an interface queue or to an interface, and an interface queue on the S7700 as required. The following takes place when the WRED drop profiles are applied:

If WRED drop profiles are applied to an interface and an interface queue on the S7700, the S7700 matches packets with WRED drop profiles in the interface queue and the interface in sequence. Then the S7700 performs congestion avoidance for the matched packets.

To set the same WRED drop profile on multiple interfaces, perform the configuration on the port group to reduce the workload.

Before applying a WRED drop profile, run the **drop-profile** command to create a WRED drop profile.

Procedure

- Applying a WRED drop profile to an interface
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     interface interface-type interface-number
     ```
     The interface view is displayed.
  3. Run:
     ```
     qos wred drop-profile-name
     ```
     A WRED drop profile is applied to the interface.
- Applying a WRED drop profile to a port group

1. Run:
   ```
   system-view
   ```
   The system view is displayed.
2. Run:
   ```
   port-group port-group-name
   ```
   The port group view is displayed.

   NOTE

   Create a port group before performing this task. For details on how to create a port group, see (Optional) Configuring a Port Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.
3. Run:
   ```
   qos wred drop-profile-name
   ```
   The WRED drop profile is applied to a port group.

- Applying a WRED drop profile to an interface queue
  1. Run:
     ```
     system-view
     ```
     The system view is displayed.
  2. Run:
     ```
     interface interface-type interface-number
     ```
     The interface view is displayed.
  3. Run:
     ```
     qos queue queue-index wred drop-profile-name
     ```
     The WRED drop profile is applied to an interface queue.

  drop-profile-name specifies the name of a WRED drop profile and must be the same as the name of a WRED drop profile in **3.2.4 Creating a WRED Drop Profile**.

----End

3.2.6 Checking the Configuration

After congestion avoidance is configured, you can view the name, index, and parameters of the WRED drop profile.

Prerequisites

The configurations of the WRED drop profile are complete.

Procedure

- Run the **display drop-profile** [**all** | **name** *drop-profile-name*] command to check the configuration of the WRED drop profile.
- Run the **display qos configuration interface** *interface-type interface-number* command to check all the QoS configurations on the interface.

----End

3.3 Configuring Congestion Management

After congestion management is configured, if congestion occurs on a network, the S7700 determines the sequence of forwarding packets according to the defined scheduling policy.

3.3.1 Establishing the Configuration Task

Before configuring congestion management, familiarize yourself with the applicable environment, complete the pre-configuration tasks, and obtain the required data. This will help you complete the configuration task quickly and accurately.

Applicable Environment

When congestion occurs, you can configure congestion management in the following situations:

- The same delay and jitter are set for various types of packets, and packets of core services such as video and voice services need to be processed first.
- Packets of non-core services of the same priority, such as email, are processed in a fair manner, and services of different priorities are processed according to the weights.

Pre-configuration Tasks

Before configuring congestion management, complete the following tasks:

- Configuring priority mapping based on simple traffic classification to map priorities of packets to PHBs
- Configuring the remarking action of inner priorities based on complex traffic classification

NOTE

Before configuring congestion management, you need to perform either of the preceding tasks to map packets to different queues for scheduling.

Data Preparation

To configure congestion management, you need the following data.

No.	Data
1	Mapping between the local precedence and queues.
2	Mode of queue scheduling.
3	Weight of queues in deficit round robin (DRR) scheduling mode.
4	Weight of queues in weighted round robin (WRR) scheduling mode.
5	(Optional) Minimum size of the static buffer for a queue.
6	(Optional) Maximum number of packets for a queue

3.3.2 (Optional) Setting the Length of the Interface Queue

This section describes how to set the length of the interface queue and the length of the specified priority queue.

Context

The length of the interface priority queue is automatically managed by the system. You can set the length of the interface priority queue as required by using the **qos queue** command.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

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

The interface view is displayed.

Step 3 Run:

```
qos queue queue-index length length-value
```

The length of the interface priority queue is set.

NOTE

The queue length on the G48SBC, G48TBC or LE2D2X08SED0 boards cannot be changed. The two boards support the maximum queue length by default.

----End

3.3.3 Setting the Scheduling Mode for an Interface Queue

The S7700 supports the following scheduling modes: PQ, DRR, WRR, PQ+DRR, and PQ +WRR.

Context

The S7700 supports eight interface queues that can use different scheduling algorithms. During queue scheduling, packets in a PQ queue are first scheduled. If there are multiple PQ queues, the packets are scheduled in descending order of priorities of these PQ queues. After packets in PQ queues are scheduled, packets in WRR queues are scheduled in a polling manner.

If you need to set the same scheduling parameters on multiple interfaces, you can perform the configuration on the port group to reduce the workload.

Procedure

Step 1 Run:

```
system-view
```

The system view is displayed.

Step 2 Run:

interface *interface-type interface-number*

The interface view is displayed.

Or run the **port-group** *port-group-name* command to enter the port group view.

NOTE

Create a port group before performing this task. For details about creating a port group, see (Optional) Configuring a Port Group in the *S7700 Smart Routing Switch Configuration Guide - Ethernet*.

Step 3 Run the following commands as required.

- WAN-series board

  Run:

  qos queue *queue-index* **wfq weight** *weight*

  The weight for WFQ scheduling is set.

  The value of *queue-index* ranges from 0 to 4. Queues 5 to 7 use PQ scheduling.

  NOTE

  The implementation of WFQ scheduling is similar to that of DRR scheduling. For details about DRR scheduling, see **3.1.2 Congestion Management**.

- Other boards

  1. Run:

     qos { **pq** | **wrr** | **drr** }

     The scheduling mode of queues on the interface is set to PQ, WRR, or DRR.

     Or, run:

     qos { **pq** { *start-queue-index* [**to** *end-queue-index*] } &<1-8> | { **wrr** | **drr** } { *start-queue-index* [**to** *end-queue-index*] } &<1-8> }*

     The scheduling mode of queues on the interface is set to PQ+WRR or PQ+DRR.

     By default, all queues use PQ scheduling.

     NOTE

     On the X40SFC boards, queue 6 and queue 7 use PQ scheduling, and queues 0 to 5 can use WRR or DRR scheduling.

  2. (Optional) Run:

     qos queue *queue-index* **wrr weight** *weight*

     The weight for WRR or DRR scheduling is set.

     By default, the weight for WRR or DRR scheduling is 1.

     This step is required only when the scheduling mode is WRR or PQ+WRR.

  3. (Optional) Run:

     qos queue *queue-index* **drr weight** *weight*

     The weight for DRR scheduling is set.

     By default, the weight for DRR scheduling is 1.

     This step is required only when the scheduling mode is DRR or PQ+DRR.

----End

3.3.4 Checking the Configuration

After congestion management is configured, you can view the queue-based traffic statistics and the scheduling parameters of the queues on a specified interface.

Prerequisites

The congestion management configurations are complete.

Procedure

- Run the **display qos queue length interface** *interface-type interface-number* command to check the usage of the priority queue on a specified interface.
- Run the **display qos configuration interface** *interface-type interface-number* command to check all the QoS configurations on the interface.

----End

3.4 Maintaining Congestion Avoidance and Congestion Management

This section describes how to maintain traffic avoidance and congestion management.

3.4.1 Displaying the Queue-based Statistics

You can use display commands to view the queue-based traffic statistics such as the number of forwarded and discarded packets.

Context

To view the queue-based traffic statistics, run the following command in any view.

Procedure

- **display qos queue statistics interface** *interface-type interface-number*Run the commands to view the queue-based traffic statistics based on device model.

  NOTE

  - S-series boards do not support the **display qos queue statistics** command.
  - For details about boards, see Board Classification in the *S7700 Smart Routing Switch - Hardware Description*.

----End

3.4.2 Checking the Usage of the Queue

You can use display commands to view the Usage of the Queue.

Context

To obtain the usage of queues, you can run the following command in any view.

Procedure

- Run the **display qos queue length interface** *interface-type interface-number* command to view the usage of priority queues on the interface.

  The command output on different types of boards is different:

- S-series boards

  The system allocates the minimum buffer length to each queue on the interface. When a queue uses up its buffer, it obtains the shared buffer on the interface. By default, the used length of a queue is the sum of the buffer length of the queue (N) and the shared buffer length of the interface (M).

  After the queue length is set:

  - If the queue length is smaller than M, the used queue length is M.
  - If the queue length is larger than M, the used queue length is the configured value that is converted by the system. For example, when the queue length is set to 61442, the displayed value is 61568.

- E-series boards

  By default, the used queue length is the number of bytes of packets buffered in the queue.

  After the queue length is set, the used queue length is the configured value that is converted by the system. For example, when the queue length is set to 61442, the displayed value is 61568.

NOTE

G48SBC boards, G48TBC boards and LE2D2X08SED0 boards do not support the **display qos queue length** command.

For details about boards, see Board Classification in the *S7700 Smart Routing Switch Hardware Description*.

----End

3.4.3 Clearing the Queue-based Statistics

You can use the reset command to clear the queue-based traffic statistics.

Context

To re-collect the queue-based statistics on an interface, you can use the following command in the user view to clear the previous statistics.

CAUTION

The queue-based statistics cannot be restored after you clear them. So, confirm the action before you use the command.

Procedure

- Clear the queue-based traffic statisticsS7700.

  Run the **reset qos queue statistics interface** *interface-type interface-number* command to clear the queue-based traffic statistics on the interface.

NOTE

The **reset qos queue statistics** command cannot be used on S-series boards.

For details about boards, see Board Classification in the *S7700 Smart Routing Switch - Hardware Description*.

----End

3.5 Configuration Examples

This section provides several configuration examples of congestion avoidance and congestion management.

3.5.1 Example for Configuring Congestion Avoidance and Congestion Management

After congestion avoidance and congestion management are configured, the S7700 provides different services for packets of different priorities and ensures high-priority and low-delay services.

Networking Requirements

The Switch is connected to the router through GE 2/0/1; the 802.1p priorities of voice, video, and data services on the Internet are 6, 5, and 2 respectively, and these services can reach users through the router and Switch, as shown in **Figure 3-1**. The rate of incoming interface GE 2/0/1 on the Switch is greater than the rates of outgoing interfaces GE 1/0/1 and GE 1/0/2; therefore, congestion may occur on these two outgoing interfaces. To reduce the effect caused by congestion and ensure that high-priority and short-delay services are processed first, the requirements are as follows.

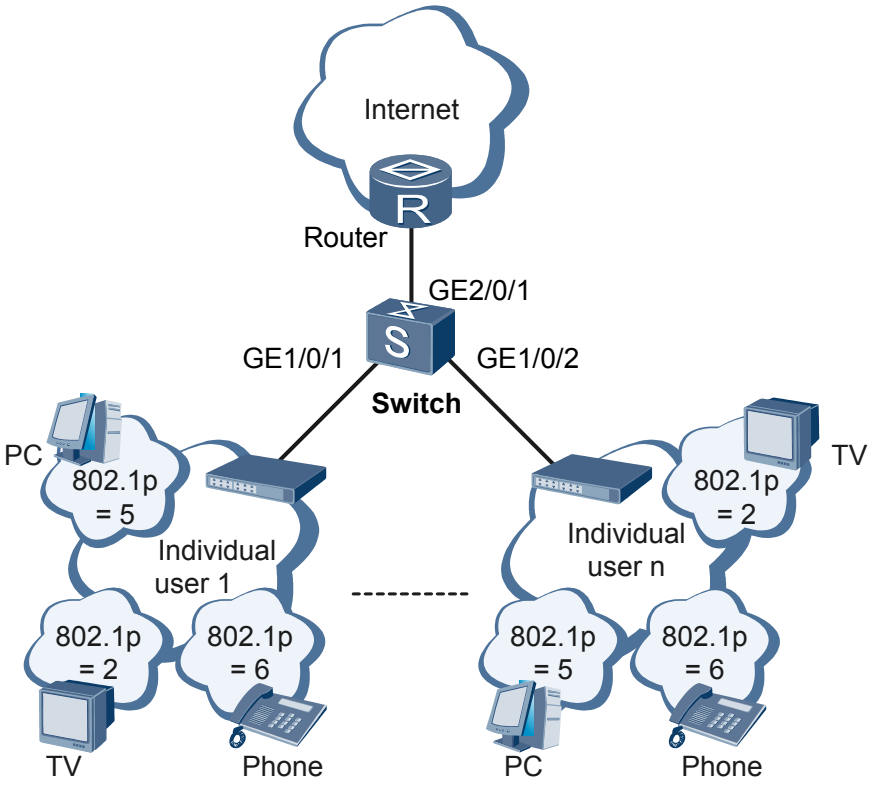
Table 3-1 Congestion avoidance parameters

Types of Services	Color	Lower Threshold (%)	Upper Threshold (%)	Drop Percent
Voice	Green	80	100	10
Video	Yellow	60	80	20
Data	Red	40	60	40

Table 3-2 Congestion management parameters

Type of Services	CoS	WRR
Voice	EF	0
Video	AF3	100
Data	AF1	50

Figure 3-1 Networking diagram for configuring congestion avoidance and congestion management



Configuration Roadmap

The configuration roadmap is as follows:

1. Configure the VLAN for each interface so that the devices can communicate with each other.
2. Create and configure a DiffServ domain on the Switch, map packets of 802.1p priorities to PHBs and colors of packets, and bind the DiffServ domain to an incoming interface on the Switch.
3. Create a WRED drop profile on the Switch and apply the WRED drop profile on an outgoing interface.
4. Set scheduling parameters of queues of different CoS on outgoing interfaces of the Switch.

Data Preparation

To complete the configuration, you need the following data:

- VLAN IDs of data packets, video packets, and voice packets, namely, 2, 5, and 6
- PHBs mapped to 802.1p priorities being 6, 5, and 2 and colors
- Name of the WRED drop profile and WRED parameters
- Scheduling parameters of queues of different CoS

Procedure

Step 1 Configure the VLAN for each interface so that the devices can communicate with each other.

```
<Quidway> system-view
[Quidway] sysname Switch
[Switch] vlan batch 2 5 6
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] port link-type trunk
[Switch-GigabitEthernet1/0/1] port trunk allow-pass vlan 2 5 6
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] port link-type trunk
[Switch-GigabitEthernet1/0/2] port trunk allow-pass vlan 2 5 6
[Switch-GigabitEthernet1/0/2] quit
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] port link-type trunk
[Switch-GigabitEthernet2/0/1] port trunk allow-pass vlan 2 5 6
[Switch-GigabitEthernet2/0/1] quit
```

Step 2 Configure priority mapping based on simple traffic classification.

Create DiffServ domain **ds1**, map packets of 802.1p priorities being 6, 5, and 2 to PHBs CS6, EF, and AF2, and color packets as green, yellow, and red.

```
[Quidway] sysname Switch
[Switch] diffserv domain ds1
[Switch-dsdomain-ds1] 8021p-inbound 6 phb ef green
[Switch-dsdomain-ds1] 8021p-inbound 5 phb af3 yellow
[Switch-dsdomain-ds1] 8021p-inbound 2 phb af1 red
[Switch-dsdomain-ds1] quit
```

Bind incoming interface GE 2/0/1 on the Switch to DiffServ domain **ds1**.

```
[Switch] interface gigabitethernet2/0/1
[Switch-GigabitEthernet2/0/1] trust upstream ds1
[Switch-GigabitEthernet2/0/1] trust 8021p inner
[Switch-GigabitEthernet2/0/1] quit
```

Step 3 Configure congestion avoidance.

Create drop profile **wred1** on the Switch and set parameters of packets of three colors.

```
[Switch] drop-profile wred1
[Switch-drop-wred1] color green low-limit 80 high-limit 100 discard-percentage 10
[Switch-drop-wred1] color yellow low-limit 60 high-limit 80 discard-percentage 20
[Switch-drop-wred1] color red low-limit 40 high-limit 60 discard-percentage 40
[Switch-drop-wred1] quit
```

Apply drop profile **wred1** on outgoing interfaces GE 1/0/1 and GE 1/0/2 of the Switch.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] qos wred wred1
[Switch-GigabitEthernet1/0/1] qos queue 5 wred wred1
[Switch-GigabitEthernet1/0/1] qos queue 3 wred wred1
[Switch-GigabitEthernet1/0/1] qos queue 1 wred wred1
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/1] qos wred wred1
[Switch-GigabitEthernet1/0/1] qos queue 5 wred wred1
[Switch-GigabitEthernet1/0/1] qos queue 3 wred wred1
[Switch-GigabitEthernet1/0/1] qos queue 1 wred wred1
[Switch-GigabitEthernet1/0/1] quit
```

Step 4 Configure congestion management.

Set scheduling parameters of queues of different CoS on outgoing interfaces GE 1/0/1 and GE 1/0/2 of the Switch.

```
[Switch] interface gigabitethernet1/0/1
[Switch-GigabitEthernet1/0/1] qos pq 7
[Switch-GigabitEthernet1/0/1] qos drr 0 to 6
[Switch-GigabitEthernet1/0/1] qos queue 3 drr weight 100
[Switch-GigabitEthernet1/0/1] qos queue 1 drr weight 50
[Switch-GigabitEthernet1/0/1] quit
[Switch] interface gigabitethernet1/0/2
[Switch-GigabitEthernet1/0/2] qos pq 7
[Switch-GigabitEthernet1/0/2] qos drr 0 to 6
[Switch-GigabitEthernet1/0/2] qos queue 3 drr weight 100
[Switch-GigabitEthernet1/0/2] qos queue 1 drr weight 50
[Switch-GigabitEthernet1/0/2] quit
[Switch] quit
```

Step 5 Verify the configuration.

Check the configuration of DiffServ domain **ds1**.

```
<Switch> display diffserv domain name ds1
diffserv domain name:ds1
 8021p-inbound 0 phb be green
 8021p-inbound 1 phb af1 green
 8021p-inbound 2 phb af1 red
 8021p-inbound 3 phb af3 green
 8021p-inbound 4 phb af4 green
 8021p-inbound 5 phb af3 yellow
 8021p-inbound 6 phb ef green
 8021p-inbound 7 phb cs7 green
 8021p-outbound be green map 0
```

Check the configuration of drop profile **wred1**.

```
<Switch> display drop-profile name wred1
Drop-profile[3]: wred1
Color     Low-limit   High-limit  Discard-percentage
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Green     80          100         10
Yellow    60          80          20
Red       40          60          40
Non-tcp   100         100         100
------------------------------------------------------------------
```

----End

Configuration Files

- Configuration file of the Switch

```
#
 sysname Switch
#
 vlan batch 2 5 6
#
diffserv domain ds1
 8021p-inbound 2 phb af1 red
 8021p-inbound 5 phb af3 yellow
 8021p-inbound 6 phb ef green
#
 drop-profile wred1
  color green low-limit 80 high-limit 100 discard-percentage 10
  color yellow low-limit 60 high-limit 80 discard-percentage 20
  color red low-limit 40 high-limit 60 discard-percentage 40
#
interface GigabitEthernet1/0/1
 port link-type trunk
 port trunk allow-pass vlan 2 5 6
 qos wred wred1
 qos pq 5 drr 0 to 6
 qos queue 1 drr weight 50
 qos queue 3 drr weight 100
```

```
 qos queue 1 wred wred1
 qos queue 3 wred wred1
 qos queue 5 wred wred1
#
interface GigabitEthernet1/0/2
 port link-type trunk
 port trunk allow-pass vlan 2 5 6
 qos wred wred1
 qos pq 5 drr 0 to 6
 qos queue 1 drr weight 50
 qos queue 3 drr weight 100
 qos queue 1 wred wred1
 qos queue 3 wred wred1
 qos queue 5 wred wred1
#
interface GigabitEthernet2/0/1
 port link-type trunk
 port trunk allow-pass vlan 2 5 6
 trust upstream ds1
 trust 8021p inner
#
return
```