

# H3C S6850 & S9850 & S9820-64H

## Traffic Policing Configuration Examples

---

Copyright © 2020 New H3C Technologies Co., Ltd. All rights reserved.

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

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

The information in this document is subject to change without notice.

# Contents

Introduction.....	1
Prerequisites .....	1
Example: Policing traffic by IP address and protocol type.....	1
Network configuration .....	1
Analysis .....	2
Applicable hardware and software versions .....	2
Restrictions and guidelines .....	2
Procedures.....	3
Verifying the configuration.....	4
Configuration files .....	6
Example: Allocating bandwidth based on VLANs .....	7
Network configuration .....	7
Analysis .....	8
Applicable hardware and software versions .....	8
Restrictions and guidelines .....	9
Procedures.....	9
Configuring VLAN settings .....	9
Configuring traffic policing .....	10
Verifying the configuration.....	12
Configuration files .....	13
Example: Configuring aggregate CAR .....	15
Network configuration .....	15
Applicable hardware and software versions .....	16
Procedures.....	16
Verifying the configuration.....	17
Configuration files .....	17
Related documentation.....	18

# Introduction

This chapter provides examples for configuring traffic policing and aggregate CAR to control network traffic.

## Prerequisites

The configuration examples in this document were created and verified in a lab environment, and all the devices were started with the factory default configuration. When you are working on a live network, make sure you understand the potential impact of every command on your network.

This document assumes that you have basic knowledge of traffic policing.

## Example: Policing traffic by IP address and protocol type

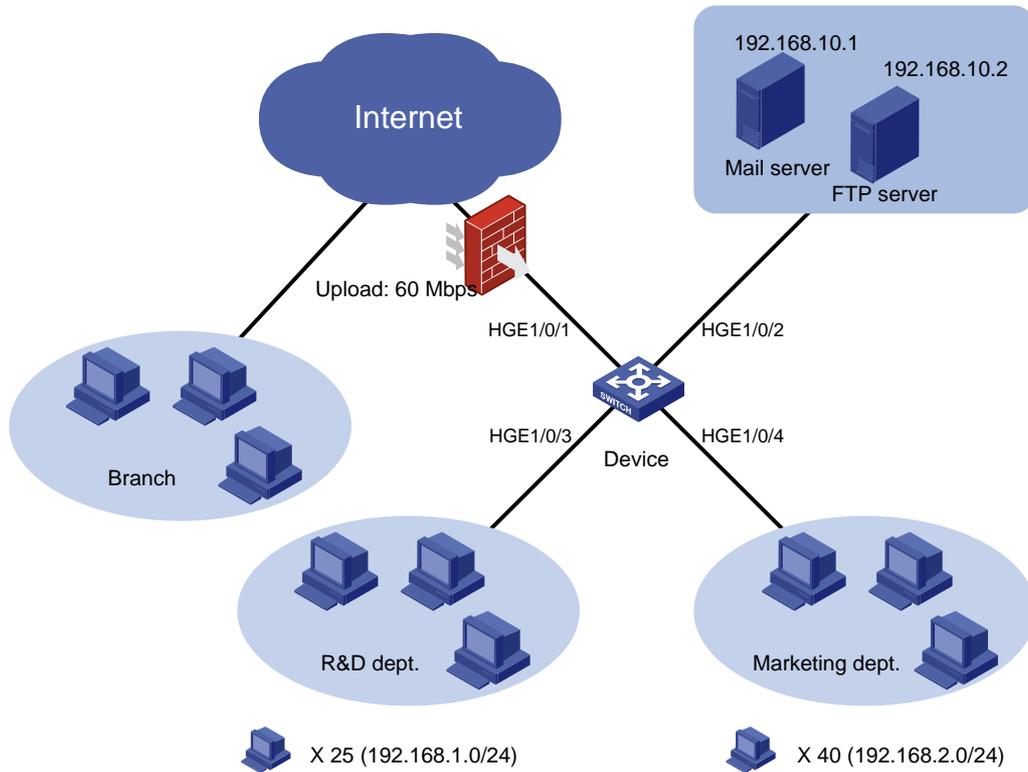
### Network configuration

As shown in [Figure 1](#), a company uses a dedicated line to access the Internet, with an uplink bandwidth of 60 Mbps. All end devices use the firewall as the gateway. The mail server forwards emails for all clients to the external network. The FTP server provides data services for the branch through the Internet.

Configure traffic policing to classify and rate limit the uplink traffic as follows:

- **HTTP traffic**—Rate limit HTTP traffic to a total rate of 40 Mbps (15 Mbps for the 25 hosts in the R&D department and 25 Mbps for the 40 hosts in the Marketing department).
- **Email traffic**—Rate limit email traffic to 2 Mbps.
- **FTP traffic**—Rate limit FTP traffic to 10 Mbps.

Figure 1 Network diagram



## Analysis

To meet the network requirements, you must perform the following tasks:

- Configure ACLs to classify packets of different types.
- Associate classes with policing actions to rate limit packets of different types.

## Applicable hardware and software versions

The following matrix shows the hardware and software versions to which this configuration example is applicable:

Hardware	Software version
S6850 switch series S9850 switch series	Release 6555P01
S9820-64H switch	Release 6555P01

## Restrictions and guidelines

In a traffic behavior, the traffic policing action cannot be configured together with the following priority marking actions:

- Local precedence marking.
- Drop precedence marking.

- 802.1p priority marking.
- DSCP marking.
- IP precedence marking.

Otherwise, the class-behavior association for that traffic behavior cannot be applied correctly.

## Procedures

### 1. Police HTTP traffic from the R&D department:

# Create advanced IPv4 ACL 3000 to match HTTP traffic from the R&D department.

```
<Device> system-view
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule permit tcp destination-port eq 80 source
192.168.1.0 0.0.0.255
[Device-acl-ipv4-adv-3000] quit
```

# Create a class named **rd\_http**, and use advanced IPv4 ACL 3000 as the match criterion.

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

# Create a behavior named **rd\_http**, and configure traffic policing with the CIR of 15 Mbps.

```
[Device] traffic behavior rd_http
[Device-behavior-rd_http] car cir 15360
[Device-behavior-rd_http] quit
```

# Create a QoS policy named **rd\_http**, and associate the class **rd\_http** with the behavior **rd\_http** in the QoS policy.

```
[Device] qos policy rd_http
[Device-qospolicy-rd_http] classifier rd_http behavior rd_http
[Device-qospolicy-rd_http] quit
```

# Apply the QoS policy **rd\_http** to the inbound direction of interface HundredGigE 1/0/3.

```
[Device] interface hundredgige 1/0/3
[Device-HundredGigE1/0/3] qos apply policy rd_http inbound
[Device-HundredGigE1/0/3] quit
```

### 2. Police HTTP traffic from the Marketing department:

# Create advanced IPv4 ACL 3001 to match HTTP traffic from the Marketing department.

```
[Device] acl advanced 3001
[Device-acl-ipv4-adv-3001] rule permit tcp destination-port eq 80 source
192.168.2.0 0.0.0.255
[Device-acl-ipv4-adv-3001] quit
```

# Create a class named **mkt\_http**, and use advanced IPv4 ACL 3001 as the match criterion.

```
[Device] traffic classifier mkt_http
[Device-classifier-mkt_http] if-match acl 3001
[Device-classifier-mkt_http] quit
```

# Create a behavior named **mkt\_http**, and configure traffic policing with the CIR of 25 Mbps.

```
[Device] traffic behavior mkt_http
[Device-behavior-mkt_http] car cir 25600
[Device-behavior-mkt_http] quit
```

# Create a QoS policy named **mkt\_http**, and associate the class **mkt\_http** with the behavior **mkt\_http** in the QoS policy.

```

[Device] qos policy mkt_http
[Device-qospolicy-mkt_http] classifier mkt_http behavior mkt_http
[Device-qospolicy-mkt_http] quit
# Apply the QoS policy mkt_http to the inbound direction of interface HundredGigE 1/0/4.
[Device] interface hundredgige 1/0/4
[Device-HundredGigE1/0/4] qos apply policy mkt_http inbound
[Device-HundredGigE1/0/4] quit

```

**3. Police email traffic and FTP traffic:**

```

# Create advanced IPv4 ACL 3002 to match email traffic.
[Device] acl advanced 3002
[Device-acl-ipv4-adv-3002] rule permit tcp destination-port eq smtp source
192.168.10.1 0.0.0.0
[Device-acl-ipv4-adv-3002] quit
# Create a class named email, and use advanced IPv4 ACL 3002 as the match criterion.
[Device] traffic classifier email
[Device-classifier-email] if-match acl 3002
[Device-classifier-email] quit
# Create a behavior named email, and configure traffic policing with the CIR of 2 Mbps.
[Device] traffic behavior email
[Device-behavior-email] car cir 2048
[Device-behavior-email] quit
# Create basic IPv4 ACL 2001 to match FTP traffic.
[Device] acl basic 2001
[Device-acl-ipv4-basic-2001] rule permit source 192.168.10.2 0.0.0.0
[Device-acl-ipv4-basic-2001] quit
# Create a class named ftp, and use basic IPv4 ACL 2001 as the match criterion.
[Device] traffic classifier ftp
[Device-classifier-ftp] if-match acl 2001
[Device-classifier-ftp] quit
# Create a behavior named ftp, and configure traffic policing with the CIR of 10 Mbps.
[Device] traffic behavior ftp
[Device-behavior-ftp] car cir 10240
[Device-behavior-ftp] quit
# Create a QoS policy named email&ftp, and associate the classes email and ftp with the
behavior email and ftp in the QoS policy, respectively.
[Device] qos policy email&ftp
[Device-qospolicy-email&ftp] classifier email behavior email
[Device-qospolicy-email&ftp] classifier ftp behavior ftp
[Device-qospolicy-email&ftp] quit
# Apply the QoS policy email&ftp to the inbound direction of interface HundredGigE 1/0/2.
[Device] interface hundredgige 1/0/2
[Device-HundredGigE1/0/2] qos apply policy email&ftp inbound
[Device-HundredGigE1/0/2] quit

```

## Verifying the configuration

```
# Verify QoS policies applied to interfaces.
```

[Device] display qos policy interface

Interface: HundredGigE1/0/2

Direction: Inbound

Policy: email&ftp

Classifier: email

Operator: AND

Rule(s) :

If-match acl 3002

Behavior: email

Committed Access Rate:

CIR 2048 (kbps), CBS 128000 (Bytes), EBS 0 (Bytes)

Green action : pass

Yellow action : pass

Red action : discard

Green packets : 0 (Packets)

Red packets : 0 (Packets)

Classifier: ftp

Operator: AND

Rule(s) :

If-match acl 2001

Behavior: ftp

Committed Access Rate:

CIR 10240 (kbps), CBS 640000 (Bytes), EBS 0 (Bytes)

Green action : pass

Yellow action : pass

Red action : discard

Green packets : 0 (Packets)

Red packets : 0 (Packets)

Interface: HundredGigE1/0/3

Direction: Inbound

Policy: rd\_http

Classifier: rd\_http

Operator: AND

Rule(s) :

If-match acl 3000

Behavior: rd\_http

Committed Access Rate:

CIR 15360 (kbps), CBS 960000 (Bytes), EBS 0 (Bytes)

Green action : pass

Yellow action : pass

Red action : discard

Green packets : 0 (Packets)

Red packets : 0 (Packets)

Interface: HundredGigE1/0/4

Direction: Inbound

Policy: mkt\_http

```
Classifier: mkt_http
Operator: AND
Rule(s) :
  If-match acl 3001
Behavior: mkt_http
Committed Access Rate:
  CIR 25600 (kbps), CBS 1600000 (Bytes), EBS 0 (Bytes)
  Green action : pass
  Yellow action : pass
  Red action   : discard
  Green packets : 0 (Packets)
  Red packets  : 0 (Packets)
```

## Configuration files

```
#
traffic classifier email operator and
  if-match acl 3002
#
traffic classifier ftp operator and
  if-match acl 2001
#
traffic classifier mkt_http operator and
  if-match acl 3001
#
traffic classifier rd_http operator and
  if-match acl 3000
#
traffic behavior email
  car cir 2048 cbs 128000 ebs 0 green pass red discard yellow pass
#
traffic behavior ftp
  car cir 10240 cbs 640000 ebs 0 green pass red discard yellow pass
#
traffic behavior mkt_http
  car cir 25600 cbs 1600000 ebs 0 green pass red discard yellow pass
#
traffic behavior rd_http
  car cir 15360 cbs 960000 ebs 0 green pass red discard yellow pass
#
qos policy email&ftp
  classifier email behavior email
  classifier ftp behavior ftp
#
qos policy mkt_http
  classifier mkt_http behavior mkt_http
#
qos policy rd_http
```

```

classifier rd_http behavior rd_http
#
interface HundredGigE1/0/2
port link-mode bridge
qos apply policy email&ftp inbound
#
interface HundredGigE1/0/3
port link-mode bridge
qos apply policy rd_http inbound
#
interface HundredGigE1/0/4
port link-mode bridge
qos apply policy mkt_http inbound
#
acl basic 2001
rule 0 permit source 192.168.10.2 0
#
acl advanced 3000
rule 0 permit tcp source 192.168.1.0 0.0.0.255 destination-port eq www
#
acl advanced 3001
rule 0 permit tcp source 192.168.2.0 0.0.0.255 destination-port eq www
#
acl advanced 3002
rule 0 permit tcp source 192.168.10.1 0 destination-port eq smtp

```

# Example: Allocating bandwidth based on VLANs

## Network configuration

As shown in [Figure 2](#), the device aggregates traffic from the branches and transmits the traffic to the backbone network through a leased line. Each branch site assigns packets of different applications to different VLANs.

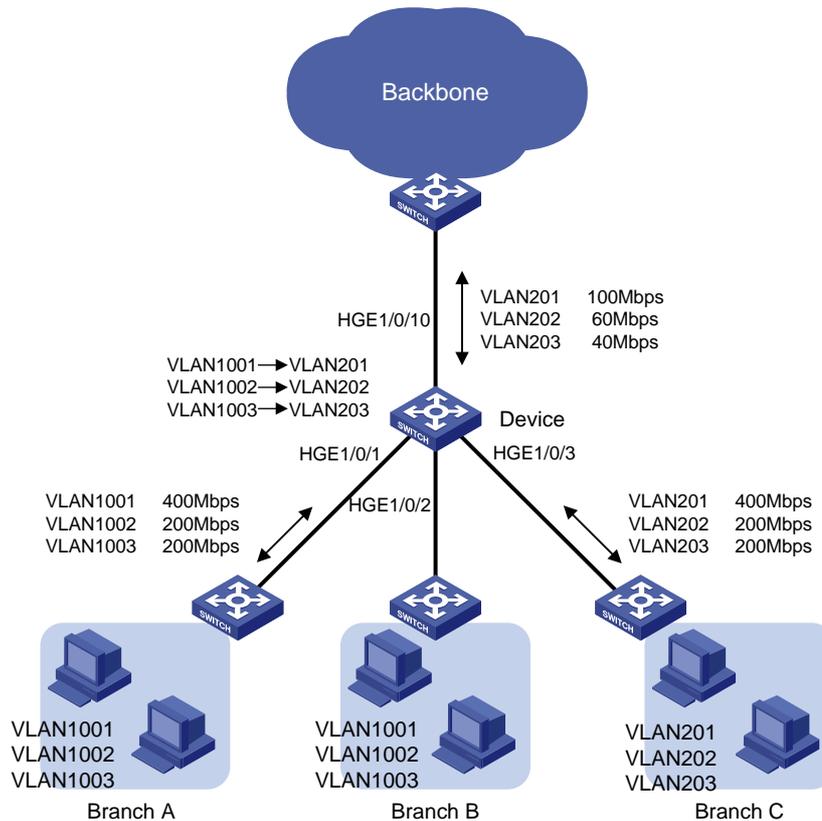
- Configure one-to-one VLAN mapping on the following interfaces of the device to re-map traffic of different applications to VLANs as per the transmission scheme on the backbone network:
  - HundredGigE 1/0/1.
  - HundredGigE 1/0/2.
- Configure traffic policing to allocate bandwidth to traffic from different VLANs, as shown in [Table 1](#).

**Table 1 Bandwidth allocation**

HGE 1/0/1 and HGE 1/0/2 (uplink or downlink)			HGE 1/0/3 (uplink or downlink)			HGE 1/0/10 (uplink or downlink)		
VLAN 1001	VLAN 1002	VLAN 1003	VLAN 201	VLAN 202	VLAN 203	VLAN 201	VLAN 202	VLAN 203

HGE 1/0/1 and HGE 1/0/2 (uplink or downlink)			HGE 1/0/3 (uplink or downlink)			HGE 1/0/10 (uplink or downlink)		
400 Mbps	200 Mbps	200 Mbps	400 Mbps	200 Mbps	200 Mbps	100 Mbps	60 Mbps	40 Mbps

Figure 2 Network diagram



## Analysis

To meet the network requirements, you must perform the following tasks:

- Configure VLAN-based traffic classes.
- Configure per-VLAN traffic policing behaviors.
- Associate each class with its specific traffic behavior.

## Applicable hardware and software versions

The following matrix shows the hardware and software versions to which this configuration example is applicable:

Hardware	Software version
S6850 switch series	Release 6555P01
S9850 switch series	Release 6555P01
S9820-64H switch	Release 6555P01

# Restrictions and guidelines

In a traffic behavior, the traffic policing action cannot be configured together with the following priority marking actions:

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

Otherwise, the class-behavior association for that behavior cannot be applied correctly.

## Procedures

### Configuring VLAN settings

1. Configure HundredGigE 1/0/1 and HundredGigE 1/0/2 as follows:
  - Configure HundredGigE 1/0/1 and HundredGigE 1/0/2 as trunk ports.
  - Assign them to VLANs 1001 through 1003 and VLANs 201 through 203.
  - Remove them from VLAN 1.
  - Configure one-to-one VLAN mappings on the two interfaces.

```
<Device> system-view
[Device] interface hundredgige 1/0/1
[Device-HundredGigE1/0/1] port link-type trunk
[Device-HundredGigE1/0/1] port trunk permit vlan 1001 to 1003 201 to 203
[Device-HundredGigE1/0/1] undo port trunk permit vlan 1
[Device-HundredGigE1/0/1] vlan mapping 1001 translated-vlan 201
[Device-HundredGigE1/0/1] vlan mapping 1002 translated-vlan 202
[Device-HundredGigE1/0/1] vlan mapping 1003 translated-vlan 203
[Device-HundredGigE1/0/1] quit
[Device] interface hundredgige 1/0/2
[Device-HundredGigE1/0/2] port link-type trunk
[Device-HundredGigE1/0/2] port trunk permit vlan 1001 to 1003 201 to 203
[Device-HundredGigE1/0/2] undo port trunk permit vlan 1
[Device-HundredGigE1/0/2] vlan mapping 1001 translated-vlan 201
[Device-HundredGigE1/0/2] vlan mapping 1002 translated-vlan 202
[Device-HundredGigE1/0/2] vlan mapping 1003 translated-vlan 203
[Device-HundredGigE1/0/2] quit
```

2. Configure HundredGigE 1/0/3 and HundredGigE 1/0/10 as follows:
  - Configure HundredGigE 1/0/3 and HundredGigE 1/0/10 as trunk ports.
  - Assign them to VLANs 201 through 203.
  - Remove them from VLAN 1.

```
[Device] interface hundredgige 1/0/3
[Device-HundredGigE1/0/3] port link-type trunk
[Device-HundredGigE1/0/3] port trunk permit vlan 201 to 203
[Device-HundredGigE1/0/3] undo port trunk permit vlan 1
```

```

[Device-HundredGigE1/0/3] quit
[Device] interface hundredgige 1/0/10
[Device-HundredGigE1/0/10] port link-type trunk
[Device-HundredGigE1/0/10] port trunk permit vlan 201 to 203
[Device-HundredGigE1/0/10] undo port trunk permit vlan 1
[Device-HundredGigE1/0/10] quit

```

## Configuring traffic policing

1. Configure traffic policing for the traffic from and to branches:

**# Create a class named `vlan201`, and configure CVLAN 201 as the match criterion.**

```

[Device-classifier-vlan201] if-match customer-vlan-id 201
[Device-classifier-vlan201] quit

```

**# Create a class named `vlan202`, and configure CVLAN 202 as the match criterion.**

```

[Device] traffic classifier vlan202
[Device-classifier-vlan202] if-match customer-vlan-id 202
[Device-classifier-vlan202] quit

```

**# Create a class named `vlan203`, and configure CVLAN 203 as the match criterion.**

```

[Device] traffic classifier vlan203
[Device-classifier-vlan203] if-match customer-vlan-id 203
[Device-classifier-vlan203] quit

```

**# Create a behavior named `car400`, and configure a CIR of 400 Mbps.**

```

[Device] traffic behavior car400
[Device-behavior-car400] car cir 409600
[Device-behavior-car400] quit

```

**# Create a behavior named `car200`, and configure a CIR of 200 Mbps.**

```

[Device] traffic behavior car200
[Device-behavior-car200] car cir 204800
[Device-behavior-car200] quit

```

**# Create a QoS policy named `ABCupdown`, and associate the classes with the behaviors.**

```

[Device] qos policy ABCupdown
[Device-qospolicy-ABCupdown] classifier vlan201 behavior car400
[Device-qospolicy-ABCupdown] classifier vlan202 behavior car200
[Device-qospolicy-ABCupdown] classifier vlan203 behavior car200
[Device-qospolicy-ABCupdown] quit

```

**# Apply the QoS policy to both directions of HundredGigE 1/0/1, HundredGigE 1/0/2, and HundredGigE 1/0/3.**

```

[Device] interface hundredgige 1/0/1
[Device-HundredGigE1/0/1] qos apply policy ABCupdown inbound
[Device-HundredGigE1/0/1] qos apply policy ABCupdown outbound
[Device-HundredGigE1/0/1] quit
[Device] interface hundredgige 1/0/2
[Device-HundredGigE1/0/2] qos apply policy ABCupdown inbound
[Device-HundredGigE1/0/2] qos apply policy ABCupdown outbound
[Device-HundredGigE1/0/2] quit
[Device] interface hundredgige 1/0/3
[Device-HundredGigE1/0/3] qos apply policy ABCupdown inbound
[Device-HundredGigE1/0/3] qos apply policy ABCupdown outbound

```

```
[Device-HundredGigE1/0/3] quit
```

2. Configure traffic policing for the traffic from and to the backbone network:

# Create a behavior named **car100**, and configure a CIR of 100 Mbps.

```
[Device] traffic behavior car100
[Device-behavior-car100] car cir 102400
[Device-behavior-car100] quit
```

# Create a behavior named **car60**, and configure a CIR of 60 Mbps.

```
[Device] traffic behavior car60
[Device-behavior-car60] car cir 61440
[Device-behavior-car60] quit
```

# Create a behavior named **car40**, and configure a CIR of 40 Mbps.

```
[Device] traffic behavior car40
[Device-behavior-car40] car cir 40960
[Device-behavior-car40] quit
```

# Create a QoS policy named **BONEupdown**, and associate the classes with the behaviors.

```
[Device] qos policy BONEupdown
[Device-qospolicy-BONEupdown] classifier vlan201 behavior car100
[Device-qospolicy-BONEupdown] classifier vlan202 behavior car60
[Device-qospolicy-BONEupdown] classifier vlan203 behavior car40
[Device-qospolicy-BONEupdown] quit
```

# Apply the QoS policy to both directions of HundredGigE 1/0/10.

```
[Device] interface hundredgige 1/0/10
[Device-HundredGigE1/0/10] qos apply policy BONEupdown inbound
[Device-HundredGigE1/0/10] qos apply policy BONEupdown outbound
[Device-HundredGigE1/0/10] quit
```

Figure 3 shows how the switches process the uplink traffic from a branch to the backbone network. The figure uses VLAN 1001 as an example.

**Figure 3 Uplink traffic processing**

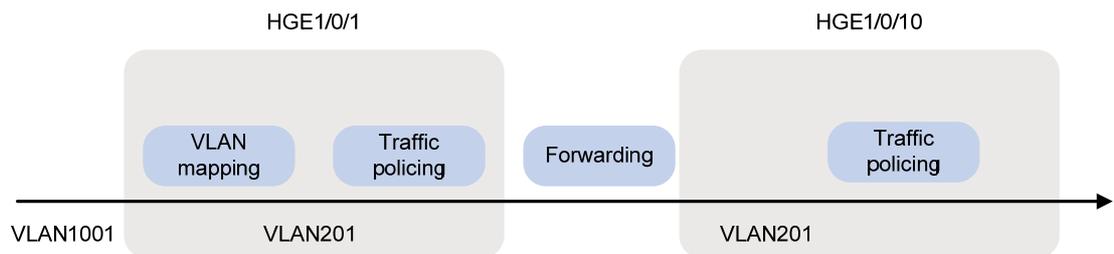
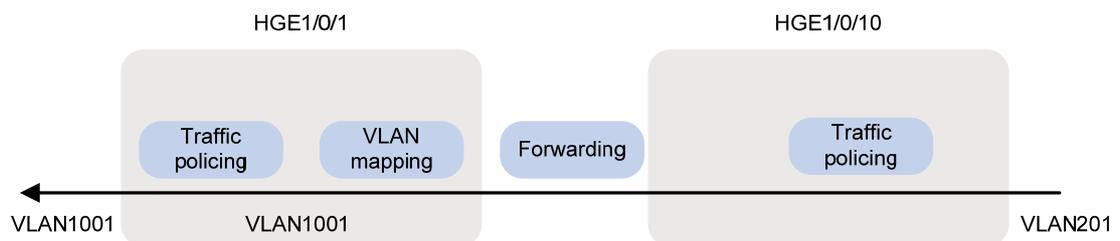


Figure 4 shows how the switches process the downlink traffic from the backbone network to a branch. The figure uses VLAN 201 as an example.

**Figure 4 Downlink traffic processing**



# Verifying the configuration

Verify the configuration on any interface, for example, HundredGigE 1/0/10.

# Verify QoS policies applied to interface HundredGigE 1/0/10.

```
[Device] display qos policy interface hundredgige 1/0/10
```

```
Interface: HundredGigE1/0/10
```

```
Direction: Inbound
```

```
Policy: BONEupdown
```

```
Classifier: vlan201
```

```
Operator: AND
```

```
Rule(s) :
```

```
  If-match customer-vlan-id 201
```

```
Behavior: car100
```

```
Committed Access Rate:
```

```
CIR 102400 (kbps), CBS 6400000 (Bytes), EBS 0 (Bytes)
```

```
Green action : pass
```

```
Yellow action : pass
```

```
Red action   : discard
```

```
Green packets : 0 (Packets)
```

```
Red packets  : 0 (Packets)
```

```
Classifier: vlan202
```

```
Operator: AND
```

```
Rule(s) :
```

```
  If-match customer-vlan-id 202
```

```
Behavior: car60
```

```
Committed Access Rate:
```

```
CIR 61440 (kbps), CBS 3840000 (Bytes), EBS 0 (Bytes)
```

```
Green action : pass
```

```
Yellow action : pass
```

```
Red action   : discard
```

```
Green packets : 0 (Packets)
```

```
Red packets  : 0 (Packets)
```

```
Classifier: vlan203
```

```
Operator: AND
```

```
Rule(s) :
```

```
  If-match customer-vlan-id 203
```

```
Behavior: car40
```

```
Committed Access Rate:
```

```
CIR 40960 (kbps), CBS 2560000 (Bytes), EBS 0 (Bytes)
```

```
Green action : pass
```

```
Yellow action : pass
```

```
Red action   : discard
```

```
Green packets : 0 (Packets)
```

```
Red packets  : 0 (Packets)
```

```
Interface: HundredGigE1/0/10
```

```
Direction: Outbound
```

```
Policy: BONEupdown
```

```
Classifier: vlan201
```

```

Operator: AND
Rule(s) :
  If-match customer-vlan-id 201
Behavior: car100
Committed Access Rate:
  CIR 102400 (kbps), CBS 6400000 (Bytes), EBS 0 (Bytes)
  Green action : pass
  Yellow action : pass
  Red action : discard
  Green packets : 0 (Packets)
  Red packets : 0 (Packets)
Classifier: vlan202
Operator: AND
Rule(s) :
  If-match customer-vlan-id 202
Behavior: car60
Committed Access Rate:
  CIR 61440 (kbps), CBS 3840000 (Bytes), EBS 0 (Bytes)
  Green action : pass
  Yellow action : pass
  Red action : discard
  Green packets : 0 (Packets)
  Red packets : 0 (Packets)
Classifier: vlan203
Operator: AND
Rule(s) :
  If-match customer-vlan-id 203
Behavior: car40
Committed Access Rate:
  CIR 40960 (kbps), CBS 2560000 (Bytes), EBS 0 (Bytes)
  Green action : pass
  Yellow action : pass
  Red action : discard
  Green packets : 0 (Packets)
  Red packets : 0 (Packets)

```

## Configuration files

```

#
traffic classifier vlan201 operator and
  if-match customer-vlan-id 201
#
traffic classifier vlan202 operator and
  if-match customer-vlan-id 202
#
traffic classifier vlan203 operator and
  if-match customer-vlan-id 203
#

```

```

traffic behavior car40
  car cir 40960 cbs 2560000 ebs 0 green pass red discard yellow pass
#
traffic behavior car60
  car cir 61440 cbs 3840000 ebs 0 green pass red discard yellow pass
#
traffic behavior car100
  car cir 102400 cbs 6400000 ebs 0 green pass red discard yellow pass
#
traffic behavior car200
  car cir 204800 cbs 12800000 ebs 0 green pass red discard yellow pass
#
traffic behavior car400
  car cir 409600 cbs 25600000 ebs 0 green pass red discard yellow pass
#
qos policy ABCupdown
  classifier vlan201 behavior car400
  classifier vlan202 behavior car200
  classifier vlan203 behavior car200
#
qos policy BONEupdown
  classifier vlan201 behavior car100
  classifier vlan202 behavior car60
  classifier vlan203 behavior car40
#
interface HundredGigE1/0/10
  port link-mode bridge
  port link-type trunk
  undo port trunk permit vlan 1
  port trunk permit vlan 201 to 203
  qos apply policy BONEupdown inbound
  qos apply policy BONEupdown outbound
#
interface HundredGigE1/0/1
  port link-mode bridge
  port link-type trunk
  undo port trunk permit vlan 1
  port trunk permit vlan 201 to 203 1001 to 1003
  vlan mapping 1001 translated-vlan 201
  vlan mapping 1002 translated-vlan 202
  vlan mapping 1003 translated-vlan 203
  qos apply policy ABCupdown inbound
  qos apply policy ABCupdown outbound
#
interface HundredGigE1/0/2
  port link-mode bridge
  port link-type trunk
  undo port trunk permit vlan 1

```

```

port trunk permit vlan 201 to 203 1001 to 1003
vlan mapping 1001 translated-vlan 201
vlan mapping 1002 translated-vlan 202
vlan mapping 1003 translated-vlan 203
qos apply policy ABCupdown inbound
qos apply policy ABCupdown outbound
#
interface HundredGigE1/0/3
port link-mode bridge
port link-type trunk
undo port trunk permit vlan 1
port trunk permit vlan 201 to 203
qos apply policy ABCupdown inbound
qos apply policy ABCupdown outbound
#

```

## Example: Configuring aggregate CAR

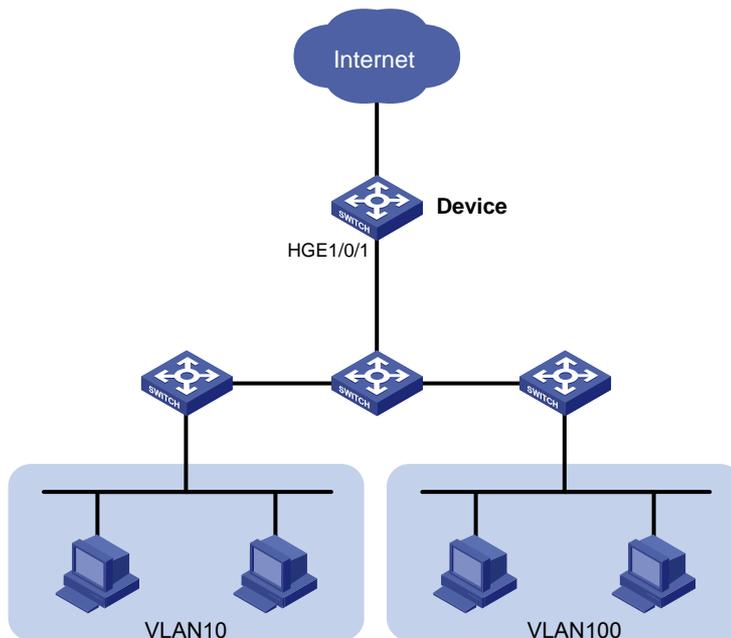
### Network configuration

As shown in [Figure 5](#), the access layer devices add VLAN tags to the traffic from VLAN 10 and VLAN 100 before sending the traffic to the device.

Configure aggregate CAR on HundredGigE 1/0/1 to meet the following requirements:

- Limit the incoming traffic from VLAN 10 and VLAN 100 to 200 Mbps.
- Drop the excess traffic.

**Figure 5 Network diagram**



# Applicable hardware and software versions

The following matrix shows the hardware and software versions to which this configuration example is applicable:

Hardware	Software version
S6850 switch series S9850 switch series	Release 6555P01
S9820-64H switch	Release 6555P01

## Procedures

# Configure interface HundredGigE 1/0/1 as a trunk port.

```
<Device> system-view
[Device] interface hundredgige 1/0/1
[Device-HundredGigE1/0/1] port link-type trunk
```

# Assign the interface to VLANs 10 and 100.

```
[Device-HundredGigE1/0/1] port trunk permit vlan 10 100
```

# Remove the interface from VLAN 1.

```
[Device-HundredGigE1/0/1] undo port trunk permit vlan 1
[Device-HundredGigE1/0/1] quit
```

# Create an aggregate CAR action.

```
[Device] qos car aggcar-1 aggregative cir 204800
```

# Create class 1, and use SVLAN ID 10 as the match criterion.

```
[Device] traffic classifier 1
[Device-classifier-1] if-match service-vlan-id 10
[Device-classifier-1] quit
```

# Create behavior 1, and reference the aggregate CAR action in the behavior.

```
[Device] traffic behavior 1
[Device-behavior-1] car name aggcar-1
[Device-behavior-1] quit
```

# Create class 2, and use SVLAN ID 100 as the match criterion.

```
[Device] traffic classifier 2
[Device-classifier-2] if-match service-vlan-id 100
[Device-classifier-2] quit
```

# Create behavior 2, and reference the aggregate CAR action in the behavior.

```
[Device] traffic behavior 2
[Device-behavior-2] car name aggcar-1
[Device-behavior-2] quit
```

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

```
[Device] qos policy car
[Device-qospolicy-car] classifier 1 behavior 1
[Device-qospolicy-car] classifier 2 behavior 2
[Device-qospolicy-car] quit
```

```
# Apply the QoS policy car to the inbound direction of HundredGigE 1/0/1.
[Device] interface hundredgige 1/0/1
[Device-HundredGigE1/0/1] qos apply policy car inbound
```

## Verifying the configuration

Verify the configuration on any interface, for example, HundredGigE 1/0/1.

```
# Verify QoS policies applied to interface HundredGigE 1/0/1.
[Device] display qos policy interface hundredgige 1/0/1
Interface: HundredGigE1/0/1
  Direction: Inbound
  Policy: car
  Classifier: 1
    Operator: AND
    Rule(s) :
      If-match service-vlan-id 10
    Behavior: 1
      Committed Access Rate:
        Car name: aggcar-1
  Classifier: 2
    Operator: AND
    Rule(s) :
      If-match service-vlan-id 100
    Behavior: 2
      Committed Access Rate:
        Car name: aggcar-1
```

## Configuration files

```
#
  qos car aggcar-1 aggregative cir 204800 cbs 12800000 ebs 0 green pass yellow pass
  red discard
#
  traffic classifier 1 operator and
  if-match service-vlan-id 10
  traffic classifier 2 operator and
  if-match service-vlan-id 100
#
  traffic behavior 1
  car name aggcar-1
  traffic behavior 2
  car name aggcar-1
#
  qos policy car
  classifier 1 behavior 1
  classifier 2 behavior 2
#
  interface HundredGigE1/0/1
```

```
port link-mode bridge
port link-type trunk
undo port trunk permit vlan 1
port trunk permit vlan 10 100
qos apply policy car inbound
```

## Related documentation

- *H3C S6850 & S9850 Switch Series ACL and QoS Configuration Guide-Release 655x*
- *H3C S6850 & S9850 Switch Series ACL and QoS Command Reference-Release 655x*
- *H3C S9820-64H Switch ACL and QoS Configuration Guide-Release 655x*
- *H3C S9820-64H Switch ACL and QoS Command Reference-Release 655x*