QOS

QSR-1920, QSR-2920, QSR-3920

Оглавление

1. QOS

1.1. Overview

1.1.1. Background

In the traditional IP network, the forwarding device treats all packets equally, adopts “First in, first out” (FIFO) to process all packets and tries best effort to transmit the packet to the destination, so it cannot provide any guarantee for the reliability and delay of the packet transmission.

However, with the development of the IP network, the new applications based on the IP network emerge in endlessly, which put forward new requirements for the service quality of the IP network, especially the demand for the service packets with high real-time requirement is more obvious. For example, the network flow media, VoIP and other real-time services put forward high requirement for the transmission delay of the packets. If the packet transmission delay is long, the user cannot accept (relatively, E-mail and FTP services are not sensitive to the transmission delay). To support the communication services with different service quality requirements, it is required that the network can intelligently distinguish different communication types, so as to provide the corresponding service. The capability of distinguishing the communication types is the basic premise of providing different service qualities for different communications, so the best-effort service mode of the traditional IP network cannot meet the requirements of the present IP network application. The QoS (Quality of Service) technology is to solve the problem, so as to meet the different service quality requirements of the users for the network.

1.1.2. Service Model

QoS provides the following three kinds of service models, that is, Best-Effort service, Integrated service, and Differentiated service (DiffServ for short).

Best-Effort is a single service model and also the simplest service model. The application program can send out any quantity of packets at any time without getting the permission or informing the network in advance. For the best-effort service, the network tries best to send the packets, but does not provide ant guarantee for the transmission delay and reliability of the packets. Best-Effort is the default service model of Internet and is applicable to most of network applications, such as FTP and E-Mail. It is realized via the FITO queue mechanism.

IntServ is one service model that can provide various service types. It can meet various QoS requirements. Before sending packets, the service model needs to apply for the specified service resources from the network. The request is completed via the RSVP signaling. RSVP applies for the network resources for the application before the application program starts to send packets, so it belongs to the out-band signaling. Before sending data, the application program first informs the network of its own traffic parameters and the needed specified service quality request, including bandwidth, delay and so on. After receiving the resource request of the application program, the network executes the resource distributing check, that is, judge whether to distribute resources for the application program based on the resource application of the application program and the present resources of the network. Once the network confirms to distribute resources for the application program, the network maintains one state for the specified flow (Flow, confirmed by the IP addresses, port numbers and protocol numbers of the two sides) and executes the packet classification, traffic monitoring, queuing and scheduling based on the state. After receiving the confirming information of the network (that is, confirm that the network already reserves resources for the packets of the application program), the application program can send packets. As long as the packets of the application program are controlled within the range described by the traffic parameters, the network will undertake to meet the QoS requirements of the application program.

DiffServ classifies the communications according to the service requirements, and then processes the ingress and egress packets according to the classification result, so as to ensure that the network is always in the good communication connection status. It is one multi-channel service model and can meet the QoS requirements of different flows. The largest difference with IntServ is that DiffServ can reserve resources in the network without signaling exchange. It just functions on one port of one transmission device in the network, processing the ingress and egress packets of the port. DiffServ does not need to maintain the status information for each kind of communication. It distinguishes the QoS level of each packet according to the configured QoS mechanism and provides the service for the packet according to the level. Therefore, the mechanism providing the QoS scheme is also called CoS. There are many classification methods and the common modes are to classify according to the priority of the IP packet, classify according to the source, destination address and port of the packet, classify according to the packet protocol, classify according to the packet size and packet ingress port, and so on.

Priority mapping, flow classification, traffic monitoring, traffic shaping, congestion management and congestion avoidance are the main components of DiffServ. The flow classification identifies the packets according to some matching rules and is the basis and premise of DiffServ; traffic monitoring, traffic shaping, congestion management and congestion avoidance distribute and schedule the resources for the network traffic from different aspects and they are the embodiment of the DiffServ idea.

1.1.3. Introduction to QoS Functions

Flow Classification and Marking

Flow classification adopts some rule to identify the packets that comply with one feature, divides the packets of different features to multiple classes, and then uses the corresponding QoS mechanism to provide different services for different classes. Therefore, the flow classification is the premise and basis of providing different services.

The flow classification rule can use the priority field of the packet, including IP DSCP, 802.1P, and MPLS-EXP priority, to identify the traffic with different priority features; the network manager also can set the policy of the flow classification. For example, integrate the source address, destination address, MAC address, IP protocol or port number of the application program to classify the flow. The general classification basis is limited to the header information for encapsulating the packet. Using the packet content as the classification standard is relatively rare. The classification result does not have the range limitation and it can be the small range determined by the quintuple (source address, source port number, protocol number, destination address, destination port number) and also can be all packets to one segment. When classifying the packets at the network edge, set the priority in the ToS field of the packet IP header. In this way, we can directly use the IP priority as the classification standard inside the network. However, the queue technology also can use the priority to perform different processing for the packets. The downstream network can select to accept the classification result of the upstream network, and also can re-classify according to its own standard. The flow classification is to provide differentiated services. It should be associated with one flow control or resource distribution action. Which flow control action to adopt depends on the stage and the current load condition of the network. For example, when the packet enters the network, monitor it according to the CIR; shape before flowing out of the node; during congestion, perform the queue scheduling management; when the congestion is intensified, adopt the congestion avoidance measures and so on.

Flow marking means to set or modify the attributes of one kind of packets. After classifying the packets to different types by the flow classification, the flow marking can modify the attributes of one flow packet to prepare for the subsequent processing of the flow packet. For example, we can specify to modify the DSCP of the packet matching one flow classification rule to 5, while

specify the DSCP of the packet matching another classification rule as 0. Distinguish the packets by traffic marking. Perform the differentiated processing according to the DSCP value of the packet in the subsequent operations. For example, first send the packet with the DSCP tag 5. When the congestion happens, first drop the packet with DSCP 0.

The following attributes can perform the traffic marking:

1. The CoS value of the packet
2. The DSCP value in the IP packet header
3. The EXP value of the MPLS packet
4. The Precedence field in the ToS domain of the IP packet
5. QoS Group ID (valid only for the device)

Traffic Monitoring

Traffic monitoring limits the rate for the packets entering the interface by the token bucket and perform various operations for the packets within the committed access rate and exceeding the rate, including sending, dropping and modifying the IP DSCP, IP priority of the packet or marking QoS private attribute and so on. Its typical application is to limit the traffic entering one network. Usually, drop the data exceeding the rate or reduce its priority. The following figure is the general processing flow of the traffic monitoring:

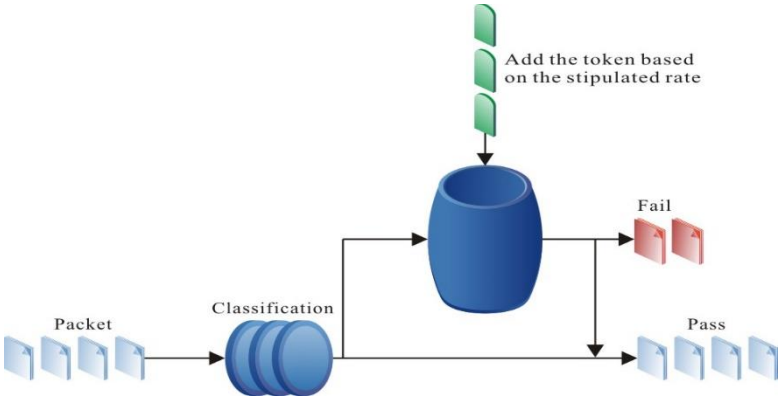

Figure 1-1The general processing flow of the traffic monitoring

Token bucket is one common traffic measuring method. It has two parameters, that is, Burst size and Mean rate, used to report whether the packet matches with the configured rate parameter, but as one measuring tool, the token bucket does not filter, change or take one measure for the communication.

Token bucket places the token according to the rate set by the user. Besides, the user also can set the capacity of the token bucket. When the number of the tokens in the token bucket exceeds the capacity of the bucket, the tokens in the token bucket overflows and the token quantity does not increase any more. When adopting the token bucket to process the packet and if there are tokens in the token bucket, the packets can pass the bucket (the token quantity can be negative, which is called token borrowing). Meanwhile, the tokens in the token bucket reduce according to the length of the packet; when the tokens in the token bucket are 0 (or negative), the packet cannot pass the bucket.

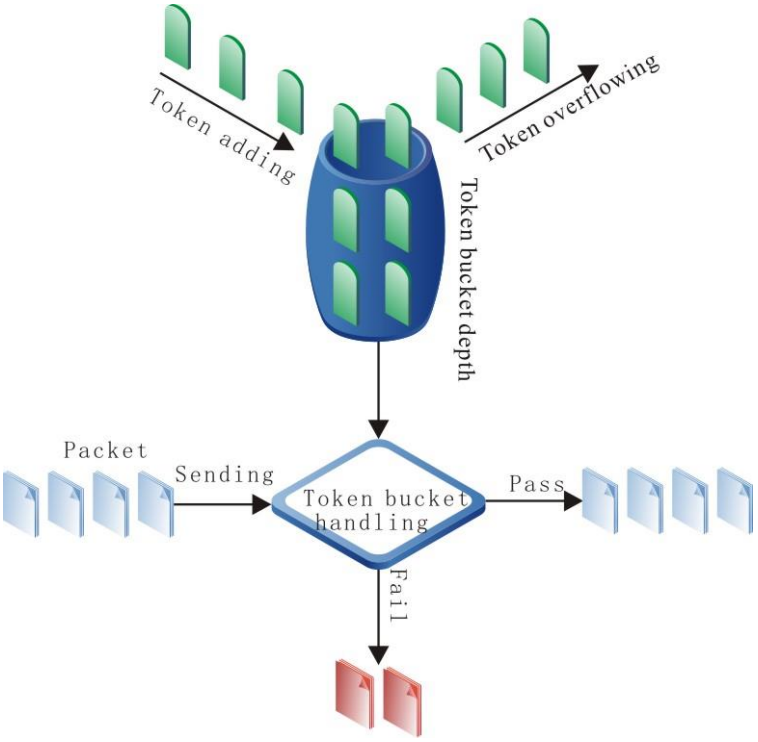

Figure 1-2 Typical token bucket processing flow

The green box in the above figure indicates the token, the gray box indicates the data flow, and the small bucket is used to hold the tokens. The meaning of the algorithm is: When passing one unit of data flow (such as 1 Byte) from the token bucket, consume one token in the bucket. When there are tokens in the token bucket, the packet is marked as green. Otherwise, it is marked as red. When the data flow does not use the bandwidth distributed to it, the token bucket accumulates the tokens and here, the packets indicated by all tokens in the bucket can be sent. This can permit the burst transmission of the data. When there is no token in the token bucket, the packet cannot be sent. Only when there is new token generated in the bucket, the packet can be sent. In this way, we can limit the packet traffic to be smaller than or equal to the generating rate of the token, so as to limit the traffic.

The token bucket has three mechanisms of assessing the rate, that is, single-bucket single-speed, double-bucket single-speed, and double-bucket double-speed:

- Single-bucket single-speed token bucket: Use one token bucket to measure whether the data comply with or violate the rule. The capacity of the bucket is Bc. If the data does not make the tokens in the token bucket smaller than 0, the data complies with the rule. Otherwise, the overflow data is the violation data.
- Double-bucket single-speed token bucket: Use two token buckets to measure data and only define CIR. The capacity of the first bucket is Bc. If the data does not overflow the first bucket, the data complies with the rule; the capacity of the second bucket is Be. When the data of the first bucket overflows and the data of the second bucket does not overflow, the data becomes the data exceeding the rule; if the data flow is too large and overflows the second bucket, the overflow data is the data violating the rule.
- Double-bucket double-speed token bucket: Define two rates, that is, CIR and PIR. Meanwhile, use two token buckets to measure data. The capacity of the first token bucket is Bc, used to measure whether the data complies with or exceed CIR. If not making the tokens in the first token bucket be smaller than 0, the data complies with the rule. Otherwise, the data exceeds the rule; the capacity of the second token bucket is

Be, used to measure whether the data rate is larger than PIR. When the data makes the tokens in the second token bucket be smaller than 0, the overflow data violates the rule.

Traffic Shaping

The typical function of the traffic shaping means to limit the traffic of flowing out from one network, making the packets be sent with an average rate. Usually, it is divided to the port traffic shaping and queue traffic shaping. When the sending rate of the packets exceeds the shaping rate, the speeding packets are buffered in the queue and then are sent out with an average rate. The difference between the traffic shaping and traffic monitoring: When using the traffic monitoring to control the packet traffic, the speeding packets are not buffered, but are directly dropped, while the traffic shaping buffers the speeding packets, reducing the dropped packets caused by the burst traffic. However, the traffic shaping may increase the delay, while the traffic monitoring nearly does not increase the delay. The following figure is the general processing flow of the traffic shaping:

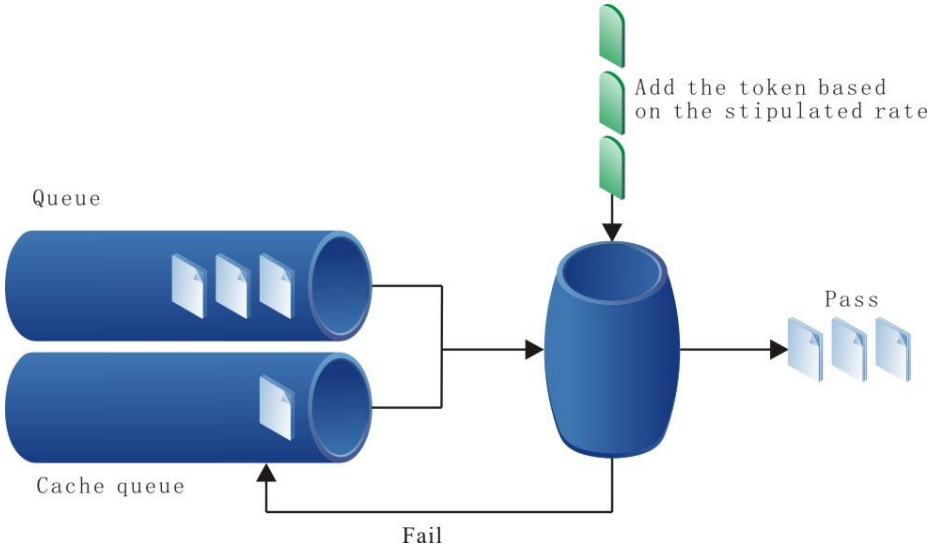

Figure 1-3 The general processing flow of the traffic shaping

Congestion Management

When the device traffic load is light, do not generate the congestion and the packets are forwarded out when reaching the port. When the arriving rate of the packets is larger than the sending rate of the port and exceeds the processing limit of the port or the device resources are not enough, congestion happens to the device. The congestion may make the communication of the whole network become unreliable. The end-to-end delay, jitter and packet loss rate used to measure the network service quality all increase. If enabling the congestion management and when the congestion happens, the packets queue at the port and waits for the port to forward. The congestion management usually adopts the queue technology and the port determines which queue the packet should be placed according to the packet priority and queue mechanism and how to schedule and forward packets. The common scheduling includes FIFO, FQ, PQ, CBWFQ, SPQ and so on.

FIFO is the default queue of the interface in the system, processing the packets by “First In, First Out”.

FQ (Fair Queuing): It is put forward to share the network resources fairly and try to make the delay and jitter of all flows reach best. Classify the packets by flow and each flow is distributed with one queue. Each queue schedules by turns fairly.

PQ is short for priority queue, that is, high (high priority) queue, medium (medium priority) queue, normal (normal priority) queue and low (low priority) queue. The default queue is the normal

queue. The packets in the four queues are scheduled and sent according to the strict priority. When there are packets in the high queue to be sent, the packets in the medium queue cannot be scheduled or sent until the high queue is empty, and so on. Obviously, the “starve to death” phenomenon may happen to the low queue in PQ, that is, it cannot be scheduled.

CBWFQ is class-based weighted fair queue. It lets the user define the data class according to the communication parameters (priority, access list, protocol type and so on) and distribute one queue for each data class, and then set the occupied bandwidth, drop policy for each queue and perform the traffic monitoring and shaping. The queue weight is the bandwidth value of the queue and also the bandwidth that can be used by the queue when the interface is congested. Currently, CBWFQ supports LLQ, Bandwidth and BE queues. The default data enters the BE queue.

SPQ is a strict priority queue. The queue priority is set according to the user's DSCP value for packets. The packets without priority enter the default priority queue (the lowest priority). The queue is scheduled strictly from high priority to low priority. Each queue can be set by the user to limit the use bandwidth.

Congestion Avoidance

The congestion avoidance technology monitors the communication load of the network, so as to avoid the congestion before the network congestion happens. The common used technology is RED (Random Early Detection). The difference with the tail drop method is that RED drops the data in the queue at random. Based on the RED algorithm, generate WRED (Weighted Random Early Detection). It selects the dropped packet according to the DSCP or IP priority and can provide different performance features for different service types of data. It also can avoid the TCP global synchronization.

In the WRED algorithm, the dropped packets depend on the average length, minimum threshold and maximum threshold of the queue. When the average length of the queue is smaller than the minimum threshold, do not drop the packets; when the average length of the queue is larger than the minimum threshold and smaller than the maximum threshold, the packet dropping rate increases with the average queue length; at last, when the average length of the queue is larger than the maximum threshold, drop all the packets. The formula of calculating the average length of the queue is: the average length of the queue = (previous average queue length × (1-1 / (2 ^ n))) + (current queue length × (1 / (2 ^ n))). Here, n can be configured by the command. The average length of the queue reflects the change trend of the queue and is not sensitive with the burst change of the queue length, avoiding the unfair treat for the burst data flow.

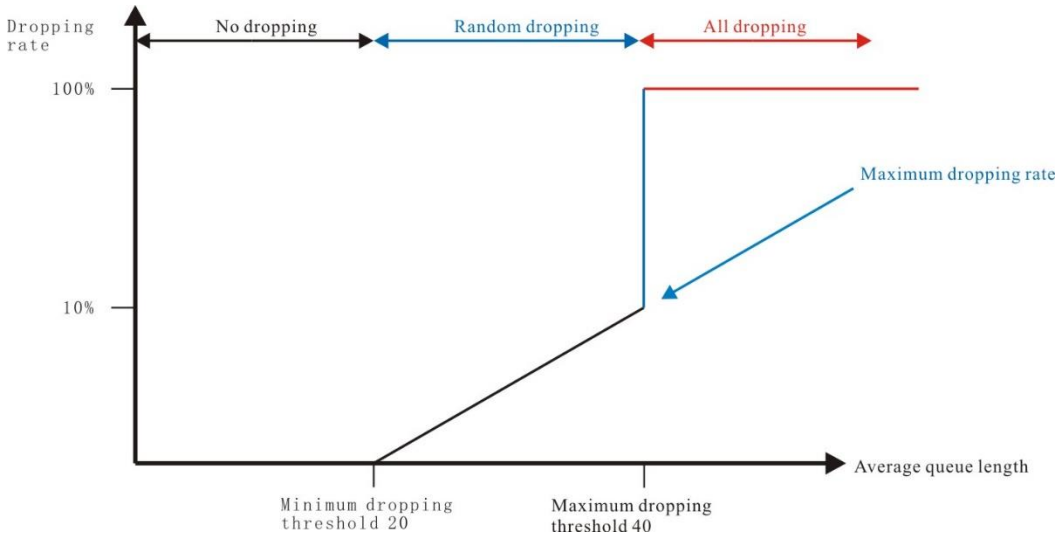

Figure 1-4 WRED diagram

1.2. QoS Function Configuration

Table 1-1 QoS function configuration list

Configuration Task	
Configure the traffic monitoring	Configure the interface-based CAR
	Configure the flow-based CAR
Configure the traffic shaping	Configure the general traffic shaping
Configure the congestion management	Configure the QoS used bandwidth
	Configure the FIFO queue
	Configure the FQ queue
	Configure the PQ queue
	Configure the CBWFQ queue
	Configure the SPQ queue
Configure the congestion avoidance	Configure the WRED
Configure the sub interface re-direction	Configure the sub interface re-direction

1.2.1. Configure Traffic Monitoring

Configuration Condition

None

Configure Interface-based CAR

Provide the rate limitation and marking at the ingress and egress directions for all IP and MPLS packets on the interface. It is used on the network edge device, limiting the traffic rate of entering or leaving the device. We can configure multiple CAR policy on the interface. When the packet passes, check every policy one by one until the packet matches with one policy; if not matching any policy, the packet is directly forwarded.

Table 1-2 Configure the interface-based CAR

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the interface-based CAR	**rate-limit** { **input | output** } *cir conform-brust exceed-brust* **conform-action** { { **continue** | **drop** | **transmit** | **set-qos-transmit** | **set-qos-continue** | **set-prec-transmit** | **set-prec-continue | set-dscp-transmit | set-dscp-continue** | **set-mpls-exp-imposition-transmit** | **set-mpls-exp- imposition -continue** } [*action-value*] } **exceed-action** { { **continue** | **drop** | **transmit** | **set-qos-transmit** | **set-qos-continue** | **set-prec-transmit** | **set-prec-continue | set-dscp-transmit | set-dscp-continue** | **set-mpls-exp-imposition-transmit** | **set-mpls-exp-imposition -continue** } [*action-value*] } [**color-keep**]	Mandatory By default, do not configure the interface-based CAR. For details, refer to the related commands of the QoS command manual.

Note:

- **color-keep** is non-color-blind mode. If the packet is colored by the previous CAR polict, keep the color.
- The configured token bucket parameters of CAR BC (*conform-brust*) and BE (*exceed-brust*) indicate the layer-1 and layer-2 bucket lengths respectively. The BE in the CAR parameters of some friend-manufacturers' devices indicates the sum of layer-1 and layer-2 bucket depths. Usually, it is suggested to configure BC as rate (*cir*)/8/2.

Configure Flow-based CAR

After classifying the packets on the interface, limit the rate and mark at the egress and ingress directions. It is used on the network edge devices, limiting the traffic rate of entering or leaving the device. We can configure multiple CAR policies on the interface. When the packet passes, check every policy one by one until the packet matches with one policy; if not matching any policy, the packet is directly forwarded. Currently, the classification matching conditions on CAR include QOS Group ID, ACL, MPLS-EXP and so on.

Table 1-3 Configure the flow-based CAR

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the flow-based CAR	**rate-limit** { **input | output** } [[**access-group** *access-list*] | [**mpls experimental topmost** *exp*] | [**qos-group** *group-val*]] *cir conform-brust exceed-brust* **conform-action** {{ **continue** | **drop** | **transmit** | **set-qos-transmit** | **set-qos-continue** | **set-prec-transmit** | **set-prec-continue | set-dscp-transmit | set-dscp-continue** | **set-mpls-exp-imposition-transmit** | **set-mpls-exp-imposition -continue** } [*action-value*] } **exceed-action** { { **continue** | **drop** | **transmit** | **set-qos-transmit** | **set-qos-continue** | **set-prec-transmit** | **set-prec-continue | set-dscp-transmit | set-dscp-continue** | **set-mpls-exp-imposition-transmit** | **set-mpls-exp-imposition -continue** } [*action- value*] } [**color-keep**]	Mandatory By default, do not configure CAR matching ACL. Match all packets of IP and MPLS, QoS Group ID, MPLS-EXP CAR. For details, refer to the related commands of the QoS command manual.

Note:

- CAR cannot configure matching QoS Group ID at the ingress direction of the interface; at the egress direction of the interface, we cannot mark Qos Group ID.
- CAR cannot configure **set-mpls-exp-imposition-transmit** and **set-mpls-exp-imposition-continue** at the egress direction of the interface.
- Configuring **color-keep** means that the CAR is non-color-bind mode. If the packet is colored by the previous CAR policy, keep the color.
- The configured token bucket parameters of CAR BC (*conform-brust*) and BE (*exceed-brust*) indicate the layer-1 and layer-2 bucket lengths respectively. The BE in the CAR parameters of some friend-manufacturers' devices indicates the sum of layer-1 and layer-2 bucket depths. Usually, it is suggested to configure BC as rate (*cir*)/8/2.

1.2.2. Configure Traffic Shaping

Configuration Condition

None

Configure General Traffic Shaping

General traffic shaping (GTS) is configured on the interface, performing the traffic shaping control for the packets at the egress direction of the interface.

Table 1-4 Configure the general traffic shaping

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the specified interface	**interface** *interface-name*	-
Configure the general traffic shaping	**traffic-shape** *conform-rate* [*permit-sub-burst*]	Mandatory By default, do not configure the general traffic shaping. If not specifying *permit-sub-burst*, it is the traffic passing by the rate *conform-rate* within 25ms by default (that is *conform-rate**25/1000)

1.2.3. Configure Congestion Management

The default adopted congestion management policy of the interface is the FIFO queue, treating the packets entering the packets fairly by the principle of “First in, First out”.

Configuration Condition

None

Configure QoS Used Bandwidth

By default, the QoS used bandwidth adopts the logical bandwidth of the interface, corresponding to the value configured by the **bandwidth** command on the interface. We also can adopt the **qos max-bandwidth** command to specify the QoS used bandwidth.

Table 1-5 Configure the QoS used bandwidth

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the QoS used bandwidth	**qos max-bandwidth** *bandwidth*	Mandatory By default, the QoS used bandwidth adopts the logical bandwidth of the interface.

Note:

- After configuring the QoS used bandwidth, the bandwidth calculation of the QoS queue in the interface takes the QoS used bandwidth as reference.
- QoS used bandwidth needs to be set to be consistent with the actual available bandwidth of the physical interface. For example, the default bandwidth of some logical interface is smaller than the associated physical interface (for example, the default bandwidth of the tunnel interface is 10k and the associated physical interface is larger than the value). To work normally, we need to set the QoS used bandwidth to be consistent with the actual available of the physical interface.

Configure FIFO Queue

FIFO does not classify the packets. When the packets enter the interface, FIFO always lets the packet enter the queue by the order of the packets reaching the interface. Meanwhile, FIFO lets the packets leave the queue by the order of the packets entering the queue. The packet first entering the queue leaves the queue first and the packet entering the queue later leaves the queue later.

FIFO queue is used on the interface by default and does not need to be configured. Besides, the user can configure the FIFO queue depth according to the packet cache demand, as follows:

Table 1-6 Configure the FIFO queue depth

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the FIFO queue depth	**hold-queue** *queue-length* **out**	Optional By default, FIFO queue depth is 256 packets.

Configure FQ Queue

FQ classifies the packets by flow. For the IP packets, perform the hash algorithm according to the IP quintuple, including the source IP address, destination IP address, source port number, destination port number and protocol number, to hash different flows to different queues. For MPLS packets, perform hash algorithm according to some feature information of the MPLS header. The FQ queue quantity is configured by the user and the queues schedule by turns fairly.

Table 1-7 Configure the FQ queue

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Enable FQ on the interface	**fair-queue** [*queue-num*]	Mandatory The interface enables FQ. The optional parameter *queue-num* is used to specify the number of the FQ queues. If not specifying, there are 32 queues by default. By default, do not enable FQ on the interface.

Step	Command	Description
Configure the FQ queue depth	**fair-queue queue-limit** *queue-limit-value*	Optional By default, the FQ queue depth is 256 packets.

Configure PQ Queue

The application of the PQ policy: The user classifies the packets by configuring the IP or ingress interface classification rule as desired, and then the packets enter the PQ queues, that is, high queue, medium queue, normal queue, and low queue. Their priorities reduce in turn. By default, the data flow enters the normal queue. After the flow classification, the packets enter the corresponding queues. When scheduling to leave the queue, PQ strictly complies with the priority order from high to low, first sending the packets in the high-priority queue. When the queue with high priority is empty, send the packets in the lower-priority queue.

Table 1-8 Configure the PQ queue

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the classification rule	**priority-list** *list-number* **protocol {ip | mpls | qos-group}** { **high** | **normal** | **medium** | **low** } { **fragments** | **gt** *packet-size* | **lt** *packet-size* | **list** *access-list-name* | **tcp** *port-number* | **udp** *port-number*}	Optional *packet-size* contains the length of the packet link header; *access-list-name* only supports IP ACL; PQ does not support MAC ACL.
Configure the classification rule of the ingress interface	**priority-list** *list-number* **interface** *interface-name* { **high** | **low** | **medium** |**normal** }	Optional According to the ingress interface, distribute the packets to the specified queue. By default, the packet enters the default queue.
Configure the PQ default queue	**priority-list** *list-number* **default** { **high** | **medium** | **normal** | **low** }	Optional By default, the default queue is normal queue.

Step	Command	Description
Configure the PQ queue depth	**priority-list** *list-number* **queue-limit** *high-queue-length medium-queue-length normal-queue-length low-queue-length*	Optional The user can adjust the depths of the four PQ queues as desired. By default, the depth of each queue is 256 packets.
Enter the interface configuration mode	**interface** *interface-name*	-
Apply the PQ policy	**priority-group** *list-number*	Mandatory Apply the PQ policy with *list-number* to the interface and the range of *list-number* is 1-16.

Configure CBWFQ Queue

Configuring CBWFQ includes the following steps:

1. Use the class-map command to define the communication class, distinguishing the flows according to one rule;
2. Use the policy-map command to define the policy, defining the rule for the communication class, including bandwidth, dropping policy, queue depth and perform the marking, traffic monitoring and shaping for the data class;
3. Apply the policy: Use the service-policy command to apply the policy to the interface.

Define the communication class: The communication class is used to distinguish the flows entering one policy, convenient for the later operations for the flows. There is one default communication class in the system class-default: match the default data flow. The user can define the communication class according to the common rule, including the class name and class mapping rule.

Table 1-9 Configure class-map

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Create communication class	**class-map** [**match-all** | **match-any**] *class-name*	Mandatory By default, use match-all. match-all means that when matching all items in class-map, the matching succeeds; match-any means that when matching any item in match-any, the matching succeeds.
Configure the communication class matching the ingress interface	**match input-interface** *input-interface-name*	Optional
Configure the communication class matching the egress interface	**match output-interface** *output-interface-name*	Optional Configure the communication class matching the egress interface mainly to be used with the re-direction function of the sub interface.
Configure the communication class matching the IP priority	**match ip precedence** *precedence-value*	Optional
Configure the communication class matching the IP DSCP	**match ip dscp** *dscp-value*	Optional
Configure the communication class matching the MPLS-EXP	**match mpls experimental topmost** *exp-value*	Optional
Configure the communication class matching the VLAN ID	**match vlan** *vlan-id*	Optional In one match, we can match multiple (four at most) VLAN IDs. As long as matching any one, the matching succeeds.

Step	Command	Description
Configure the communication type matching vrf	**match vrf** *vrf-name*	Optional
Configure the communication class matching the 802.1p CoS	**match cos** *cos-value*	Optional In one match, we can match multiple (four at most) CoS. As long as matching any one, the matching succeeds.
Configure the communication class matching the QoS Group	**match qos-group** *qos-group-id*	Optional QoS Group is valid only at the local device; In one match, we can match multiple (four at most) QoS Group IDs. As long as matching any one, the matching succeeds.
Configure the communication class matching the nested class-map	**match class-map** *class-name*	Optional Define the current communication class nested to match other communication class.
Configure the communication class matching the application object list	**match application application-name**	Optional
Configure the communication class matching the application object group list	**Match application-group** **applicationgroup-name**	Optional

Note:

- We can only configure matching QoS Group at the egress direction. During the general application, mark QoS Group at the ingress direction and match QoS Group at the egress direction.
- Matching output interface can only be used for the re-direction function of the sub interface.
- When the communication class matches the access list, the corresponding access list definition should adopt the permit rule, but not the deny rule, because the refused flow

will enter the matching of the next communication class and is meaningless for the local communication class.

Define policy: CBWFQ policy can be applied at the ingress and egress directions of the interface. The policy at the ingress direction includes the flow classification and marking, while the policy at the egress direction includes the flow classification, marking and the mechanism of distributing queue for each communication class, traffic monitoring and shaping, as well as drop policy. The supported queues of the system include LLQ, Bandwidth and BE queues. Each policy has one default communication class (class-default) and default BE queue.

Table 1-10 Configure the policy-map

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Create CBWFQ policy	**policy-map** *policy-name*	Mandatory
Enter the policy configuration mode	**class** *class-name*	Mandatory Apply the communication class to the policy. If at the egress direction, the user can specify the queue type for the communication class and by default, it is BE queue; while at the ingress direction, there is no queue.
Specify the EF queue for the communication class	**priority** { **percent** *percentage* | *bandwidth* } [*burst*] [**limited**]	Optional The bandwidth can be absolute bandwidth value and also can be percentage. After configuring **limited**, the priority queue bandwidth is limited to the value configured by the user and cannot occupy the remaining bandwidth.
Specify the AF queue for the communication class	**bandwidth** { **percent** *percentage* | *bandwidth-in-kbps* }	Optional The bandwidth can be absolute bandwidth value and also can be percentage.

Step	Command	Description
Specify the queue depth for the communication type	**queue-limit** *queue-depth*	Optional
Enable the queue WRED drop policy for the communication class	**random-detect** [**prec-based** | **dscp-based**]	Optional
Configure the WRED parameters for the communication class	**random-detect** [**precedence** | **dscp**] *pre-dscp-val min-threshold max-threshold* [*pro- denominator*]	Optional
Configure fair-queue for the communication class	**fair-queue** [*numb-of-queue*]	Optional
Specify the nested CBWFQ sub policy for the communication class	**service-policy** *sub-policy-name*	Optional
Configure marking the packet IP priority for the communication class	**set ip precedence** *prec-value*	Optional
Configure marking the packet DSCP for the communication class	**set ip dscp** *dscp-value*	Optional
Configure marking the EXP of the MPLS packet for the communication class	**set mpls experimental** { **imposition** | **topmost** } *exp-value*	Optional **Imposition** and **topmost** are mutually exclusive.
Configure marking 802.1p CoS for the communication class	**set cos** *cos-value*	Optional

Step	Command	Description
Configure marking QoS Group for the communication class	**set qos-group** *qos-group-id*	Optional QoS Group is valid only for the local device.
Configure the traffic monitoring for the communication class	**police cir** *cir-value* [**bc** *bc-value* [**be** *be-value* | **pir** *pir-value* **be** *pir-be-value*]]	Optional Police is to perform the traffic monitoring for the communication class in the QoS policy. There are various monitoring modes, including single-speed double-color, single-speed three-color and double-speed three-color, and perform the related actions for the colored packets. By default, **conform-action** is forwarding; **exceed-action** and **violate-action** are dropping. For details, refer to QoS Command Manual.
Configure the traffic shaping for the communication class	**shape average** { **percent** *percentage* | *bandwidth* }	Optional The traffic shaping bandwidth can be absolute value and also can be percentage.
Configure dropping for the communication class	**drop**	Optional The **drop** command is mutually exclusive; after configuring the **drop** command for one communication class, we cannot configure other commands any more.

Note:

- In policy-map, configure the **priority** or **bandwidth** or **shape average** command for one communication class, indicating that the communication class has the independent queue. Here, the policy-map can only be applied to the egress direction of the interface, but cannot be applied to the ingress direction of the interface.
- **queue-limit**, **fair-queue** and **wred** are all the attributes of the queue. If one communication class does not configure the **priority** or **bandwidth** or **shape average**

command first, we cannot configure the above three commands in the communication class. Except for class-default, its default queue is BE queue, so we can directly configure the queue attributes on it.

- The expressing forms of the AF queue bandwidths in one policy-map should be consistent, absolute value or percentage. For some (such as SA card asynchronization) interface card with escape, when configuring the bandwidth distribution on this kind of interfaces, it is suggested to use the bandwidth absolute value, but not percentage.
- In one policy-map, the user can configure 254 queues at most, including LLQ (priority), AF (bandwidth), and new BE queue. And the user cannot configure the fair-queue queue in the LLQ queue.
- policy-map of imposition-exp marked with QoS Group and MPLS can only be applied to the ingress direction of the interface, while the policy-map marked with 802.1p CoS can only be applied to the egress direction of the Ethernet sub interface.
- The description of the CBWFQ bandwidth calculation:

Configure as follows:

```
Policy-map: 1
 class 1
  bandwidth percent 20
 class 2
  priority percent 50
 class 3
  shape average percent 20
 class 4
```

policy-map 1 is applied to one 1000M interface. The corresponding guarantee bandwidth and maximum bandwidth of the policy-map 1 policy instance are both 1000M.

The policy-map 1 will generate four queues; class 1 corresponds to the af queue; the maximum bandwidth is 1000M; the guarantee bandwidth is 20/100*1000 = 200M. class 2 corresponds to the ef queue, the maximum bandwidth is 1000M, and the guarantee bandwidth is 50/100*1000 = 500M.

Class 3 corresponds to the be queue. Except for the guarantee bandwidth of af and ef, the remaining guarantee bandwidth of the policy instance is distributed to the be queues on average). The guarantee bandwidth of class 3 is 1000-500-200/2 (two be queues) = 150M, the maximum bandwidth is 20/100*1000=200M. Class 4 does not configure the queue packet to enter the default queue.

The description for the be queue calculation: Each policy instance has one default be queue and the queue is unavailable for the user.

For the case with nesting, the description of the further instance:

There is the configuration of policy-map 1; configure one policy-map sub again, as follows:

```
Policy-map: sub
 class sub-1
  bandwidth percent 20
 class sub-2
  priority percent 50
 class sub-3
```

```
 shape average percent 20
class sub-4
```

1: If nesting the policy sub under class 1:

```
Policy-map: 1
 class 1
  bandwidth percent 20
  service-policy sub
 class 2
  priority percent 50
.....
```

In this case, the bandwidth calculation of the sub policy instance is:

If the guarantee bandwidth of class 1 is 200M, and the maximum bandwidth is 1000M (refer to the above description), the guarantee bandwidth of the sub policy is 200M and the maximum bandwidth is 1000M.

The guarantee bandwidth of sub-1 is 20/100*200 = 40M, the maximum bandwidth is 1000M, the guarantee bandwidth of sub-2 is 50/100*200 = 100M, the maximum bandwidth is 1000M, the guarantee bandwidth of sub-3 is (200-40-100)/2 = 30M, and the maximum bandwidth is 20/100*1000M = 200M.

2: If nesting policy sub in class 3:

```
Router#show policy-map 1
Policy-map: 1
...
 class 2
  priority percent 50
 class 3
  shape average percent 20
  service-policy sub
 class 4
```

In this case, the bandwidth calculation of the sub policy instance is:

If the guarantee bandwidth of class 3 is 150M, and the maximum bandwidth is 200M (refer to the above description), the guarantee bandwidth of the sub policy is 150M and the maximum bandwidth is 200M.

The guarantee bandwidth of sub-1 is 20/100*150 = 30M, the maximum bandwidth is 200M, the guarantee bandwidth of sub-2 is 50/100*150 = 75M, the maximum bandwidth is 200M, the guarantee bandwidth of sub-3 is (150-30-75)/2 = 22.5M, and the maximum bandwidth is 20/100*200M = 40M.

3: When configuring shape and bandwidth or priority in one class at the same time, the guarantee bandwidth the configured bandwidth or priority value, and the maximum bandwidth is the configured shape value.

Configure as follows:

```
Policy-map: 1
 class 1
  bandwidth percent 20
  shape average percent 30
 class 2
  ...
```

- The guarantee bandwidth of class 1 is 200M, the maximum bandwidth is the configured of shape 30/100*1000M = 300M. For the case, it is not suggested that the shape bandwidth value configured by the user is smaller than the bandwidth value of bandwidth or priority.

Apply policy: Adopt the **service-policy** command to apply the policy to the ingress or egress direction of the interface.

Table 1-11 Apply the CBWFQ policy

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Apply the CBWFQ policy	**service-policy** { **input** | **output** } *policy-name*	Mandatory

Note:

- policy-map with the queue features (configured with **queue-limit**, **wred**, **shape**, **bandwidth**, **priority**) cannot be applied to the ingress direction.

By default, for CBWFQ, the queue scheduling policy order of exceeding the guarantee bandwidth is BE >AF >EF; in some special applications, we may need to modify the order. We can use the following command to adjust dynamically.

Table 1-12 Configure CBWFQ guarantee bandwidth scheduling policy

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Adjust CBWFQ scheduling policy	**qos-cbwfq excess-prio high** { **af** | **be** | **ef** } **middle** { **af** | **be** | **ef** } **low** { **af** | **be** | **ef** }	Mandatory By default, the queue scheduling policy order of exceeding the guarantee bandwidth is BE >AF >EF

Note:

- It is suggested to use the **qos-cbwfq excess-prio** command for the special application. Usually, adopt the default configuration.

Configure the SPQ Queue

SPQ is a strict priority queue. The queue priority is set according to the user's DSCP value for packets. The packets without priority enter the default priority queue (the lowest priority). The queue is scheduled strictly from high priority to low priority. Each queue can be set by the user to limit the use bandwidth.

SPQ supports 65 queues. 0-63 queue users can configure the mapping DSCP value, the packet will enter the corresponding queue according to the configured DSCP value, and the un-configured packet will enter the default queue 64. The priorities of queues 63 to 0 are from high to low, and the priority of queue 64 is the lowest. At the same time, users can limit the bandwidth of each queue configuration. The queues are scheduled from the highest-priority queue to the lowest-priority queue.

Table 1-13 Configure the SPQ queue

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Apply the SPQ policy	**spqueue** *queueId* **pq cir** *cir-value* [**pir** *pir-value* [*queue-length*] | **limited** [*queue-length*]]	Mandatory *queue-length* is the queue depth, and by default, it is 256.

1.2.4. Configure Congestion Avoidance

When the congestion happens to the interface, WRED selects to drop the data with low priority. This provides the differentiated performance features for the different service types of data. Usually, the lower the IP priority of the data is, the more possibly the data is dropped. When enabling WRED on the interface, the user can modify the drop parameters as desired, including average queue threshold, drop probability denominator and so on. For the WRED applications with the same drop parameter requirement, we can use the WRED group, which is convenient for the user to set the parameters. The default drop policy of the system congestion avoidance is tail drop.

WRED can be enabled in the class queues of CBWFQ and also can be enabled in the interface. The former is described in “Configure the congestion management—Configure the CBWFQ queue”. The following mainly describes enabling the WRED function in the interface.

Configuration Condition

None

Configure WRED

Table 1-14 Configure the WRED

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the WRED group	**random-detect-group** *group-name*	Mandatory
Configure the minimum threshold, maximum threshold and drop probability percentage of the average queue based on the IP DSCP drop mode	**dscp** *dscp minimum-threshold maximum-threshold* [*discard-probability-percent*]	Optional
Configure the minimum threshold, maximum threshold and drop probability percentage of the average queue based on the IP Precedence drop mode	**precedence** *precedence minimum-threshold maximum-threshold* [*discard-probability-percent*]	Optional
Return to the global configuration mode	**exit**	-

Step	Command	Description
Enter the interface configuration mode	**interface** *interface-name*	-
Enable the WRED group	**random-detect attach** *group-name*	Mandatory (alternative of enabling WRED group and enabling WRED)
Enable WRED	**random-detect** [**prec-based | dscp-based**]	Mandatory (alternative of enabling WRED group and enabling WRED)
Configure the minimum threshold, maximum threshold and drop probability percentage of the average queue based on the IP Precedence drop mode	**random-detect precedence** *precedence minimum-threshold maximum-threshold* [*discard-probability-percent*]	Optional
Configure the minimum threshold, maximum threshold and drop probability percentage of the average queue based on the IP DSCP drop mode	**random-detect dscp** *dscp minimum-threshold maximum-threshold* [*discard-probability-percent*]	Optional

Note:

- On one interface, we can configure only one WRED or WRED group.
- If CBWFQ is enabled on one interface, we cannot apply WRED or WRED group on the interface.

1.2.5. Configure Sub Interface Re-direction

By default, each sub interface of the system performs the QoS control independently (queue scheduling, traffic shaping, traffic monitoring and so on), not related with the corresponding main interface. However, in some application scenarios, for example, adopt three sub interfaces to distinguish three kinds of services, while the three kinds of services share one WAN line and the bandwidth is also shared. To adopt the QoS queue (PQ queue, CBWFQ queue and so on) to distribute the bandwidth for the services dynamically, we need to configure the re-direction for the sub interfaces, making the packets sent by the sub interface be re-directed to the corresponding main interface for QoS queue scheduling. And then configure the QoS queue on the main interface to control the bandwidth distribution.

Configuration Condition

None

Configure Sub Interface Re-direction

Table 1-15 Configure the sub interface re-direction function

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enable the re-direction function of the sub interface	**qos sub-interface redirect**	Mandatory By default, do not enable the re-direction function of the sub interface.
Add the sub interface to be re-directed	**redirect** *interface-name* [*target-interface-name*]	Mandatory

Note:

- If there are several sub interfaces in one main interface, the system permits some to add and the others not to add to the re-direction, the sub interfaces not added to the re-direction still perform the QoS control independently.
- After configuring the sub interface re-direction, we also need to configure the QoS control policy (queue scheduling, traffic shaping, traffic monitoring and so on) on the corresponding main interface.

1.2.6. Configure RSVP Guaranteed Bandwidth

RSVP guaranteed bandwidth is used for RSVP TE tunnel forwarding of the MPLS module to ensure that the RSVP TE tunnel packets of MPLS are sent first.

There is no guarantee by default. The RSVP guaranteed bandwidth needs to be configured according to the scenario to ensure RSVP packets.

Configuration Condition

None

Configure RSVP Guaranteed Bandwidth

Table 1-16 Configure RSVP guaranteed bandwidth

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the specified interface	**interface** *interface-name*	-
Configure RSVP guaranteed bandwidth	**qos rsvp-bandwidth** *bandwidth* [**rlimit-max**]	Mandatory

Note:

- By default, after RSVP exceeds the guaranteed bandwidth, it can preempt the remaining bandwidth. After rlimit-max is configured, the remaining bandwidth will not be preempted.

1.2.7. QoS Monitoring and Maintaining

Table 1-17 QoS monitoring and maintaining

Command	Description
clear policy-map interface *interface-name* [**input** | **output**]	Clear the statistics information of the CBWFQ policy on the interface
clear queueing interface *interface-name*	Clear the statistics information of the queue on the specified interface
show class-map [*class-name*]	Display the class-map information configured by the user
show fifo [**interface** *interface-name*]	Display the configuration and statistics information of the FIFO queue on the interface
show policy-map	Display the CBWFQ policy information configured by the user

Command	Description
show policy-map user-config policy-name [class { class-name | class-default }]	Display the CBWFQ policy information specified by the user
show policy-map interface *interface-name* [**pvc** *pvc-number*] [**input** | **output**]	Display the information of the CBWFQ policy on the interface
show priority { **list** [*list-number*] | **queue** [**interface** *interface-name* [pvc *pvc-number*]] }	Display the configured PQ queue rule information or the statistics information of the PQ queue on the interface
show qos-cbwfq excess-prio	Display the scheduling priority of the queues after exceeding the guarantee bandwidth in CBWFQ
show qos-apply interface { cbq | pq | wfq }	Display the information of applying the QoS queue on the interface
show qos interface [*interface-name*]	Display the QoS information of the interface
show qos sub-interface redirect	Display the configured sub interface re-direction information
show queueing interface *interface-name*	Display the statistics information of the queue on the specified interface
show random-detect-group [*group-name*]	Display the configuration information of the WRED group
show rate-limit interface *interface-name*	Display the configuration of CAR in one interface and the statistics information during running
show traffic-shape [**interface** *interface-name*]	Display the GTS configuration and statistics information on the interface
show wred [**interface** *interface-name*]	Display the WRED configuration information

1.3. QoS Typical Configuration Example

1.3.1. Configure Basic CAR

Network Requirements

- There is the video server and data server in the network.
- It is required to configure CAR to limit the egress traffic of the video server not larger than 30M 9:00-18:00 every day, configure the IP priority of the video traffic as 5, and the egress traffic of the data server not larger than 50M in one day.

Network Topology

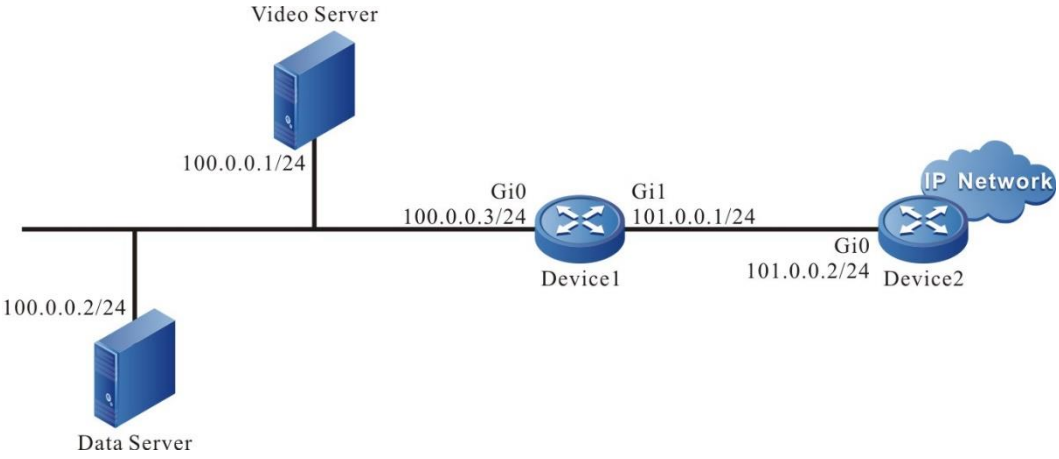

Figure 1-5 Networking of configuring the basic CAR

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: Configure the ACL rule list matching the video server and data server traffic.

#Configure the time domain of 9:00-18:00 every day.

```
Device1#configure terminal
Device1(config)#time-range video-time
Device1(config-time-range)#periodic daily 9:00 to 18:00
Device1(config-time-range)#exit
```

#Configure the ACL rule list matching the video server and associating the time domain.

```
Device1(config)#ip access-list extended video
Device1(config-ext-nacl)#10 permit ip host 100.0.0.1 any time-range video-time
Device1(config-ext-nacl)#exit
```

#Configure the ACL rule list matching the data server.

```
Device1(config)#ip access-list extended data
Device1(config-ext-nacl)#10 permit ip host 100.0.0.2 any
Device1(config-ext-nacl)#exit
```

Step 3: Configure the CAR matching ACL.

#On Device1, connect the ingress direction of the interface of the video server; configure the CAR matching the video traffic ACL; limit the video traffic to 30M in the specified time; conform traffic action is to forward packets and mark the IP priority as 5.

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#rate-limit input access-group video 30000000 1875000 0 conform-action set-prec-transmit 5 exceed-action drop
```

#On Device1, connect the ingress direction of the interface of the data server, configure the CAR matching the data traffic ACL, and limit the data traffic speed to 50M.

```
Device1(config-if-gigabitethernet0)#rate-limit input access-group data 50000000 3125000 0 conform-action transmit exceed-action drop
```

Step 4: Check the result.

#View the CAR packet statistics. If there is traffic, we can see that whether the configured CAR matches the packet.

```
Device1#show rate-limit interface gigabitethernet 0

 display input direct carInst running-info:

  matches: access-group video
  color-mode: blind
     params: 30000000 bps, 1875000 limit, 0 extended limit
     conformed 106484 packets, 82312132 bytes; action: set-prec-transmit
     exceeded 64649 packets, 49973677 bytes; action: drop
     current burst: 55362 ns

  matches: access-group data
  color-mode: blind
     params: 50000000 bps, 3125000 limit, 0 extended limit
     conformed 251160 packets, 127589280 bytes; action: transmit
     exceeded 0 packets, 0 bytes; action: drop
     current burst: 69683072 ns
```

1.3.2. Configure Non-Color-Blind CAR

Network Requirements

- There is the video server and data server in the network.

- It is required to configure CAR to ensure the video traffic 20M, but not exceeding 20M. The total of the video traffic and data traffic does not exceed 50M. That is to say, the traffic passing Device1 does not exceed 50M. When the video traffic is larger than 20M, ensure that the passing video traffic is 20M; when the video traffic is smaller than 20M, the remaining bandwidth can be occupied by the data traffic.

Network Topology

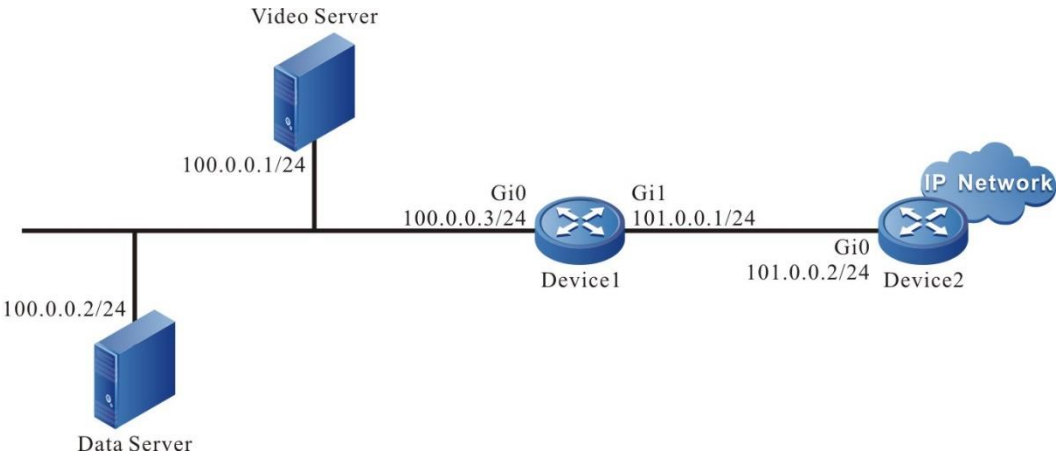

Figure 1-6 Networking of configuring the non-color-blind CAR

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: Configure the ACL rule list matching the video server traffic.

```
Device1#configure terminal
Device1(config)#ip access-list extended video
Device1(config-ext-nacl)#10 permit ip host 100.0.0.1 any
Device1(config-ext-nacl)#exit
```

Step 3: Configure CAR on the interface.

#On Device1, connect the ingress direction of the interface of the video and data servers, configure CAR matching the video traffic ACL, limit the video traffic to 20M, and the conform traffic action is to continue to match the next CAR (continue).

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#rate-limit input access-group video 20000000
2500000 0 conform-action continue exceed-action drop
```

#On Device1, connect the ingress direction of the interface of the video and data servers, configure the interface-based CAR, limit the speed to 50M and the mode is color-keep.

```
Device1(config-if-gigabitethernet0)#rate-limit input 50000000 6250000 0 conform-
action transmit exceed-action drop color-keep
```

Step 4: Check the result.

#View the CAR packet statistics. If there is traffic, we can see that whether the configured CAR matches the packet.

```
Device1#show rate-limit interface gigabitethernet 0


 display input direct carInst running-info:


  matches: access-group video
  color-mode: blind
     params: 20000000 bps, 2500000 limit, 0 extended limit
     conformed 106745 packets, 82513885 bytes; action: continue
     exceeded 143893 packets, 111229289 bytes; action: drop
     current burst: -8800 ns


  matches: all traffic
  color-mode: color-aware
     params: 50000000 bps, 6250000 limit, 0 extended limit
     conformed 253633 packets, 148346649 bytes; action: transmit
     exceeded 86934 packets, 44162472 bytes; action: drop
     current burst: -38400 ns
```

1.3.3. Configure GTS

Network Requirements

- The aggregation router is down-linked to lots of network routers via the MSTP line of the carrier. The physical bandwidth of each MSTP line is 4M. Adopt the point-to-multipoint mode to connect; on Devie1, the sub interface corresponds to the network router.
- On the sub interface of the aggregation router connecting the carrier MSTP device interface, configure GTS; shape the traffic to the network node.
- Adjust the method of the device QoS calculating the bandwidth to make it close to the carrier.

Network Topology

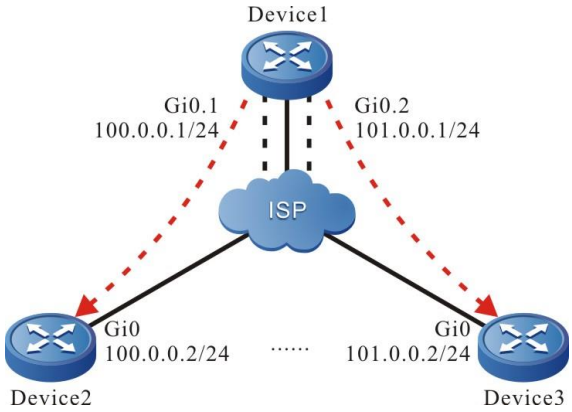

Figure 1-7 Networking of configuring GTS

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: Configure GTS on the network sub interface of the aggregation router Device1 connecting the MSTP device interface (Provide two examples).

```
Device1#configure terminal
Device1(config)#interface gigabitethernet 0.1
Device1(config-if-gigabitethernet0.1)#traffic-shape 4000000 100000
Device1(config-if-gigabitethernet0.1)#exit
Device1(config)#interface gigabitethernet 0.2
Device1(config-if-gigabitethernet0.2)#traffic-shape 4000000 100000
Device1(config-if-gigabitethernet0.2)#exit
```

Step 3: Configure the QoS packet calculation to add 20 bytes, making it close to the line transmission cost of the carrier, and modify the logical bandwidth of the interface to be consistent with the line bandwidth.

```
Device1(config)#interface gigabitethernet 0.1
Device1(config-if-gigabitethernet0.1)#qos account output length add 20
Device1(config-if-gigabitethernet0.1)#bandwidth 4000
Device1(config-if-gigabitethernet0.1)#exit
Device1(config)#interface gigabitethernet 0.2
Device1(config-if-gigabitethernet0.2)#qos account output length add 20
Device1(config-if-gigabitethernet0.2)#bandwidth 4000
Device1(config-if-gigabitethernet0.2)#exit
```

Step 4: The configuration of the uplink interfaces on the network router Device2 and Device3 is similar to Device1.

Step 5: Check the result.

#The traffic sent by the sub interface of the downlink network site does not exceed the value configured by the traffic shaping.

Note:

- The device QoS calculation is processed by adding the length of the data part of the link layer to the header of the link layer, while the MSTP line provided by the carrier is realized by binding one to multiple 2M channels and there is internal transmission cost. According to the actual application experience, the cost of each packet is about 20 bytes, so we need to configure qos account output length add 20, making each packet be added with 20 bytes when the device performs the bandwidth calculation. This complies with the actuality of the carrier MSTP line and ensures that the packets are not lost when being transmitted on the carrier MSTP line.
- The command **bandwidth 4000** is to modify the logical bandwidth of the interface to be consistent with the actual leased line bandwidth. The PQ and CBWFQ queues calculate the bandwidth guarantee based on this and also can use the **qos max-bandwidth** command to modify the QoS used bandwidth.

1.3.4. Configure PQ

Network Requirements

- There is authentication server, video server and several terminals in the network; the intranet bandwidth is 100M and the egress bandwidth is 20M.
- It is required to ensure the traffic of the authentication server first, then ensure the video traffic, and at last, the terminal traffic.

Network Topology

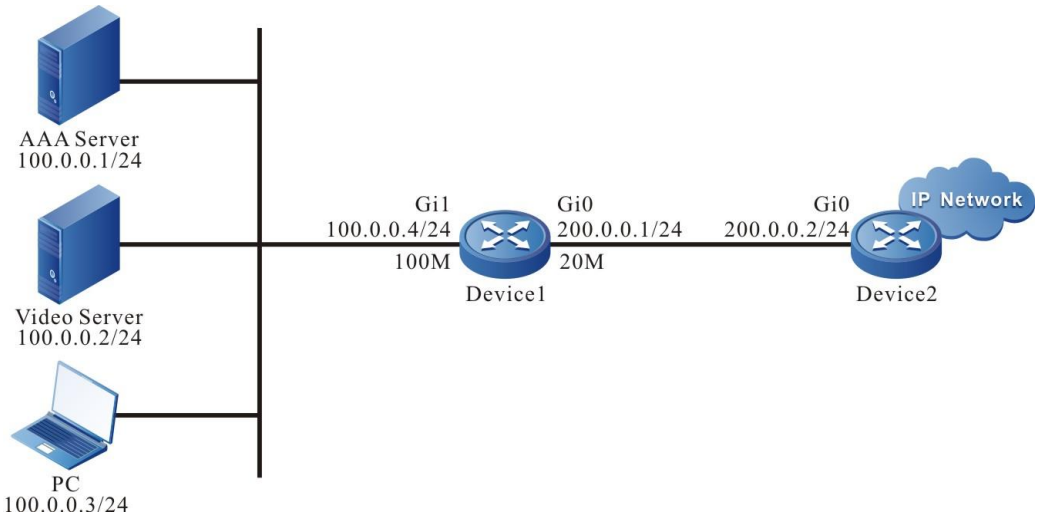

Figure 1-8 Networking of configuring PQ

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: Configure the ACL rule list matching the video server and data server traffic.

```
Device1#configure terminal
Device1(config)#ip access-list extended aaa
Device1(config-ext-nacl)#10 permit ip host 100.0.0.1 any
Device1(config-ext-nacl)#exit
Device1(config)#ip access-list extended video
Device1(config-ext-nacl)#10 permit ip host 100.0.0.2 any
Device1(config-ext-nacl)#exit
```

Step 3: Configure the PQ rule 1.

#Configure the authentication server traffic to enter the high-priority queue of PQ.

```
Device1(config)#priority-list 1 protocol ip high list aaa
```

#Configure the video server traffic to enter the medium-priority queue of PQ.

```
Device1(config)#priority-list 1 protocol ip medium list video
```

#Configure other traffic to enter the normal-priority queue of PQ by default (the default is normal, which cannot be configured).

```
Device1(config)#priority-list 1 default normal
```

Step 4: Apply PQ rule 1 on the interface and configure GTS and interface bandwidth.

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#priority-group 1
Device1(config-if-gigabitethernet0)#traffic-shape 20000000 500000
Device1(config-if-gigabitethernet0)#bandwidth 20000
```

Note:

- If the actual leased line bandwidth is not consistent with the physical bandwidth of the interface, we need to configure GTS on the interface. Otherwise, the packets may be dropped on the carrier line and the QoS control on the device does not take effect. If the actual leased line bandwidth is not consistent with the bandwidth configured on the interface, we also need to modify the bandwidth or qos max-bandwidth of the interface. If the 1000M Ethernet is connected to 20M carrier MSTP line, we need to configure 20M GTS at the egress direction of the interface and modify the bandwidth of the interface to 20M.
- The device QoS calculation is processed by adding the length of the data part of the link layer to the header of the link layer, while the MSTP line provided by the carrier is realized by binding one to multiple 2M channels and there is internal transmission cost. According to the actual application experience, the cost of each packet is about 20 bytes, so if connecting the MSTP line of the carrier, we also need to configure **qos account output length add 20**, making each packet be added with 20 bytes when the device performs the bandwidth calculation. This complies with the actuality of the carrier MSTP

line and ensures that the packets are not lost when being transmitted on the carrier MSTP line.

Step 5: Check the result.

#Check the status of the packet entering the queue.

```
Device1#show priority queue

Queue of gigabitethernet0 priority-group 1
-------------------------------------------------------------------------------
----------------------
QID  DEPTH(packets)  HOLD(bytes/pkts)  INPUT(bytes/pkts)  OUTPUT(bytes/pkts)
DROP(bytes/pkts)
1(high)      256          0 /0          42515 /55          42515 /55          0 /0
2(medium)    256          0 /0          743712 /1464       743712 /1464       0 /0
3(normal)    256          0 /0          544464 /684        544464 /684        0 /0
4(low)       256          0 /0          0 /0               0 /0               0 /0
```

DEPTH is the length of the PQ queue. HOLD is the current cached packet bytes of the queue; INPUT and OUTPUT are the statistics of the in-queue and out-queue packets respectively; DROP is the statistics of the dropped packets of the PQ queue.

1.3.5. Configure FQ

Network Requirements

- PC initiates various service traffics, the egress bandwidth is 2M and the intranet bandwidth is 100M.
- On Device1, enable FQ to make different kinds of traffics be sent fairly during congestion.

Network Topology

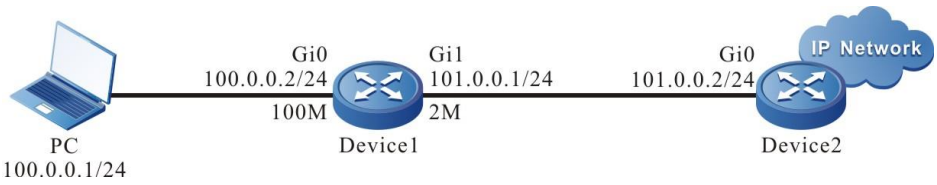

Figure 1-9 Networking of configuring FQ

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: The egress interface enables FQ.

```
Device1#configure terminal
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)#fair-queue
```

Step 3: Check the result.

#View the information about the FQ queue packets entering the queue and the sending status.

```
Device1#show wfq
Interface gigabitethernet1
-------------------------------------------------------------------------------
------------
--QID----QUEUE LENGTH----HoldPkts------SendPkts--------DropPkts-----
  0    256          53          218          1739
  1    256          61          207          1742
  2    256          61          207          1742
  3    256          61          207          1742
  4    256          61          207          1744
  5    256          61          207          1743
  6    256          61          207          1744
  7    256          61          207          1744
  8    256          61          208          1745
  9    256          61          207          1744
 10    256          61          207          1744
 11    256          61          207          1744
 12    256          61          207          1744
 13    256          61          207          1744
 14    256          61          207          1744
 15    256          61          207          1742
 16    256          61          207          1770
 17    256          61          207          1767
 18    256          61          215          1763
 19    256          61          219          1762
 20    256          61          219          1762
 21    256          61          219          1762
 22    256          61          219          1762
 23    256          61          219          1762
 24    256          61          219          1762
 25    256          61          219          1762
 26    256          61          219          1762
 27    256          61          219          1762
 28    256          61          219          1762
```

```
29   256          61          219          1762
30   256          59          219          1764
31   256          56          219          1767
-------------------------------------------------------------------------------
-----------
```

1.3.6. Configure CBWFQ

Network Requirements

- The data flow is sent from Device1, Device2, and Device3 to the network after Device5 via Device4; the link bandwidth between Device4 and Device5 is 50M;
- On Device1, mark the DSCP domain of the traffic sent from Device1 as AF41; on Device2, mark the DSCP domain of the traffic sent from Device2 as AF42; on Device3, mark the DSCP domain of the traffic sent from Device3 as AF43.
- On Device4, configure QoS to guarantee the protocol packets. For the traffic with the DSCP domain AF41 and AF42 sent to Device5, guarantee 60% low-delay bandwidth; guarantee 30% bandwidth for the traffic with DSCP domain AF43, but the highest occupied bandwidth does not exceed 40%.

Network Topology

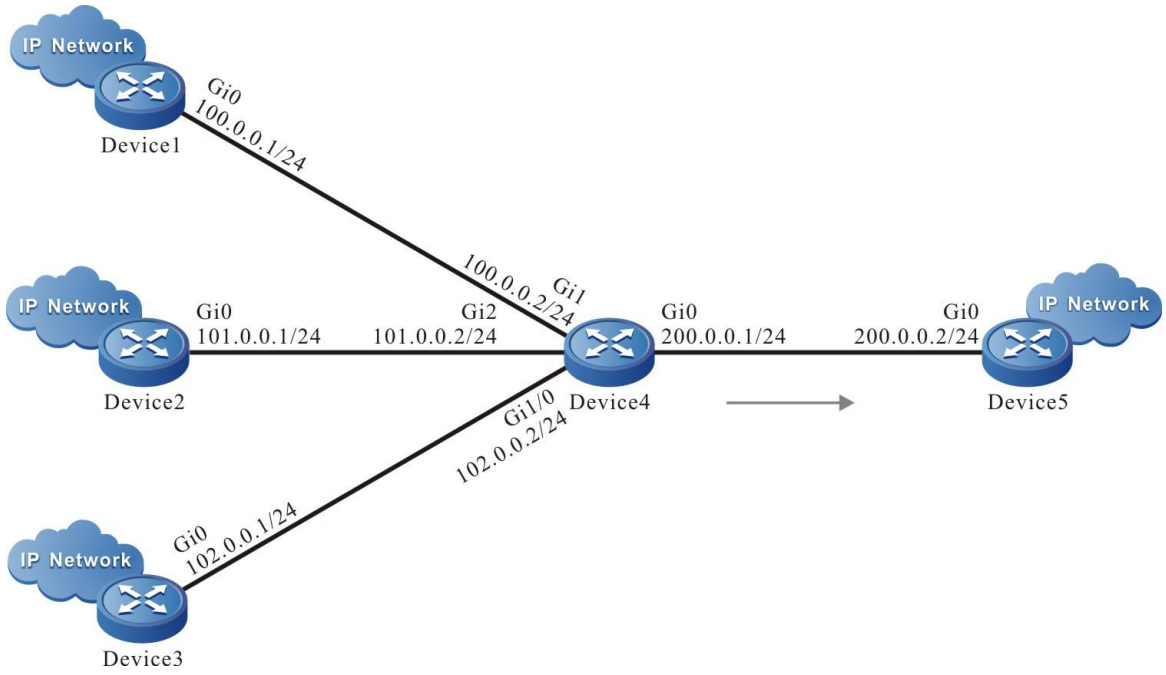

Figure 1-10 Networking of configuring CBWFQ

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: On Device1, Device2 and Device3, configure CBWFQ to mark the packet DSCP.

#On Device1, mark the DSCP domain of all packets going out from the interface gigabitethernet0 as AF41.

```
Device1#configure terminal
```

```
Device1(config)#policy-map set_dscp
Device1(config-pmap)#class class-default
Device1(config-pmap-c)#set ip dscp af41
Device1(config-pmap-c)#exit
Device1(config-pmap)#exit
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#service-policy output set_dscp
Device1(config-if-gigabitethernet0)#exit
```

#On Device2, mark the DSCP domain of all packets going out from the interface gigabitethernet0 as AF42.

```
Device2#configure terminal
Device2(config)#policy-map set_dscp
Device2(config-pmap)#class class-default
Device2(config-pmap-c)#set ip dscp af42
Device2(config-pmap-c)#exit
Device2(config-pmap)#exit
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#service-policy output set_dscp
Device2(config-if-gigabitethernet0)#exit
```

#On Device3, mark the DSCP domain of all packets going out from the interface gigabitethernet0 as AF43.

```
Device3#configure terminal
Device3(config)#policy-map set_dscp
Device3(config-pmap)#class class-default
Device3(config-pmap-c)#set ip dscp af43
Device3(config-pmap-c)#exit
Device3(config-pmap)#exit
Device3(config)#interface gigabitethernet 0
Device3(config-if-gigabitethernet0)#service-policy output set_dscp
Device3(config-if-gigabitethernet0)#exit
```

Step 3: On Device4, configure the classes matching the traffics.

#Configure the class matching the DSCP domain AF41 and AF42.

```
Device4#configure terminal
Device4(config)#class-map match-any af4142
Device4(config-cmap)#match ip dscp af41
```

```
Device4(config-cmap)#match ip dscp af42
Device4(config-cmap)#exit
```

#Configure the class matching the DSCP domain AF43.

```
Device4(config)#class-map match-all af43
Device4(config-cmap)#match ip dscp af43
Device4(config-cmap)#exit
```

Step 4: Configure the CBWFQ policy on Device4.

```
Device4(config)#policy-map qos
Device4(config-pmap)#class af4142
Device4(config-pmap-c)#priority percent 60
Device4(config-pmap-c)#exit
Device4(config-pmap)#class af43
Device4(config-pmap-c)#bandwidth percent 30
Device4(config-pmap-c)#shape average percent 40
Device4(config-pmap-c)#exit
Device4(config-pmap)#exit
```

The step of configuring the bandwidth guarantee also can use the absolute value.

#Check the configured policy.

```
Device4#show policy-map user-config qos
Policy-map: qos
 class af4142
 priority percent 60
class af43
 bandwidth percent 30
 shape average percent 40
```

Step 5: Configure GTS, interface bandwidth and the policy of applying the configuration at the egress direction of the interface on Device4.

```
Device4#configure terminal
Device4(config)#interface gigabitethernet 0
Device4(config-if-gigabitethernet0)#traffic-shape 50000000 1250000
Device4(config-if-gigabitethernet0)#bandwidth 50000
Device4(config-if-gigabitethernet0)#service-policy output qos
Device4(config-if-gigabitethernet0)#exit
```

Note:

- If the actual leased line bandwidth is not consistent with the physical bandwidth of the interface, we need to configure GTS on the interface. Otherwise, the packets may be dropped on the carrier line and the QoS control on the device does not take effect. If the actual leased line bandwidth is not consistent with the bandwidth configured on the interface, we also need to modify the bandwidth or qos max-bandwidth of the interface. If the 1000M Ethernet is connected to 20M carrier MSTP line, we need to configure 20M GTS at the egress direction of the interface and modify the bandwidth of the interface to 20M.
- The device QoS calculation is processed by adding the length of the data part of the link layer to the header of the link layer, while the MSTP line provided by the carrier is realized by binding one to multiple 2M channels and there is internal transmission cost. According to the actual application experience, the cost of each packet is about 20 bytes, so if connecting the MSTP line of the carrier, we also need to configure **qos account output length add 20**, making each packet be added with 20 bytes when the device performs the bandwidth calculation. This complies with the actuality of the carrier MSTP line and ensures that the packets are not lost when being transmitted on the carrier MSTP line.

Step 6: Check the result.

#View the statistics information of the packets entering the queue.

```
Device4#show policy-map interface gigabitethernet 0
 interface gigabitethernet0
  Service-policy output: qos

    Class-map: af4142 (match-any)
     58 packets 44834 bytes
    5 minute offered rate 3624 bps
    match ip dscp af41
    match ip dscp af42
    Queueing
    queue limit 256 packets
    (queue depth/total drops) 0/0
    (packets output/bytes output) 58/44834
    Priority: 60% (30000 Kbps)  , burst bytes 625000, b/w exceed drops: 0

   Class-map: af43 (match-all)
    1260 packets 640080 bytes
    5 minute offered rate 51720 bps
    match ip dscp af43
    Queueing
```

queue limit 256 packets

(queue depth/total drops) 1/0

(packets output/bytes output) 1259/639572

Bandwidth: 30% (15000 Kbps)

Shaping

shape (average) cir 20000000, bc 4000000

Class-map: class-default (match-any)

0 packets 0 bytes

5 minute offered rate 0 bps

Queueing

queue limit 256 packets

(queue depth/total drops) 0/0

(packets output/bytes output) 0/0

match any

From the above information, we can see the basic configuration information, the absolute value converted from the bandwidth guarantee, the statistics of each kind of matched packets and 5-minute traffic statistics, as well as the packet stacking of the current QoS queue. We can simply judge whether the configuration is correct and effective.

1.3.7. Configure HQoS

Network Requirements

- There is video service, voice service and data service in the network; the video service includes class-1 video, class-2 video and class-3 video.
- Configure QoS to guarantee the protocol packets, ensuring that the video service occupies 30% bandwidth, voice service occupies 30% low-delay bandwidth, and the data service is scheduled fairly; for class-1 video in the video service, ensure 40% low-delay bandwidth; for class-2 video, ensure 30% bandwidth; for class-3 video, ensure 29% bandwidth.

Network Topology

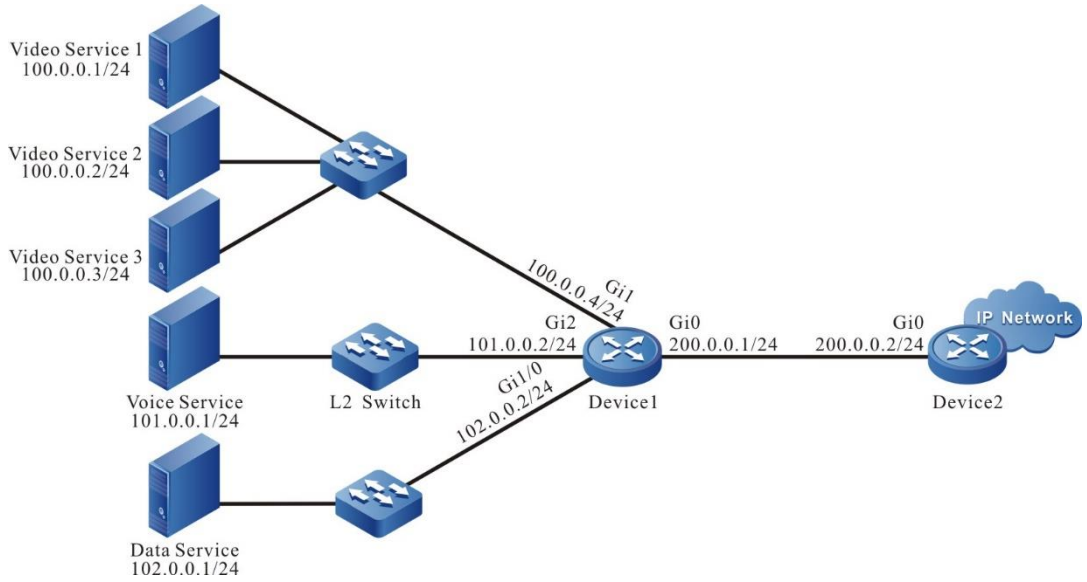

Figure 1-11 Networking of configuring HQoS

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: On Device1, configure the ACL rule list matching the services.

#Configure the ACL rule list matching the video service.

```
Device1#configure terminal
Device1(config)#ip access-list extended video
Device1(config-ext-nacl)#10 permit ip 100.0.0.0 0.0.0.255 any
Device1(config-ext-nacl)#exit
```

#Configure the ACL rule list matching the three sub services of the video.

```
Device1(config)#ip access-list extended video1
Device1(config-ext-nacl)#10 permit ip host 100.0.0.1 any
Device1(config-ext-nacl)#exit
Device1(config)#ip access-list extended video2
Device1(config-ext-nacl)#10 permit ip host 100.0.0.2 any
Device1(config-ext-nacl)#exit
Device1(config)#ip access-list extended video3
Device1(config-ext-nacl)#10 permit ip host 100.0.0.3 any
Device1(config-ext-nacl)#exit
```

#Configure the ACL rule list matching the voice service.

```
Device1(config)#ip access-list extended voice
Device1(config-ext-nacl)#10 permit ip host 101.0.0.1 any
```

```
Device1(config-ext-nacl)#exit
```

Step 3: On Device1, configure the classes matching the services.

#Configure the class matching the video service.

```
Device1(config)#class-map video
Device1(config-cmap)#match access-group video
Device1(config-cmap)#exit
Device1(config)#class-map video1
Device1(config-cmap)#match access-group video1
Device1(config-cmap)#exit
Device1(config)#class-map video2
Device1(config-cmap)#match access-group video2
Device1(config-cmap)#exit
Device1(config)#class-map video3
Device1(config-cmap)#match access-group video3
Device1(config-cmap)#exit
```

#Configure the class matching the voice service.

```
Device1(config)#class-map voice
Device1(config-cmap)#match access-group voice
Device1(config-cmap)#exit
```

Step 4: Configure the CBWFQ policy on Device1.

#Configure the CBWFQ sub policy of ensuring the video sub service.

```
Device1(config)#policy-map video
Device1(config-pmap)#class video1
Device1(config-pmap-c)#priority percent 40
Device1(config-pmap-c)#exit
Device1(config-pmap)#class video2
Device1(config-pmap-c)#bandwidth percent 30
Device1(config-pmap-c)#exit
Device1(config-pmap)#class video3
Device1(config-pmap-c)#bandwidth percent 29
Device1(config-pmap-c)#exit
Device1(config-pmap)#exit
```

#Configure the CBWFQ main policy of ensuring all services and nest the sub policy of ensuring the video sub service in the video class. The data service enters the default class-default queue.

```
Device1(config)#policy-map qos
Device1(config-pmap)#class video
Device1(config-pmap-c)#bandwidth percent 30
Device1(config-pmap-c)#service-policy video
Device1(config-pmap-c)#exit
Device1(config-pmap)#class voice
Device1(config-pmap-c)#priority percent 30
Device1(config-pmap-c)#exit
Device1(config-pmap)#class class-default
Device1(config-pmap-c)#fair-queue
Device1(config-pmap-c)#exit
Device1(config-pmap)#exit
```

Step 5: Apply the configured policy at the egress direction of the Device1 interface.

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#service-policy output qos
Device1(config-if-gigabitethernet0)#exit
```

Note:

- If the actual leased line bandwidth is not consistent with the physical bandwidth of the interface, we need to configure GTS on the interface. Otherwise, the packets may be dropped on the carrier line and the QoS control on the device does not take effect. If the actual leased line bandwidth is not consistent with the bandwidth configured on the interface, we also need to modify the bandwidth or qos max-bandwidth of the interface. If the 1000M Ethernet is connected to 20M carrier MSTP line, we need to configure 20M GTS at the egress direction of the interface and modify the bandwidth of the interface to 20M.
- The device QoS calculation is processed by adding the length of the data part of the link layer to the header of the link layer, while the MSTP line provided by the carrier is realized by binding one to multiple 2M channels and there is internal transmission cost. According to the actual application experience, the cost of each packet is about 20 bytes, so if connecting the MSTP line of the carrier, we also need to configure **qos account output length add 20**, making each packet be added with 20 bytes when the device performs the bandwidth calculation. This complies with the actuality of the carrier MSTP line and ensures that the packets are not lost when being transmitted on the carrier MSTP line.

Step 6: Check the result.

#View the CBWFQ queue statistics information.

```
Device1#show policy-map interface gigabitethernet 0
```

```
interface gigabitethernet0
 Service-policy output: qos

  Class-map: video (match-all)
    26338 packets 11721879 bytes
    5 minute offered rate 8525000 bps
    match access-group video
    Queueing
    queue limit 1024 packets
    (queue depth/total drops) 0/0
    (packets output/bytes output) 26336/11721098
    Bandwidth: 30% (300000 Kbps)

    Service-policy: video

      Class-map: video1 (match-all)
        7519 packets 2563979 bytes
        5 minute offered rate 1864712 bps
        match access-group video1
        Queueing
        queue limit 256 packets
        (queue depth/total drops) 0/0
        (packets output/bytes output) 7518/2563638
        Priority: 40% (120000 Kbps)  , burst bytes 3000000, b/w exceed drops: 0

      Class-map: video2 (match-all)
        5914 packets 2602160 bytes
        5 minute offered rate 1892480 bps
        match access-group video2
        Queueing
        queue limit 256 packets
        (queue depth/total drops) 0/0
        (packets output/bytes output) 5914/2602160
        Bandwidth: 30% (90000 Kbps)

      Class-map: video3 (match-all)
        12906 packets 6556248 bytes
```

5 minute offered rate 4768184 bps
match access-group video3
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 12905/6555740
Bandwidth: 29% (87000 Kbps)

Class-map: class-default (match-any)
0 packets 0 bytes
5 minute offered rate 0 bps
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 0/0
match any

Class-map: voice (match-all)
13083 packets 6646164 bytes
5 minute offered rate 6646160 bps
match access-group voice
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 0/0
Priority: 30% (300000 Kbps) , burst bytes 7500000, b/w exceed drops: 0

Class-map: class-default (match-any)
202543 packets 68107754 bytes
5 minute offered rate 19459360 bps
match any
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 1/78
Fair-queue 256: per-flow queue limit 64 Packets

From the above information, we can see the basic configuration information, the absolute value converted from the bandwidth guarantee, the statistics of each kind of matched packets and 5-minute traffic statistics, as well as the packet stacking of the current QoS queue. We can simply judge whether the configuration is correct and effective.

1.3.8. Configure MPLS QoS

Network Requirements

- As shown in the following MPLS L3VPN network, there are four VPNs.
- On PE1, configure the QoS policy; for the flow from the left CE in the following figure, the packet of VPN1 marks MPLS EXP as 1; the packet of VPN2 marks MPLS EXP as 2; the packet of VPN3 marks MPLS EXP as 3; the packet of VPN4 marks MPLS EXP as 4.
- On PE1, configure the QoS policy, ensuring the sending of the protocol packets; for the traffic with MPLS EXP 1, ensure the 10M bandwidth; for the traffic with MPLS EXP 2, ensure the 20M bandwidth; for the traffic with MPLS EXP 3, ensure the 30M bandwidth; the remaining bandwidth is occupied by the traffic with MPLS EXP 4.
- On PE2, configure QoS; for the flow from PE1, after the MPLS tag of the packet with MPLS EXP 3 pops out, the IP priority is marked as 3; for the traffic with MPLS EXP 2, limit the rate to 10M.

Network Topology

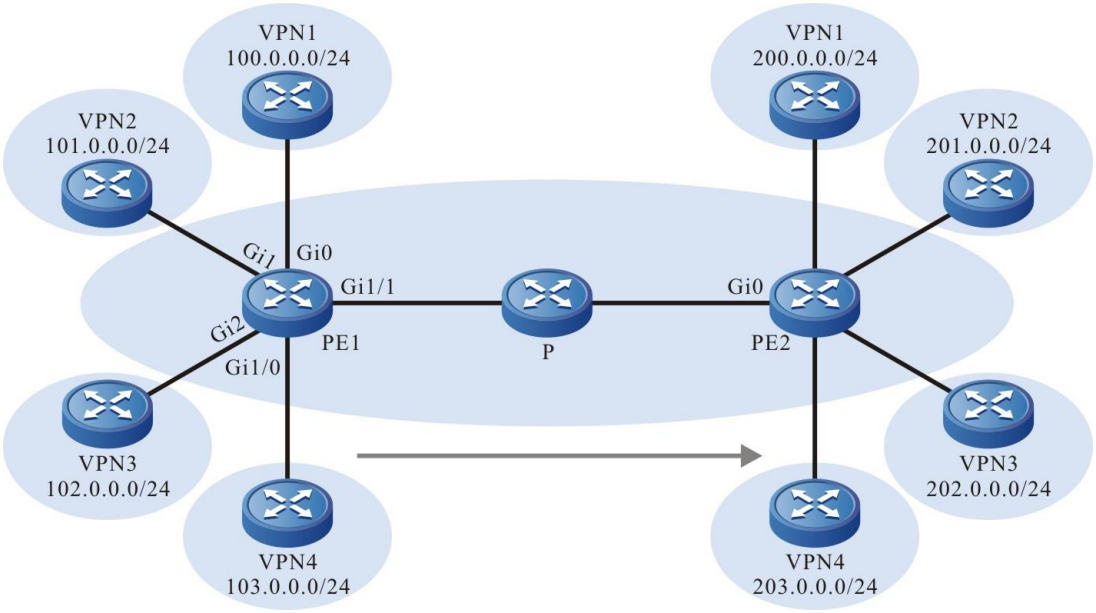

Figure 1-12 Networking of configuring MPLS QoS

Configuration Steps

Step 1: Set up the MPLS L3VPN environment, as shown in the above figure. For details, refer to the “MPLS L3VPN” chapter of the configuration manual.

Step 2: On PE1, configure the ACL rule list matching the four VPNs connected to CE.

```
PE1#configure terminal
PE1(config)#ip access-list extended vpn1
PE1(config-ext-nacl)#10 permit ip 100.0.0.1 0.0.0.255 any
```

```
PE1(config-ext-nacl)#exit
PE1(config)#ip access-list extended vpn2
PE1(config-ext-nacl)#10 permit ip 101.0.0.1 0.0.0.255 any
PE1(config-ext-nacl)#exit
PE1(config)#ip access-list extended vpn3
PE1(config-ext-nacl)#10 permit ip 102.0.0.1 0.0.0.255 any
PE1(config-ext-nacl)#exit
PE1(config)#ip access-list extended vpn4
PE1(config-ext-nacl)#10 permit ip 103.0.0.1 0.0.0.255 any
PE1(config-ext-nacl)#exit
```

Step 3: On PE1, configure the classes matching the VPN traffics.

```
PE1(config)#class-map vpn1
PE1(config-cmap)#match access-group vpn1
PE1(config-cmap)#exit
PE1(config)#class-map vpn2
PE1(config-cmap)#match access-group vpn2
PE1(config-cmap)#exit
PE1(config)#class-map vpn3
PE1(config-cmap)#match access-group vpn3
PE1(config-cmap)#exit
PE1(config)#class-map vpn4
PE1(config-cmap)#match access-group vpn4
PE1(config-cmap)#exit
```

Step 4: On PE1, configure and apply the CBWFQ policy of marking MPLS EXP at the ingress.

#Configure the CBWFQ policy.

```
PE1(config)#policy-map set_exp
PE1(config-pmap)#class vpn1
PE1(config-pmap-c)#set mpls experimental imposition 1
PE1(config-pmap-c)#exit
PE1(config-pmap)#class vpn2
PE1(config-pmap-c)#set mpls experimental imposition 2
PE1(config-pmap-c)#exit
PE1(config-pmap)#class vpn3
```

```
PE1(config-pmap-c)#set mpls experimental imposition 3
PE1(config-pmap-c)#exit
PE1(config-pmap)#class vpn4
PE1(config-pmap-c)#set mpls experimental imposition 4
PE1(config-pmap-c)#exit
PE1(config-pmap)#exit
```

#Apply the CBWFQ policy.

```
PE1(config)#interface gigabitethernet 0
PE1(config-if-gigabitethernet0)#service-policy input set_exp
PE1(config-if-gigabitethernet0)#exit
PE1(config)#interface gigabitethernet 1
PE1(config-if-gigabitethernet1)#service-policy input set_exp
PE1(config-if-gigabitethernet1)#exit
PE1(config)#interface gigabitethernet 2
PE1(config-if-gigabitethernet2)#service-policy input set_exp
PE1(config-if-gigabitethernet2)#exit
PE1(config)#interface gigabitethernet 1/0
PE1(config-if-gigabitethernet1/0)#service-policy input set_exp
PE1(config-if-gigabitethernet1/0)#exit
```

There are multiple marking methods. The above configuration just describes the example. In fact, for the environment, adopt four different CBWFQ policies to apply on the four interfaces, which is more convenient for marking all packets passing the interface.

Step 5: On PE1, configure and apply the CBWFQ policy of the egress bandwidth guarantee.

#Configure the class of matching MPLS EXP.

```
PE1(config)#class-map exp1
PE1(config-cmap)#match mpls experimental topmost 1
PE1(config-cmap)#exit
PE1(config)#class-map exp2
PE1(config-cmap)#match mpls experimental topmost 2
PE1(config-cmap)#exit
PE1(config)#class-map exp3
PE1(config-cmap)#match mpls experimental topmost 3
PE1(config-cmap)#exit
PE1(config)#class-map exp4
PE1(config-cmap)#match mpls experimental topmost 4
PE1(config-cmap)#exit
```

#Configure the CBWFQ policy.

```
PE1(config)#policy-map qos
PE1(config-pmap)#class exp1
PE1(config-pmap-c)#bandwidth 10000
PE1(config-pmap-c)#exit
PE1(config-pmap)#class exp2
PE1(config-pmap-c)#bandwidth 20000
PE1(config-pmap-c)#exit
PE1(config-pmap)#class exp3
PE1(config-pmap-c)#bandwidth 30000
PE1(config-pmap-c)#exit
PE1(config-pmap)#exit
```

The packets with MPLS EXP 4 enter the class-default queue.

#Apply the CBWFQ policy.

```
PE1(config)#interface gigabitethernet 1/1
PE1(config-if-gigabitethernet1/1)#service-policy output qos
PE1(config-if-gigabitethernet1/1)#exit
```

Note:

- If the actual leased line bandwidth is not consistent with the physical bandwidth of the interface, we need to configure GTS on the interface. Otherwise, the packets may be dropped on the carrier line and the QoS control on the device does not take effect. If the actual leased line bandwidth is not consistent with the bandwidth configured on the interface, we also need to modify the bandwidth or qos max-bandwidth of the interface. If the 1000M Ethernet is connected to 20M carrier MSTP line, we need to configure 20M GTS at the egress direction of the interface and modify the bandwidth of the interface to 20M.
- The device QoS calculation is processed by adding the length of the data part of the link layer to the header of the link layer, while the MSTP line provided by the carrier is realized by binding one to multiple 2M channels and there is internal transmission cost. According to the actual application experience, the cost of each packet is about 20 bytes, so if connecting the MSTP line of the carrier, we also need to configure **qos account output length add 20**, making each packet be added with 20 bytes when the device performs the bandwidth calculation. This complies with the actuality of the carrier MSTP line and ensures that the packets are not lost when being transmitted on the carrier MSTP line.

Step 6: Check the result.

#View the statistics information of the CBWFQ queue.

```
PE1#show policy-map interface gigabitethernet 1/1
 interface gigabitethernet1/1
  Service-policy output: qos
```

Class-map: exp1 (match-all)
13083 packets 6646164 bytes
5 minute offered rate 6646160 bps
match mpls experimental topmost 1
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 0/0
Bandwidth: 10000 Kbps

Class-map: exp2 (match-all)
89293 packets 45360844 bytes
5 minute offered rate 12960240 bps
match mpls experimental topmost 2
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 89296/45362368
Bandwidth: 20000 Kbps

Class-map: exp3 (match-all)
29342 packets 23356232 bytes
5 minute offered rate 6673208 bps
match mpls experimental topmost 3
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 29343/23357028
Bandwidth: 30000 Kbps

Class-map: class-default (match-any)
202543 packets 68107754 bytes
5 minute offered rate 19459360 bps
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0

```
(packets output/bytes output) 202550/68109787
match any
```

From the above information, we can see the basic configuration information, the absolute value converted from the bandwidth guarantee, the statistics of each kind of matched packets and 5-minute traffic statistics, as well as the packet stacking of the current QoS queue. We can simply judge whether the configuration is correct and effective.

Step 7: Configure the QoS of PE2 marking and rate limitation.

#Configure the class of matching the packets with MPLS EXP 3.

```
PE2#configure terminal
PE2(config)#class-map exp3
PE2(config-cmap)#match mpls experimental topmost 3
PE2(config-cmap)#exit
```

#Configure the IP priority as 3 after the MPLS tag of the packet whose MPLS EXP is marked by CBWFQ as 3 pops out and apply at the ingress direction of the interface on the P device.

```
PE2(config)#policy-map set_ip
PE2(config-pmap)#class exp3
PE2(config-pmap-c)#set ip precedence 3
PE2(config-pmap-c)#exit
PE2(config-pmap)#exit
PE2(config)#interface gigabitethernet 0
PE2(config-if-gigabitethernet0)#service-policy input set_ip
PE2(config-if-gigabitethernet0)#exit
```

#Configure the ingress CAR to limit the speed of the MPLS packets with outer tag EXP 2 to 10M.

```
PE2(config)#interface gigabitethernet 0
PE2(config-if-gigabitethernet0)#rate-limit input mpls experimental topmost 2
10000000 625000 0 conform-action transmit exceed-action drop
PE2(config-if-gigabitethernet0)#
```

#VPN2 connected to PE2 receives up to 10M traffic, and the IP priority of the packet received by VPN3 connected to PE2 is marked as 3.

1.3.9. Configure QoS Sub Interface Re-direction Function

Network Requirements

- PC1, PC2 and PC3 are connected with the L2 switch, running different services respectively, and the packets carry different VLAN TAGs, communicating with Device via the sub interface in the point-to-multipoint. The L2 switch is connected with Device via the carrier MSTP line. The interface gigabitethernet 0.1 of Device carries the packets of PC1 services, gigabitethernet 0.2 carries the packets of PC2 services, and gigabitethernet 0.3 carries the packets of PC3 services.

- The carrier MSTP line provides 20M bandwidth; configure QoS on Device, ensuring that the downlink traffic sent to PC1 occupies 50% bandwidth, the downlink traffic sent to PC2 occupies 30% bandwidth and the remaining is distributed to the downlink traffic sent to PC3. When any service is idle, the idle bandwidth can be used by the other services.

Network Topology

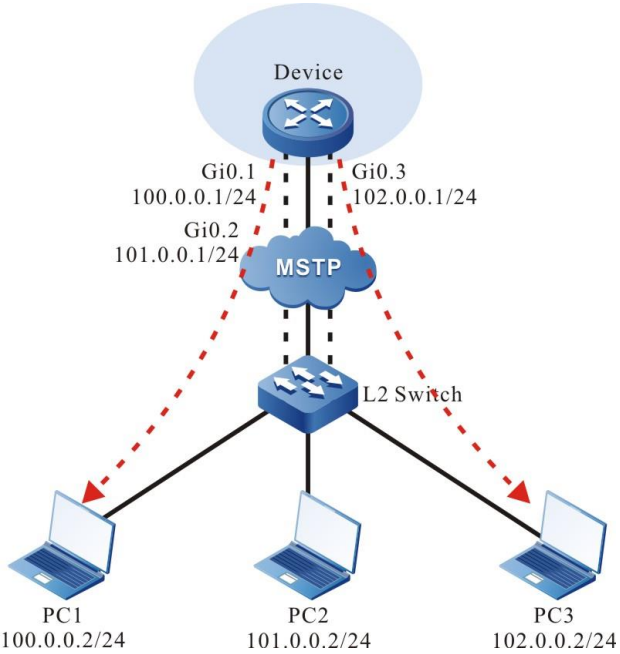

Figure 1-13 Networking of configuring QoS sub interface re-direction function

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: Enable the interface re-direction function and add three sub interfaces.

```
Device#configure terminal
Device(config)#qos sub-interface redirect
Device(config-qosredirect)#redirect gigabitethernet 0.1
Device(config-qosredirect)#redirect gigabitethernet 0.2
Device(config-qosredirect)#redirect gigabitethernet 0.3
Device(config-qosredirect)#exit
```

The traffic of the added sub interface QoS channel is re-directed to the corresponding main interface, that is, the traffic of the sub interface is controlled by the QoS control configured on the main interface.

Step 3: Configure the class of matching the traffic.

#Configure the class of matching the traffic going out from the sub interface gigabitethernet 0.1.

```
Device(config)#class-map match-all pc1
Device(config-cmap)#match output-interface gigabitethernet 0.1
```

```
Device(config-cmap)#exit
```

#Configure the class of matching the traffic going out from the sub interface gigabitethernet 0.2.

```
Device(config)#class-map match-all pc2
Device(config-cmap)#match output-interface gigabitethernet 0.2
Device(config-cmap)#exit
```

#Configure the class of matching the traffic going out from the sub interface gigabitethernet 0.3.

```
Device(config)#class-map match-all pc3
Device(config-cmap)#match output-interface gigabitethernet 0.3
Device(config-cmap)#exit
```

Note:

- The above adopts the egress interface to perform the flow classification. The mode is optional and we also can adopt the ACL and packet ToS modes to classify the traffic.

Step 4: Configure the CBWFQ policy.

```
Device(config)#policy-map qos
Device(config-pmap)#class pc1
Device(config-pmap-c)#bandwidth percent 50
Device(config-pmap-c)#exit
Device(config-pmap)#class pc2
Device(config-pmap-c)#bandwidth percent 30
Device(config-pmap-c)#exit
Device(config-pmap)#exit
```

The traffic of PC3 enters the class-default queue.

Step 5: Configure GTS on the main interface, modify the interface bandwidth, configure the QoS calculating bandwidth compensation, and apply the QoS policy.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#qos account output length add 20
Device(config-if-gigabitethernet0)#bandwidth 20000
Device(config-if-gigabitethernet0)#traffic-shape 20000000 500000
Device(config-if-gigabitethernet0)#service-policy output qos
Device(config-if-gigabitethernet0)#exit
```

Note:

- If the actual leased line bandwidth is not consistent with the physical bandwidth of the interface, we need to configure GTS on the interface. Otherwise, the packets may be dropped on the carrier line and the QoS control on the device does not take effect. If the actual leased line bandwidth is not consistent with the bandwidth configured on the interface, we also need to modify the bandwidth or qos max-bandwidth of the interface. If the 1000M Ethernet is connected to 20M carrier MSTP line, we need to configure 20M

GTS at the egress direction of the interface and modify the bandwidth of the interface to 20M.

- The device QoS calculation is processed by adding the length of the data part of the link layer to the header of the link layer, while the MSTP line provided by the carrier is realized by binding one to multiple 2M channels and there is internal transmission cost. According to the actual application experience, the cost of each packet is about 20 bytes, so if connecting the MSTP line of the carrier, we also need to configure **qos account output length add 20**, making each packet be added with 20 bytes when the device performs the bandwidth calculation. This complies with the actuality of the carrier MSTP line and ensures that the packets are not lost when being transmitted on the carrier MSTP line.

Step 6: Check the result.

#View the CBWFQ queue statistics information.

```
Device#show policy-map interface gigabitethernet 0
 interface gigabitethernet0
  Service-policy output: qos

   Class-map: pc1 (match-all)
    13083 packets 6646164 bytes
    5 minute offered rate 6646160 bps
    match output-interface gigabitethernet0.1
    Queueing
    queue limit 256 packets
    (queue depth/total drops) 0/0
    (packets output/bytes output) 0/0
    Bandwidth: 50% (10000 Kbps)

   Class-map: pc2 (match-all)
    16881 packets 7427640 bytes
    5 minute offered rate 1856912 bps
    match output-interface gigabitethernet0.2
    Queueing
    queue limit 256 packets
    (queue depth/total drops) 0/0
    (packets output/bytes output) 16882/7428080
    Bandwidth: 30% (6000 Kbps)

   Class-map: class-default (match-any)
    58310 packets 26035535 bytes
```

```
5 minute offered rate 6508880 bps
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 58311/26036043
match any
```

From the above information, we can see the basic configuration information, the absolute value converted from the bandwidth guarantee, the statistics of each kind of matched packets and 5-minute traffic statistics, as well as the packet stacking of the current QoS queue. We can simply judge whether the configuration is correct and effective.

1.3.10. Configure WRED

Network Requirements

- Lots of terminals download files from the FTP server.
- On Device, enable WRED to prevent TCP global synchronization.
- Modify the minimum drop threshold of the packet queue with WRED IP priority 2 to 20% of the average queue depth, the maximum drop threshold to 90% of the average queue depth, and the maximum threshold to 50.

Network Topology

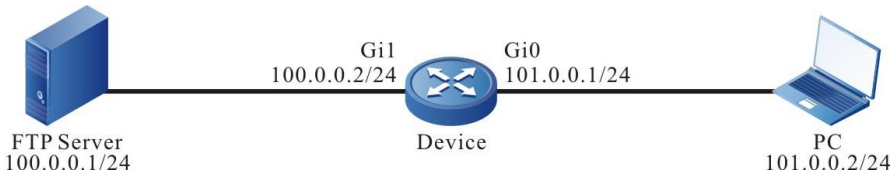

Figure 1-14 Networking of configuring the WRED

Configuration Steps

Step 1: Configure the IP address and configuration route of the interface. (omitted)

Step 2: The interface enables the priority-based WRED.

```
Device#configure terminal
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#random-detect prec-based
```

Step 3: Modify the minimum drop threshold of the packet queue with IP priority 2 to 20% of the average queue depth, the maximum drop threshold to 90% of the average queue depth, and the drop probability to 20%.

```
Device(config-if-gigabitethernet0)#random-detect precedence 2 20 90 20
```

Step 4: Check the result.

#View the WRED configuration information.

```
Device#show wred

 gigabitethernet0
     Queueing strategy: random early detection (WRED)
     Exp-weight-constant: 9

    Class   MinThre MaxThre DiscardP
    0       30%     100%    10   %
    1       30%     100%    10   %
    2       20%     90%     20%
    3       30%     100%    10   %
    4       30%     100%    10   %
    5       30%     100%    10   %
    6       30%     100%    10   %
    7       30%     100%    10   %
```

#When the network is congested, multiple concurrent TCP connections will not generate the global synchronization phenomenon.

1.3.11. Configure IPv6 CBWFQ

Network Requirements

- The whole network is an IPv6 network. The data flow is sent from Device1, Device2 and Device3 to the network connected by Device5 through Device4. The link bandwidth between Device4 and Device5 is 100M.
- Mark the DSCP value of the traffic from interface Gi0 as AF11 on Device1, mark the DSCP value of the traffic from interface Gi0 as AF21 on Device2, and mark the priority of the traffic from interface Gi0 as 6 on Device3.
- On Device4, configure QoS to ensure 30% low-delay bandwidth for the traffic with the DSCP value of AF11 sent to Device5, 30% bandwidth forwarding for the traffic with the DSCP value of AF21, and 30% and no more than 40% bandwidth forwarding for the traffic with the priority 6.

Network Topology

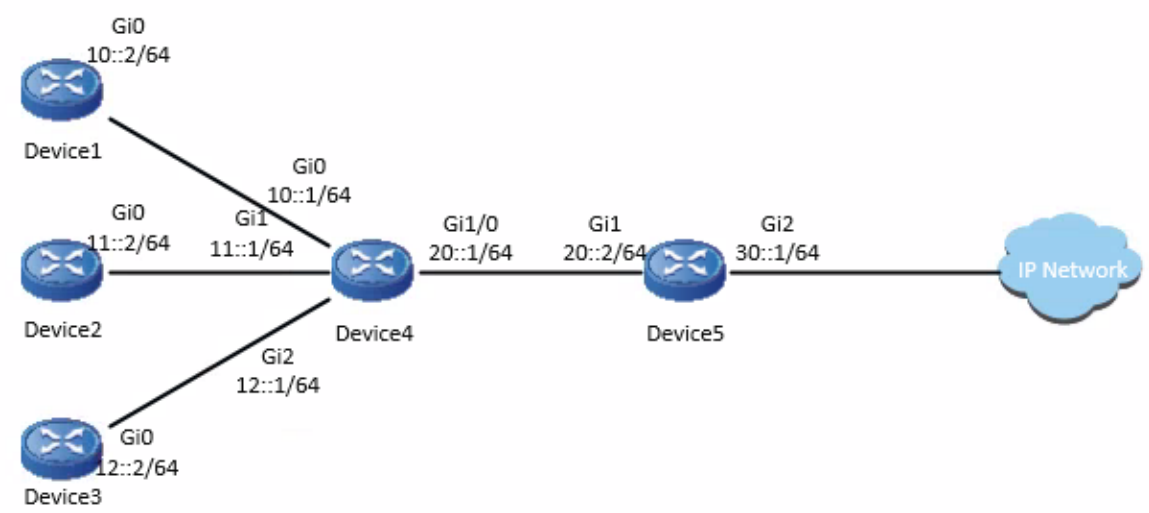

Figure 1-15 Networking of configuring IPv6 CBWFQ

Configuration Steps

Step 1: Configure the IP address of the interface, and configure the route (omitted).

Step 2: Configure CBWFQ to modify the packet mark.

#Configure Device1, marking the DSCP value of the packet from the interface gigabitethernet0 as AF11.

```
Device1#configure terminal
Device1(config)#policy-map set_dscp
Device1(config-pmap)#class class-default
Device1(config-pmap-c)#set ip dscp af11
Device1(config-pmap-c)#exit
Device1(config-pmap)#exit
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#service-policy output set_dscp
Device1(config-if-gigabitethernet0)#exit
```

#Configure Device2, marking the DSCP value of the packet from the interface gigabitethernet0 as AF21.

```
Device2#configure terminal
Device2(config)#policy-map set_dscp
Device2(config-pmap)#class class-default
Device2(config-pmap-c)#set ip dscp af21
Device2(config-pmap-c)#exit
Device2(config-pmap)#exit
Device2(config)#interface gigabitethernet 0
```

```
Device2(config-if-gigabitethernet0)#service-policy output set_dscp
Device2(config-if-gigabitethernet0)#exit
```

#Configure Device3, marking the priority of the packet from the interface gigabitetherne0 as 6.

```
Device3#configure terminal
Device3(config)#policy-map set_prec
Device3(config-pmap)#class class-default
Device3(config-pmap-c)#set ip precedence 6
Device3(config-pmap-c)#exit
Device3(config-pmap)#exit
Device3(config)#interface gigabitethernet 0
Device3(config-if-gigabitethernet0)#service-policy output set_dscp
Device3(config-if-gigabitethernet0)#exit
```

Step 3: On Device4, configure the class matching the traffic.

#Configure matching the class with the DSCP value AF11.

```
Device4#configure terminal
Device4(config)#class-map match-all af11
Device4(config-cmap)#match ip dscp af11
Device4(config-cmap)#exit
```

#Configure matching the class with the DSCP value AF21.

```
Device4(config)#class-map match-all af21
Device4(config-cmap)#match ip dscp af21
Device4(config-cmap)#exit
```

#Configure matching the class with the priority 6.

```
Device4(config)#class-map match-all prec6
Device4(config-cmap)#match ip precedence 6
Device4(config-cmap)#exit
```

Step 4: On Device4, configure the CBWFQ policy.

```
Device4(config)#policy-map qos
Device4(config-pmap)#class af11
Device4(config-pmap-c)#priority percent 30
Device4(config-pmap-c)#exit
Device4(config-pmap)#class af21
Device4(config-pmap-c)#bandwidth percent 30
Device4(config-pmap-c)#exit
```

```
Device4(config-pmap)#class prec6
Device4(config-pmap-c)#bandwidth percent 30
Device4(config-pmap-c)#shape average percent 40
Device4(config-pmap-c)#exit
Device4(config-pmap)#exit
```

Note:

- Absolute values can also be used to configure bandwidth guarantee.

#Check the configured policy.

```
Device4#show policy-map user-config qos
Policy-map: qos
 class af11
  priority percent 30
 class af21
  bandwidth percent 30
 class prec6
  bandwidth percent 30
  shape average percent 40
```

Step 5: On the interface of Device4, configure GTS and interface bandwidth, and apply the CBWFQ policy at the outgoing direction.

```
Device4(config)#interface gigabitethernet 1/0
Device4(config-if-gigabitethernet1/0)#traffic-shape 100000000 2500000
Device4(config-if-gigabitethernet1/0)#bandwidth 100000
Device4(config-if-gigabitethernet1/0)#service-policy output qos
Device4(config-if-gigabitethernet1/0)#exit
```

Note:

- If the actual leased line bandwidth is inconsistent with the physical bandwidth of the interface, GTS needs to be configured on the interface. Otherwise, the packet may be discarded on the operator's line and the QoS control on the device will not work. If the actual leased line bandwidth is inconsistent with the configured bandwidth of the interface, it is generally necessary to modify the bandwidth or qos max-bandwidth of the interface. If the Gigabit Ethernet port is connected to a 20M carrier MSTP line, a 20M GTS needs to be configured in the outgoing direction of the interface, and the bandwidth of the interface needs to be modified to 20M.
- The device QoS calculation is processed according to the length of the link layer header plus the link layer data part, while the MSTP line provided by the operator is actually bundled by one or more 2M channels, and there is internal transmission overhead. According to the practical application experience, the overhead per packet is generally about 20 bytes. Therefore, if the operator's MSTP line is connected, **qos account output length add 20** needs to be configured to increase 20 bytes per packet when the device performs bandwidth calculation, which is in line with the actual situation of the

operator's MSTP line and ensure that the packet is transmitted on the operator's MSTP line without packet loss.

#View the input queue statistics information.

```
Device4#show policy-map interface gigabitethernet 1/0
 interface gigabitethernet1/0
  Service-policy output: qos

   Class-map: af11 (match-all)
    10965454 packets 1315854480 bytes
    5 minute offered rate 20300552 bps
    match ip dscp af11
    Queueing
    queue limit 256 packets
    (queue depth/total drops) 1/0
    (packets output/bytes output) 10965471/1315856520
    Priority: 30% (30000 Kbps)  , burst bytes 750000, b/w exceed drops: 0

   Class-map: af21 (match-all)
    10965491 packets 1315858920 bytes
    5 minute offered rate 20300672 bps
    match ip dscp af21
    Queueing
    queue limit 256 packets
    (queue depth/total drops) 0/0
    (packets output/bytes output) 10965502/1315860240
    Bandwidth: 30% (30000 Kbps)

   Class-map: prec6 (match-all)
    10965519 packets 1315862280 bytes
    5 minute offered rate 20300776 bps
    match ip precedence 6
    Queueing
    queue limit 256 packets
    (queue depth/total drops) 0/0
    (packets output/bytes output) 10965531/1315863720
    Bandwidth: 30% (30000 Kbps)
    Shaping
```

```
shape (average) cir 40000000, bc 8000000

Class-map: class-default (match-any)
  0 packets 0 bytes
  5 minute offered rate 0 bps
  Queueing
  queue limit 256 packets
  (queue depth/total drops) 0/0
  (packets output/bytes output) 0/0
  match any
```

From the above information, you can see the basic configuration information, the absolute value converted from bandwidth guarantee, the packet statistics and 5-minute traffic statistics matched to each class, and the current QoS queue packet accumulation. You can simply judge whether the configuration is correct and effective.

1.3.12. Configure QPPB

Network Requirements

- Use BGP to open the route between Device1 and Device2. Device1 and device2 belong to AS100 and AS200 respectively.
- On Device2, mark the BGP route learned from Device1. Ensure 30% bandwidth for the data forwarded by the BGP route on the Device2 outgoing interface, but no more than 50% bandwidth. The outgoing interface bandwidth of Device2 is 50M.

Network Topology

Figure 1-16 Networking of configuring QPPB

Configuration Steps

Step 1: Configure the IP address of the interface (omitted).

Step 2: Configure the EBGP route.

#Configure Device1.

Configure to establish a direct-connected EBGP peer with Device2, and introduce 10.0.1.0/24 into BGP through network.

```
Device1#configure terminal
Device1(config)#router bgp 100
```

```
Device1(config-bgp)#neighbor 20.0.1.2 remote-as 200
Device1(config-bgp)#network 10.0.1.0 255.255.255.0
Device1(config-bgp)#exit
```

#Configure Device2.

Configure to establish a direct-connected EBGP peer with Device1, and introduce 30.0.1.0/24 into BGP through network.

```
Device2(config)#router bgp 200
Device2(config-bgp)#neighbor 20.0.1.1 remote-as 100
Device2(config-bgp)#network 30.0.1.0 255.255.255.0
Device2(config-bgp)#exit
```

#View the BGP neighbor status of Device2.

```
Device2#show ip bgp summary
BGP router identifier 30.0.1.1, local AS number 200
BGP table version is 8
2 BGP AS-PATH entries
0 BGP community entries

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
20.0.1.1      4      100      6      6       8    0    0 00:03:15        1

Total number of neighbors 1
```

You can see that Device2 and Device1 set up the BGP neighbor successfully.

#View the route table of Device2.

```
Device2#show ip route bgp
Codes: C - Connected, L - Local, S - static, R - RIP,  B - BGP, i-ISIS
       U - Per-user Static route
       O - OSPF, OE-OSPF External, M - Management, E - IRMP, EX - IRMP external

B   10.0.1.0/24 [20/0] via 20.0.1.1, 00:03:58, gigabitethernet1
```

You can see that the route of 10.0.1.0/24 network segment is learned through BGP on Device2.

Step 3: On Device2, configure the QPPB function.

#Configure the rt_list prefix list, matching the 10.0.1.0/24 network segment.

```
Device2(config)#ip prefix-list rt_list permit 10.0.1.0/24
```

#Configure the routing policy rt_map, match the rt_list prefix list, and set ip precedence and qos-group.

```
Device2(config)#route-map rt_map 1000
Device2(config-route-map)#match ip address prefix-list rt_list
Device2(config-route-map)#set ip precedence 5
Device2(config-route-map)#set qos-group 1000
Device2(config-route-map)#exit
```

#In BGP, apply the routing policy rt_map.

```
Device2(config)#router bgp 200
Device2(config-bgp)#route-policy route-map rt_map
Device2(config-cmap)#exit
```

Note:

- For the **set qos-group** command, refer to "PBR Tools" in the chapter “Unicast Routing” of the command manual.
- For details of the **route-policy** command, refer to "BGP" in the chapter “Unicast Routing” of the command manual.

Step 4: On Device2, enable the QPPB function on the input interface gigabitethernet1.

```
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#qppb source ip-prec-map in
Device2(config-if-gigabitethernet1)#qppb source ip-qos-map in
Device2(config-if-gigabitethernet1)#exit
```

#View the BGP setting tag.

```
Device2#show ip bgp 10.0.1.0/24
BGP routing table entry for 10.0.1.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
 Not advertised to any peer
 100
  20.0.1.1 from 20.0.1.1 (20.0.1.1), ip-precedence 5, qos-group 1000

    Origin IGP, metric 0, localpref 100, valid, external, best
     Last update: 00:03:41 ago
```

It can be seen that BGP sets the flags of ip precedence 5 and qos-group 1000 for the 10.0.1.0/24 network segment.

Note:

- The QPPB function supports the configuration in the in direction and out direction, and supports marking qos-group and ip precedence.

Step 5: Configure GTS and interface bandwidth and apply CBWFQ policy on the outgoing interface gigabitethernet0 of Device2.

#Set matchng the class **rt_class** with ip precedence 5 and qos-group 1000.

```
Device2(config)# class-map match-all rt_class
Device2(config-cmap)#match ip precedence 5
Device2(config-cmap)#match qos-group 1000
Device2(config-cmap)#exit
```

#Configure the CBWFQ policy **qppb**.

```
Device2(config)#policy-map qppb
Device2(config-pmap)#class rt_class
Device2(config-pmap-c)#bandwidth percent 30
Device2(config-pmap-c)#shape average percent 50
Device2(config-pmap-c)#exit
Device2(config-pmap)#exit
```

#On the interface, configure GTS, interface bandwidth, and apply the QoS policy.

```
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#traffic-shape 50000000
Device2(config-if-gigabitethernet0)#bandwidth 50000
Device2(config-if-gigabitethernet0)#service-policy output qppb
Device2(config-if-gigabitethernet0)#exit
```

Note:

- If the actual leased line bandwidth is inconsistent with the physical bandwidth of the interface, GTS needs to be configured on the interface. Otherwise, the packet may be discarded on the operator's line and the QoS control on the device will not work. If the actual leased line bandwidth is inconsistent with the configured bandwidth of the interface, it is generally necessary to modify **bandwidth** or **qos max-bandwidth** of the interface. If the Gigabit Ethernet port is connected to a 20M carrier MSTP line, a 20M GTS needs to be configured in the outgoing direction of the interface, and the bandwidth of the interface needs to be modified to 20M.
- Since the device QoS calculation is processed according to the length of the link-layer header plus the link layer data part, while the MSTP line provided by the operator is actually bundled by one or more 2M channels, and there is internal transmission overhead. According to the practical application experience, the overhead per packet is generally about 20 bytes. Therefore, if the operator's MSTP line is connected, **qos account output length add 20** needs to be configured to increase 20 bytes per packet when the device performs bandwidth calculation, which is in line with the actual situation of the operator's MSTP line and ensures that the packet is transmitted on the operator's MSTP line without packet loss.

#View the input queue statistics information.

```
Device2#show policy-map interface gigabitethernet 0
 interface gigabitethernet0
  Service-policy output: qppb
```

```
Class-map: rt_class (match-all)
  1466811 packets 187751808 bytes
  5 minute offered rate 89156400 bps
  match ip precedence 5
  match qos-group 1000
  Queueing
  queue limit 256 packets
  (queue depth/total drops) 255/1042826
  (packets output/bytes output) 423984/54269952
  Bandwidth: 30% (15000 Kbps)
  Shaping
  shape (average) cir 25000000, bc 5000000

Class-map: class-default (match-any)
  0 packets 0 bytes
  5 minute offered rate 0 bps
  Queueing
  queue limit 256 packets
  (queue depth/total drops) 0/0
  (packets output/bytes output) 0/0
  match any
```

From the above information, you can see the basic configuration information, the absolute value converted from bandwidth guarantee, the packet statistics and 5-minute traffic statistics matching each class, and the current QoS queue packet accumulation. You can simply judge whether the configuration is correct and effective.

1.3.13. Configure SPQ

Network Requirements

- There are many services in the network. The intranet bandwidth is 100M and the outlet bandwidth is 20M.
- It is required to give priority to the packet with DSCP value of AF31, and the guaranteed bandwidth is 5M. The second priority guarantees the packet with DSCP value of AF22, and the guaranteed bandwidth is 3M.

Network Topology

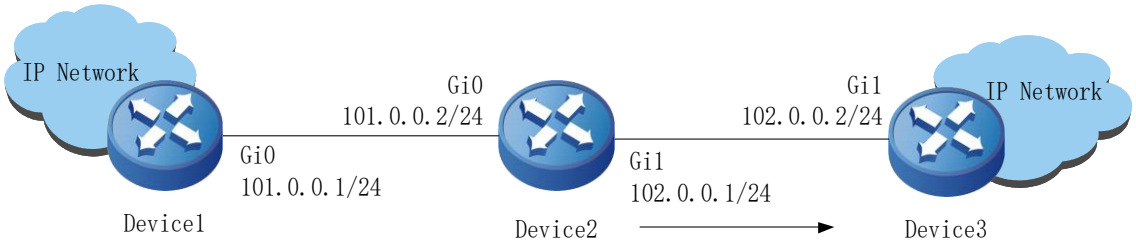

Figure 1-17 Networking of configuring SPQ

Configuration Steps

Step 1: Configure the IP address of the interface, and configure the route (omitted).

Step 2: Configure the interface SPQ policy.

```
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#spqueue 26 pq cir 5000
Device2(config-if-gigabitethernet1)#spqueue 20 pq cir 3000
Device2(config-if-gigabitethernet1)#traffic-shape 20000000 500000
Device2(config-if-gigabitethernet1)#bandwidth 20000
```

Note:

- If the actual leased line bandwidth is inconsistent with the physical bandwidth of the interface, GTS needs to be configured on the interface. Otherwise, the packet may be discarded on the operator's line and the QoS control on the device will not work. If the actual leased line bandwidth is inconsistent with the configured bandwidth of the interface, it is generally necessary to modify **bandwidth** or **qos max-bandwidth** of the interface. If the Gigabit Ethernet port is connected to a 20M carrier MSTP line, a 20M GTS needs to be configured in the outgoing direction of the interface, and the bandwidth of the interface needs to be modified to 20M.
- Since the device QoS calculation is processed according to the length of the link-layer header plus the link layer data part, while the MSTP line provided by the operator is actually bundled by one or more 2M channels, and there is internal transmission overhead. According to the practical application experience, the overhead per packet is generally about 20 bytes. Therefore, if the operator's MSTP line is connected, **qos account output length add 20** needs to be configured to increase 20 bytes per packet when the device performs bandwidth calculation, which is in line with the actual situation of the operator's MSTP line and ensures that the packet is transmitted on the operator's MSTP line without packet loss.

Step 3: Check the result.

#Check the SPQ queue in the gigabitethernet1 interface.

```
Device2#show queueing interface gigabitethernet1

SPQ statistics on Interface: gigabitethernet1
```

```
SPQ queue id: 20
-------------------------------------------------------------------------
----------
Queue_Depth   Hold-Pkts   Input-Pkts   Output-Pkts  Drop-Pkts
256        0        0         0         0
-------------------------------------------------------------------------
----------

SPQ queue id: 26
-------------------------------------------------------------------------
----------
Queue_Depth   Hold-Pkts   Input-Pkts   Output-Pkts  Drop-Pkts
256        0        0         0         0
-------------------------------------------------------------------------
----------

SPQ default queue
-------------------------------------------------------------------------
----------
Queue_Depth   Hold-Pkts   Input-Pkts   Output-Pkts  Drop-Pkts
256        0        0         0         0
-------------------------------------------------------------------------
----------
```

1.3.14. Configure CBWFQ to Match Application Identification

Network Requirements

- PC1 and PC2 are internal hosts, Device1 is a NAT device, and the IP route between devices is reachable. The intranet bandwidth is 100M and the outlet bandwidth is 20M.
- By configuring CBWFQ to match application identification on Device1, 60% low latency bandwidth is guaranteed for HTTP type services sent to the extranet and 30% bandwidth is guaranteed for FTP type services.

Network Topology

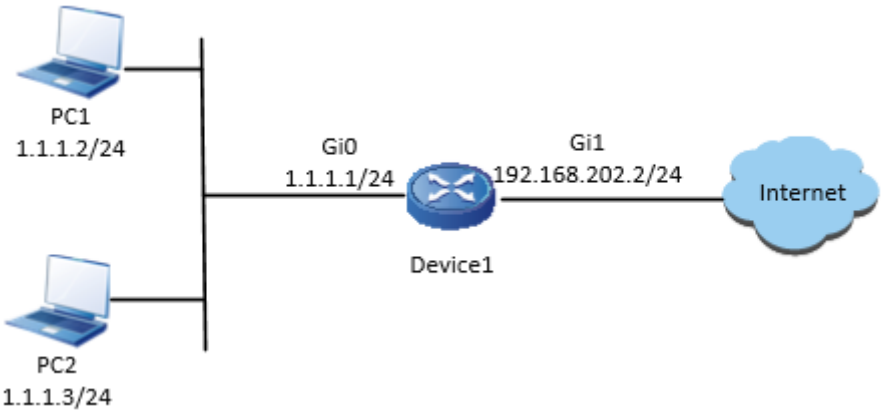

Figure 1-18 Networking of configuring CBWFQ to match the application identification

Configuration Steps

Step 1: Configure the IP address of the interface (omitted).

Step 2: Configure one gateway route to the gateway 192.168.202.254.

```
Device1#configure terminal
Device1(config)# ip route 0.0.0.0 0.0.0.0 192.168.202.254
```

Step 3: On Device1, configure the class of matching the application identification.

#Configure the class of matching the HTTP service.

```
Device1(config)#class-map http
Device1(config-cmap)#match application _http
Device1(config-cmap)#exit
```

#Configure the class of matching the FTP service.

```
Device1(config)#class-map ftp
Device1(config-cmap)#match application _ftp
Device1(config-cmap)#exit
```

Note:

- Here, _http and _ftp are the predefined application identification created by the device and can be used directly. You can also match customized application identification and application identification group. For commands, refer to "Application Identification and Visual Control AVC" in the chapter “Security” of the command manual.

Step 4: Configure the CBWFQ policy on Device1 to guarantee 60% low latency bandwidth for HTTP type services and 30% bandwidth for FTP type services.

```
Device1(config)#policy-map qos
Device1(config-pmap)#class http
```

```
Device1(config-pmap-c)#priority percent 60
Device1(config-pmap-c)#exit
Device1(config-pmap)#class ftp
Device1(config-pmap-c)#bandwisth percent 30
Device1(config-pmap-c)#exi
Device1(config-pmap)#exi
```

Note:

- The absolute value can also be used to configure bandwidth guarantee.

#Check the configured policy.

```
Device1#show policy-map user-config qos
Policy-map: qos
 class http
  priority percent 60
 class ftp
  bandwidth percent 30
```

Step 5: Configure the GTS, interface bandwidth and the policy of applying the configuration in the outgoing direction of Device1 interface.

```
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)#traffic-shape 20000000 500000
Device1(config-if-gigabitethernet1)#bandwidth 20000
Device1(config-if-gigabitethernet1)#service-policy output qos
Device1(config-if-gigabitethernet1)#exit
```

Note:

- If the actual leased line bandwidth is inconsistent with the physical bandwidth of the interface, GTS needs to be configured on the interface. Otherwise, the packet may be discarded on the operator's line and the QoS control on the device will not work. If the actual leased line bandwidth is inconsistent with the configured bandwidth of the interface, it is generally necessary to modify **bandwidth** or **qos max-bandwidth** of the interface. If the Gigabit Ethernet port is connected to a 20M carrier MSTP line, a 20M GTS needs to be configured in the outgoing direction of the interface, and the bandwidth of the interface needs to be modified to 20M.
- Since the device QoS calculation is processed according to the length of the link-layer header plus the link layer data part, while the MSTP line provided by the operator is actually bundled by one or more 2M channels, and there is internal transmission overhead. According to the practical application experience, the overhead per packet is generally about 20 bytes. Therefore, if the operator's MSTP line is connected, **qos account output length add 20** needs to be configured to increase 20 bytes per packet when the device performs bandwidth calculation, which is in line with the actual situation of the operator's MSTP line and ensures that the packet is transmitted on the operator's MSTP line without packet loss.

Step 6: View the input queue statistics information.

```
Device1#sho policy-map interface gigabitethernet 1
 interface gigabitethernet 1
  Service-policy output: qos

    Class-map: http (match-all)
      58 packets 44834  bytes
      5 minute offered rate 3624 bps
      match application _http
      Queueing
      queue limit 256 packets
      (queue depth/total drops) 0/0
      (packets output/bytes output) 58/44834
      Priority: 60% (12000 Kbps) , burst bytes 300000, b/w exceed drops: 0

    Class-map: ftp (match-all)
      1260 packets 640080 bytes
      5 minute offered rate 51720 bps
      match application _ftp
      Queueing
      queue limit 256 packets
      (queue depth/total drops) 1260/640080
      (packets output/bytes output) 0/0
      Bandwidth: 30% (6000 Kbps)

    Class-map: class-default (match-any)
      0 packets 0 bytes
      5 minute offered rate 0 bps
      Queueing
      queue limit 256 packets
      (queue depth/total drops) 0/0
      (packets output/bytes output) 0/0
```

From the above information, you can see the basic configuration information, the absolute value converted from bandwidth guarantee, the packet statistics and 5-minute traffic statistics matching each class, and the current QoS queue packet accumulation. You can simply judge whether the configuration is correct and effective.

1.4. Hardware QoS Function Configuration

1.4.1. Configure Traffic Monitoring

To ensure that the overload does not happen to the traffic passing the network and causes the congestion, the device provides the rate limitation based on the port receiving direction, limiting the total rate at the receiving direction of the port. The speeding traffic is dropped.

Configuration Condition

None

Configure Port-based Rate Limitation

To provide different rate limitations for ports at different time periods, each port is configured with eight rate limitations of different priorities. Each rate is limited and then bound to a time domain. For the entries taking effect at the same time, determine which entry takes effect by priority. The number 0 indicates the highest priority and the number 7 indicates the lowest priority. The rate limitation over the port can be configured directly without the time domain.

Table 1-18 Configure port-based rate limitation

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the L2 Ethernet interface configuration mode	**interface** *interface-name*	-
Configure port-based rate limitation	**rate-limit** { **default** *rate burst-size* | *priority rate burst-size* [**time-range** *time-range-name*] }	Mandatory By default, rate limitation over the port is not configured.

1.4.2. Configure Traffic Shaping

The traffic shaping enables the packets to be sent out at an average rate. The difference between the traffic shaping and traffic monitoring: the traffic monitoring takes effect in the ingress direction and the traffic shaping takes effect in the egress direction. The excessive traffic at the ingress direction will be dropped, but the excessive traffic at the egress direction will be cached.

Configuration Condition

None

Configure Port-based Traffic Shaping

The port-based traffic shaping allows the time domain binding to achieve different bandwidths in different time periods. Each port is configured with eight traffic shaping of different priorities and each traffic shaping is bound to a time domain. For the entries taking effect at the same time, determine which entry takes effect by priority. The number 0 indicates the highest priority and the number 7 indicates the lowest priority.

Table 1-19 Configure the port-based traffic shaping

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the L2 Ethernet interface configuration mode	**interface** *interface-name*	-
Configure the port-based traffic shaping	**traffic-shape** { *rate burst-size* | *priority rate burst-size* [**time-range** *time-range-name*] }	Mandatory By default, port-based traffic shaping is not configured.

1.5. Typical Configuration Example of Hardware QoS

1.5.1. Configure Rate Limitation

Network Requirements

- There are two servers in the network, that is, Video server and Data server.
- Configure the rate limitation function, and limit the total of the video traffic rate and the data traffic rate not to exceed 50000kbps. The data traffic rate does not exceed 20000kbps.

Network Topology

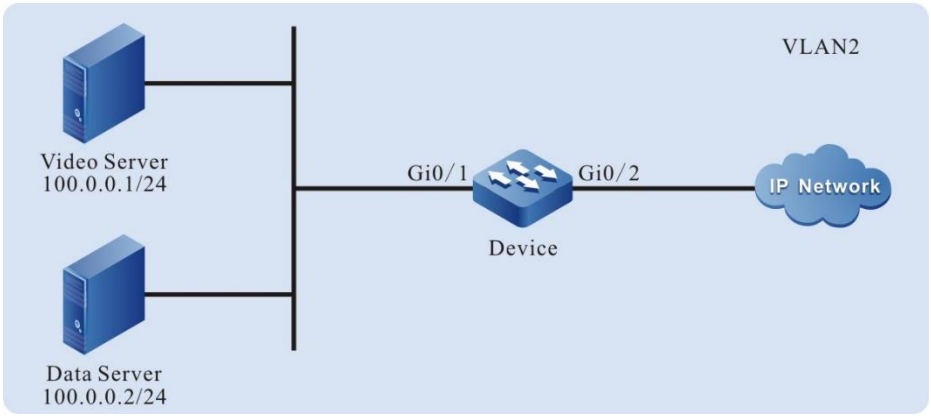

Figure 1-19 Networking of configuring the rate limitation

Configuration Steps

Step 1: Configure the link type of the VLAN and port on Device.

#Create VLAN2.

```
Device#configure terminal
Device(config)#vlan 2
```

```
Device(config-vlan2)#exit
```

#Configure the link type of port gigabitethernet0/1 as Access and permit the services of VLAN2 to pass.

```
Device(config)#interface gigabitethernet 0/1
Device(config-if-gigabitethernet0/1)#switchport mode access
Device(config-if-gigabitethernet0/1)#switchport access vlan 2
```

#Configure the link type of port gigabitethernet0/2 as Trunk and permit the services of VLAN2 to pass.

```
Device(config-if-gigabitethernet0/1)#interface gigabitethernet 0/2
Device(config-if-gigabitethernet0/2)#switchport mode trunk
Device(config-if-gigabitethernet0/2)#switchport trunk allowed vlan add 2
Device(config-if-gigabitethernet0/2)#exit
```

Step 2: Configure the rate limitation function.

#Configure the port-based rate limitation on port gigabitethernet0/1 and limit the traffic rate to 50000kbps.

```
Device(config)#interface gigabitethernet 0/1
Device(config-if-gigabitethernet0/1)#rate-limit default 50000 4096
Device(config-if-gigabitethernet0/1)#exit
```

Step 3: Check the result.

#After the video traffic and data traffic are processed by Device, the total of the video traffic rate and the data traffic sent out from port gigabitethernet0/2 does not exceed 50000kbps.

1.5.2. Configure Traffic Shaping

Network Requirements

- There are two servers in the network, that is, Video server and Data server.
- Configure the traffic shaping function; ensure that the total of the video traffic rate and the data traffic rate does not exceed 50000kbps.

Network Topology

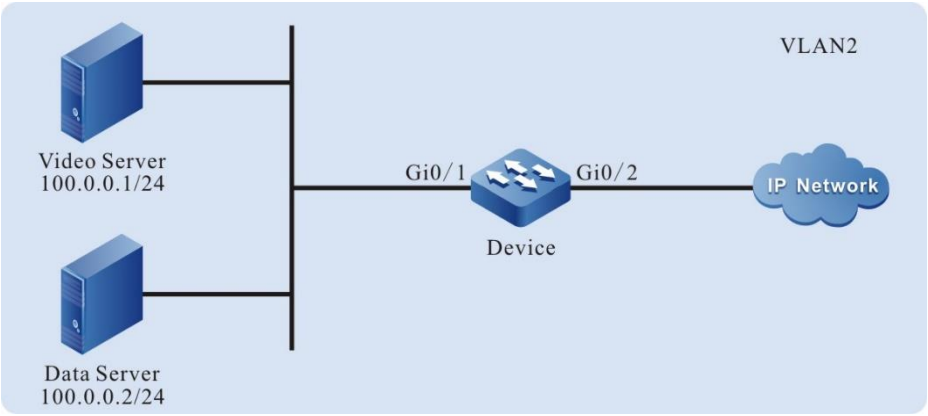

Figure 1-20 Networking of configuring the traffic shaping

Configuration Steps

Step 1: Configure the link type of the VLAN and port on Device.

#Create VLAN2.

```
Device#configure terminal
Device(config)#vlan 2
Device(config-vlan2)#exit
```

#Configure the link type of port gigabitethernet0/1 as Access and permit the services of VLAN2 to pass.

```
Device(config)#interface gigabitethernet 0/1
Device(config-if-gigabitethernet0/1)#switchport mode access
Device(config-if-gigabitethernet0/1)#switchport access vlan 2
```

#Configure the link type of port gigabitethernet0/2 as Trunk and permit the services of VLAN2 to pass.

```
Device(config-if-gigabitethernet0/1)#interface gigabitethernet 0/2
Device(config-if-gigabitethernet0/2)#switchport mode trunk
Device(config-if-gigabitethernet0/2)#switchport trunk allowed vlan add 2
Device(config-if-gigabitethernet0/2)#exit
```

Step 2: Configure the traffic shaping function.

#Configure the port-based traffic shaping on port gigabitethernet0/2 and limit the rate of the port traffic to 50000kbps.

```
Device(config-if-gigabitethernet0/2)#traffic-shape 50000 4096
Device(config-if-gigabitethernet0/2)#exit
```

Step 3: Check the result.

#After the video traffic and data traffic are processed by Device, the total of the video traffic rate and the data traffic rate sent out from port gigabitethernet0/2 does not exceed 50000kbps.

2. ОБЩАЯ ИНФОРМАЦИЯ

2.1. Замечания и предложения

Мы всегда стремимся улучшить нашу документацию и помочь вам работать лучше, поэтому мы хотим услышать вас. Мы всегда рады обратной связи, в особенности:

- ошибки в содержании, непонятные или противоречащие места в тексте;
- идеи по улучшению документации, чтобы находить информацию быстрее;
- неработающие ссылки и замечания к навигации по документу.

Если вы хотите написать нам по поводу данного документа, то используйте, пожалуйста, форму обратной связи на qtech.ru.

2.2. Гарантия и сервис

Процедура и необходимые действия по вопросам гарантии описаны на сайте QTECH в разделе «Поддержка» −> «Гарантийное обслуживание».

Ознакомиться с информацией по вопросам тестирования оборудования можно на сайте QTECH в разделе «Поддержка» −> «Взять оборудование на тест».

Вы можете написать напрямую в службу сервиса по электронной почте sc@qtech.ru.

2.3. Техническая поддержка

Если вам необходимо содействие в вопросах, касающихся нашего оборудования, то можете воспользоваться нашей автоматизированной системой запросов технического сервис-центра helpdesk.qtech.ru.

Телефон Технической поддержки +7 (495) 797-33-11 доб. 0