



HQoS

This document provides detailed understanding about HQoS and its configuration.

- [Feature History for HQoS, on page 1](#)
- [Information About HQoS, on page 1](#)
- [Prerequisites for HQoS, on page 2](#)
- [Restrictions for HQoS, on page 3](#)
- [Limitation for HQoS, on page 3](#)
- [How to Configure HQoS, on page 3](#)
- [Configuration Samples for HQoS, on page 3](#)

Feature History for HQoS

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Table 1: Feature History for HQoS

Release	Feature	Feature Information
Cisco IOS XE 26.2.1ea	HQoS Functionality: Hierarchical Quality of Service (HQoS) is a technology used to guarantee the bandwidth of multiple services of many users in the Differentiated Service (DiffServ) model through a queue scheduling mechanism.	Cisco C9550 Series Smart Switches

Information About HQoS

Information About HQoS

Hierarchical QoS (HQoS) on Cisco switches is a traffic management method that provides granular control by applying multiple levels of Quality of Service (QoS) policies. It allows for a more complex prioritization scheme than a single-level policy by using a hierarchy of policies, such as parent and child policies, to classify,

police, shape, and queue traffic based on different requirements. This allows you to manage bandwidth and prioritize traffic more effectively across different services and levels of your network.

Hierarchical policy structure

HQoS uses a multi-level structure, most commonly a two-level hierarchy of a parent and child policy.

Parent level

This is the top-level policy, often applied to a physical port, which manages overall bandwidth allocation and can have a child policy attached to it.

Child level

This is a nested policy that applies to specific traffic classes, which are identified at the parent level. The child policy can then perform detailed actions like queueing or marking for that specific traffic.

Granular traffic management

By nesting policies, HQoS allows for more sophisticated control. For example, a parent policy might shape the total bandwidth for a port, while a child policy within it can prioritize voice traffic over video traffic within that shaped bandwidth.

Support for multiple actions

HQoS supports performing different actions at different levels of the hierarchy. For example, you can have a parent policy that shapes traffic and a child policy that performs queueing.

Prerequisites for HQoS

Before configuring standard Hierarchical Quality of Service (QoS), you must have a thorough understanding of these items:

- Standard QoS concepts.
- Classic Cisco IOS QoS.
- Modular QoS CLI (MQC).
- Understanding of Virtual Output Queuing (VOQ) architecture.
- The types of applications used and the traffic patterns on your network.
- Traffic characteristics and needs of your network. For example, is the traffic on your network bursty? Do you need to reserve bandwidth for voice and video streams?
- Bandwidth requirements and speed of the network.
- Location of congestion points in the network.

Restrictions for HQoS

In a HQoS policy with parent shaping and child policy having priority level queuing and priority level policing, the statistics for policing are not updated. Only HQoS shaper statistics are updated. To view the HQoS shaper statistics, use the `show policy-map type queuing interface <interface name>` command in global configuration mode.

Limitation for HQoS

- For hierarchical QoS (HQoS) police applied in the ingress direction, child-level policing is supported. Policing of traffic is based on the police rate value defined in parent-level policing.
- Bandwidth remaining ratio is not supported for parent policy.

How to Configure HQoS

For information about child policy, refer to [How to Configure QoS](#).

Configuration Samples for HQoS

The following sections provide configuration examples for HQoS.

The configuration for Ingress HQoS is as follows:

```
policy-map child-policy
class cs1
set dscp cs2
class cs2
set dscp cs3
```



Note Child policy supports only remarking.

```
policy-map parent
class class-default
police cir 2g
service-policy child-policy
```



Note This policy applies to ingress HQoS.

Ingress HQoS output

The configuration output for ingress HQoS is as follows:

```
VTEP1(config-if)#do sh policy-map
int FortyGigabitEthernet6/0/2 in
FortyGigabitEthernet6/0/2
```

Service-policy input: parent

The configuration for service-policy input for a parent is as follows:

```
Class-map: class-default (match-any)
24982466 packets
Match: any
police:
cir 2000000000 bps, bc 62500000 bytes
conformed 19893033000 bytes; actions:
transmit
exceeded 11334205500 bytes; actions:
drop
conformed 418702000 bps, exceeded 238496000 bps
```

Service-policy: remark

The configuration for service-policy remark is as follows:

```
Device(config-if)#do sh policy-map int FiftyGigE1/0/30
FiftyGigE1/0/30
Service-policy input: parent
Class-map: class-default (match-any)
0 packets
Match: any
police:
cir 2000000000 bps, bc 10240 bytes
conformed 0 bytes; actions:
transmit
exceeded 0 bytes; actions:
drop
conformed 0000 bps, exceeded 0000 bps
Service-policy : remark
Class-map: cs1 (match-all)
0 packets
Match: dscp cs1 (8)
QoS Set
dscp cs3
Class-map: cs2 (match-all)
0 packets
Match: dscp cs2 (16)
QoS Set
dscp cs2
Class-map: cs5 (match-all)
0 packets
Match: dscp cs5 (40)
QoS Set
dscp cs3
Class-map: cs4 (match-all)
0 packets
Match: dscp cs4 (32)
QoS Set
dscp cs7
Class-map: cs3 (match-all)
0 packets
Match: dscp cs3 (24)
QoS Set
dscp cs3
Class-map: cs7 (match-all)
0 packets
```

```
Match: dscp cs7 (56)
QoS Set
dscp 1
Class-map: cs6 (match-all)
0 packets
Match: dscp cs6 (48)
QoS Set
dscp 3
Class-map: class-default (match-any)
0 packets
Match: any
Device(config-if)#
```

Child policy configuration

The configuration for queueing HQoS is as follows:

```
policy-map type queueing child-llq
class tc7
priority level 1
shape average 20g
class tc6
shape average percent 1
class tc5
shape average 15g
class tc4
shape average 5g
class tc3
shape average 15g
class tc2
shape average 15g
class tc1
shape average 4g
class class-default
shape average 10g
```

Parent policy configuration

The configuration for parent policy is as follows:

```
policy-map type queueing parent-llq
class class-default
shape average 80g
service-policy child-llq
```



Note The parent shape rate value must be greater than 1.35g. If it is less than 1.35g, the configuration will fail with an error message.

For information about child policy, refer to [How to Configure QoS](#).



Note For hierarchical QoS (HQoS) policy applied in the ingress direction, child-level policing is supported. Policing of traffic is based on the police rate value defined in parent-level policing.

HQoS configuration and interface statistics

To view the HQoS configuration and statistics on an interface, you can use the following CLI command:

```
show policy-map type queue interfae FiftyGigE1/0/3
```

The configuration output for HQoS configuration and statistics is as follows:

```
Device(config-if)#do sh policy-map ty q int FiftyGigE1/0/30
FiftyGigE1/0/30
Service-policy queueing output: hqos
Class-map: class-default (match-any)
5 packets
Match: any
Queueing
queue limit 98304 bytes
(total drops) 0
(bytes output) 619
shape (average) cir 3000000000, bc 12000000, be 12000000
target shape rate 3000000000
Service-policy queueing : cosllq
queue stats for all priority classes:
Queueing
priority level 1
queue limit 32256 bytes
(total drops) 0
(bytes output) 619
queue stats for all priority classes:
Queueing
priority level 2
queue limit 575999 bytes
(total drops) 0
(bytes output) 0
Class-map: tc7 (match-all)
2 packets
Match: traffic-class 7
shape (average) cir 1000000000, bc 400000, be 400000
target shape rate 1000000000
Priority: Strict,
Priority Level: 1
Class-map: tc6 (match-all)
0 packets
Match: traffic-class 6
shape (average) cir 1000000000, bc 4000000, be 4000000
target shape rate 1000000000
Priority: Strict,
Priority Level: 2
Class-map: tc5 (match-all)
0 packets
Match: traffic-class 5
Queueing
queue limit 98304 bytes
(total drops) 0
(bytes output) 0
shape (average) cir 30000000, bc 300000, be 300000
target shape rate 30000000
Class-map: tc4 (match-all)
0 packets
Match: traffic-class 4
Queueing
queue limit 98304 bytes
(total drops) 0
(bytes output) 0
shape (average) cir 1000000000, bc 4000000, be 4000000
target shape rate 1000000000
```

```
Class-map: tc3 (match-all)
 0 packets
Match: traffic-class 3
Queueing
queue limit 1000000 bytes
(total drops) 0
(bytes output) 0
shape (average) cir 1000000000, bc 4000000, be 4000000
target shape rate 1000000000
Class-map: tc2 (match-all)
 0 packets
Match: traffic-class 2
Queueing
queue limit 98304 bytes
(total drops) 0
(bytes output) 0
shape (average) cir 1000000000, bc 4000000, be 4000000
target shape rate 1000000000
Class-map: tc1 (match-all)
 0 packets
Match: traffic-class 1
Queueing
queue limit 98304 bytes
(total drops) 0
(bytes output) 0
shape (average) cir 1500000000, bc 6000000, be 6000000
target shape rate 1500000000
Class-map: class-default (match-any)
 3 packets
Match: any
Queueing
queue limit 98304 bytes
(total drops) 0
(bytes output) 0
shape (average) cir 100000000000, bc 400000000, be 400000000
target shape rate 100000000000
```

