Hierarchical QoS

This chapter includes details of hierarchical QoS.

- Information About Hierarchical QoS, on page 1
- Two-Level Hierarchical Policies, on page 1

Information About Hierarchical QoS

Hierarchical QoS allows you to specify QoS behavior at multiple policy levels, which provides a high degree of granularity in traffic management.

HQoS is not supported on Link Aggregation Group (LAG).

Two-Level Hierarchical Policies

Two-level hierarchical policies, also called nested polices, can be illustrated with a parent-level policy for the top level of the hierarchy and a child-level for the bottom level of the hierarchy. A two-level hierarchical policy can have queueing or marking or policing at child level and policing or shaping or bandwidth at parent level.

Four levels of priority are supported — priority level 1, 2, 3 and 4. These priority levels can be used along with the normal-priority queues. The normal-priority queues are scheduled by a different scheduler that does not give any priority treatment to the packets. Priority levels are supported only in the egress direction.

Whenever a policy with unsupported combination is applied, a failure message is displayed.

Configuring Hierarchical Policing

Hierarchical policing provides support at two levels:

- Parent level
- Child level
In the hierarchical ingress policy, **policer** command is supported at the parent level. In the hierarchical egress policy, **policer** command is not supported.

**Procedure**

### Step 1

**configure**

### Step 2

**policy-map policy-name**

**Example:**

```
RP/0/RP0:hostname(config)# policy-map policy1
```

Enters policy map configuration mode.

- Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.

### Step 3

**class default**

**Example:**

```
RP/0/RP0:hostname(config-pmap)# class default
```

Enters policy map class configuration mode.

### Step 4

**service-policy policy-map-name**

**Example:**

```
RP/0/RP0:hostname(config-pmap-c)# service-policy child
```

Attaches a policy map to an input or output interface to be used as the service policy for that interface.

### Step 5

**police rate percent percentage**

**Example:**

```
RP/0/RP0:hostname(config-pmap-c)# police rate percent 50
```

Configures traffic policing and enters policy map police configuration mode.

### Step 6

**end** or **commit**

**Example:**

```
RP/0/RP0:hostname(config-if)# end
```

or

```
RP/0/RP0:hostname(config-if)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
Entering `no` exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

Entering `cancel` leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the `commit` command to save the configuration changes to the running configuration file and remain within the configuration session.

---

**Configuration example for HQoS**

**Configuration example for Egress HQoS on L3 sub-interface**

class-map match-any CLASS_1_IPV4PREC
  match precedence 6
end-class-map

policy-map child_POLICY_
  class CLASS_1_IPV4PREC
    set qos-group 6
    police rate percent 17 peak-rate percent 25
end-policy-map

policy-map parent_POLICY
  class class-default
    service-policy child_POLICY
end-policy-map

interface TenGigE0/3/0/2
  service-policy input parent_POLICY
  ipv4 address 90.0.0.1 255.255.255.0

class-map match-any match_exp_4
match mpls experimental topmost 4
end-class-map
!
class-map match-any match_exp_5
match mpls experimental topmost 5
end-class-map
!
class-map match-any match_exp_1
match mpls experimental topmost 1
end-class-map
!
class-map match-any match_exp_2
match mpls experimental topmost 2
end-class-map
!
class-map match-any match_exp_3
match mpls experimental topmost 3
end-class-map
!
class-map match-any control_class_6
match dscp cs6
match precedence 6
match mpls experimental topmost 6
end-class-map
! class-map match-any control_class_7
  match dscp cs7
  match precedence 7
  match mpls experimental topmost 7
  end-class-map

class-map match-any match_qos_group1
  match qos-group 1
  end-class-map

class-map match-any match_qos_group2
  match qos-group 2
  end-class-map

class-map match-any match_qos_group3
  match qos-group 3
  end-class-map

class-map match-any match_qos_group4
  match qos-group 4
  end-class-map

class-map match-any match_qos_group5
  match qos-group 5
  end-class-map

class-map match-any match_qos_group6
  match qos-group 6
  end-class-map

class-map match-any match_qos_group7
  match qos-group 7
  end-class-map

policy-map policy_classify_exp_new
  class match_exp_1
    set qos-group 1
  class match_exp_2
    set qos-group 2
  class match_exp_3
    set qos-group 3
  class match_exp_4
    set qos-group 4
class match_exp_5
  set qos-group 5
!
class control_class_6
  set qos-group 6
!
class control_class_7
  set qos-group 7
!
class class-default
!
end-policy-map
!
policy-map hqos_child_policy_42xx
class match qos group1
  bandwidth 1 gbps
!
class match qos group2
  bandwidth 1 gbps
!
class match qos group3
  bandwidth 1 gbps
!
class match qos group4
  bandwidth 1 gbps
!
class match qos group5
  bandwidth 1 gbps
!
class match qos group6
  priority level 1
  police rate 1 gbps
!
!
class match qos group7
  priority level 2
  police rate 1 gbps
!
!
class class-default
!
end-policy-map
!
policy-map hqos_parent_policy_42xx
class class-default
  service-policy hqos_child_policy_42xx
  shape average 5 gbps
!
end-policy-map
!