# Quality of Service (QoS)

QoS is a set of standards and tools to ensure preferential delivery service for a class of traffic. This is being achieved by ensuring sufficient bandwidth, controlling latency and jitter, and loss characterstics to the selected class of traffic.

OS10 QoS tools help you classify data traffic based on various parameters in the received packet header. The switch examines parameters in the received packet, and then, based on the value of those parameters, places the packet into different classes of service by assigning a traffic class identifier.

QoS allows you to assign different QoS behaviors to different traffic classes. This QoS behaviors tells the device how to treat the traffic as it travels from ingress interface, all the way until it is sent out the egress interface of the switch. The different QoS behaviors you can assign to traffic classes are rate-limiting, rate-shaping, marking 802.1p or DSCP fields, queuing, and bandwidth management. QoS also helps you assign rate-limiting to CPU-directed control traffic.



# Classify Data Traffic

The first step to achieve the desired QoS behavior for a selected class of traffic is by differentiating or classifying the traffic. You can classify traffic based on Layer 2 (802.1p, 802.1q, and MAC addresses) or Layer 3 (DSCP or IP precedence, and IP addresses) header fields or based on an incoming port. The packet header fields chosen for matching are called *match criteria*. Classification can be done based on all or any of the match criterion.

OS10 provides a *qos* type to define class-maps for ingress classification of data traffic, based on the match criterion. OS10 allows you to mark or tag the classified traffic with a traffic class identifier (qos-group), if required. OS10 also provides *queuing* type to define class-maps for egress classification by matching the traffic class flow IDs (qos-group) created during ingress classification. For more information, see Configure Class-Map.

DELLEMC

# Classify Control Traffic

OS10 has pre-classified sets of control traffic towards the CPU, and each set is assigned a unique traffic class identifier. OS10 provides *control-plane* type to define class-maps for control-traffic classification by matching the preassigned traffic class flow IDs (qos-group). For more information, see Configure Class-Map.

# Define Policy

Once you have classified the traffic into a class of service, the next step is to define policies to apply on selected traffic classes. The policy is about the action (QoS behavior) that should be taken on the selected traffic classes. Some actions are executed while matched traffic travels ingress through the port. Such actions are called *ingress policy actions*. Allowed ingress policy actions in OS10 are marking 802.1p or DSCP, rate-limiting (policing), enable honoring of 802.1p (CoS), or DSCP field of incoming packets.

The policy actions which take place while matched traffic travels egress (out) are called *egress policy actions*. Allowed egress policy actions in OS10 are scheduling, rate-shaping, and WRED. You can define policies for a set of traffic classes with actions to be taken for each classes. OS10 provides *qos-type* policy-maps to define policies for traffic classes created by the qos-type class-maps. You can apply ingress policy actions only on qos-type of policies.

OS10 provides *queuing-type* policy-maps to define policies for traffic classes created by queuing-type class-maps. You can apply egress policy actions only on queuing-type of policies. OS10 provides queuing-type policy-maps to define policies for traffic classes created by queuing-type class-maps. You can apply egress policy actions only on queuing-type of policies. For more information, see Policy-Map Configuration.

Also provided are *control-plane type* policy-maps to define policies for traffic classes created by control-plane type class-maps. You can apply control-plane policy actions only on control-plane type of policies. The only available control-plane policy action is rate-limiting in this release.

# Apply Policy

Once you have defined the policy, the next step is to apply the policy to make the policy active. The control-plane policy can apply only on control-plane Configuration mode, where interfaces are not supported and it supports input direction alone. Policies defined for data traffic can apply for traffic on ingress or egress direction on an interface, or on all interfaces. This policy applies on system-qos Configuration mode for all interfaces.

Policies can apply on ingress (input) or egress (output) direction, depending on the type of action the policy refers to. Policies which contain ingress policy actions (qos-type policy) apply only in the input direction for an interface or system-qos mode. Policies which contain egress policy actions (queuing-type policy) apply only in the output direction of an interface or system-qos mode.

Topics:

- Configure QoS
- Configure Class-Map
- Policy-Map Configuration
- Interface Policy-Map
- Control-Plane Policy-Map
- System Policy-Map
- 802.1p Priorities on Ingress Traffic
- DSCP Queue Selection
- Strict Priority Queuing
- CoS or Dot1p Classification

# Configure QoS

OS10 processes network traffic based on classification and policies you create and apply to the traffic.
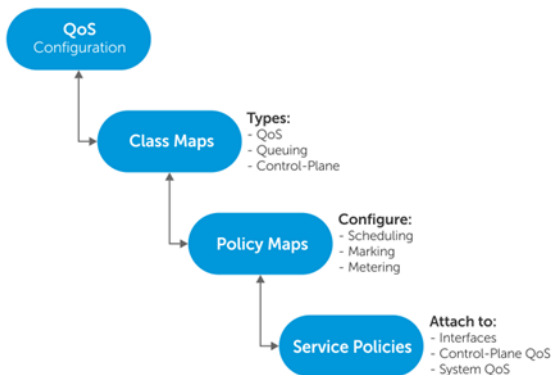


Before you start to configure QoS traffic classes and policies, see Quality of Service (QoS). Configuring QoS is a three-step process:

1  Create traffic classes based on the classification.
2  Create QoS policies and define actions to take on traffic classes.
3  Apply QoS policies to interface (port), system (all interfaces), or control-plane traffic.

System-qos or control-plane policies are special configuration modes, similar to interface configuration modes. Any QoS command in system-qos mode is applicable for all interfaces, and any command in control-plane mode is applicable for control-plane traffic.

DELLEMC

If the system has both L2 and L3 interfaces, you must apply an L2 policy-map only on the L2 interface. You must also apply an L3 policy-map only on L3 interface — if the system has both L2 and L3 interface.

If you apply a policy-map on system-qos with L2 parameters (match criteria or marking is only for L2 header fields) only, and both L2 and L3 interfaces are configured on that system, an error message displays. If you apply an L2 policy-map on an L3 configured interface, an error message displays.

# Configure Class-Map

You can implement classification or filtering packets into various traffic classes based on a packet match criteria using class-maps. OS10 allows you to create class-maps to separate packets based on a specific match criteria. You can configure three types of class-maps — control-plane, qos (default), and queueing.

- Create a class-map and configure a name for the class-map in CONFIGURATION mode.

  `class-map [type [control-plane | qos | queueing]] [match-all | match-any]class-map-name`

  - `qos` — Creates a QoS class-map type.
  - `queuing` — Creates a queuing class-map type.
  - `control-plane` — Creates a control-plane class-map type.
  - `match-all | match-any` — Sets `match-all` or `match-any` as your match filter. The default match filter is `match-any`.
  - `class-map-name` — Enter a class-map name (up to 32 characters).
- Save the configuration in CONFIGURATION mode.

  `do commit`

ⓘ **NOTE: If you create a class-map name before you enter the class-map type, `qos` is automatically set as the default class-map type.**

To apply classifications to class-maps, see Configure QoS Classifications.

**Configure Class-Map**

```
OS10(config)# class-map type qos match-any solar
OS10(config)# do commit
```

**View Class-Map**

```
OS10(config)# do show class-map
 Class-map (qos): solar (match-any)
```

# Policy-Map Configuration

Configure policy-maps to create a named object that represents a set of policies that apply to a set of traffic classes. You can configure three types of policy-maps — control-plane, qos (default), and queueing.

1. Create a policy-map and configure a name for the policy-map in CONFIGURATION mode (up to 32 characters).

   `policy-map [type {qos | queuing | control-plane}] policy-map-name`

   - `qos` — Creates a QoS type policy-map.
   - `queuing` — Creates a queuing type policy-map.
   - `control-plane` — Creates a control-plane type policy-map.
2. Associate a policy-map with a class-map in POLICY-MAP mode.

   `class class-name`
3. Save the configuration in POLICY-MAP mode.

   `do commit`

After creating policy-maps and associating the policy-maps with class-maps you can configure shaping, marking, metering, and other QoS features (see Mark Traffic and Meter Traffic).

**Create Class-Map C1**

```
OS10(config)# class-map c1
OS10(conf-cmap-qos)# match cos 1
OS10(conf-cmap-qos)# do commit
```

**Show Class-Map**

```
OS10(conf-cmap-qos)# do show class-map type qos c1
    Class-map (qos):  c1 (match-cos)
```

**Create Policy-Map P1 for Class C1**

```
OS10(config)# policy-map p1
OS10(conf-pmap-qos)# class c1
OS10(conf-pmap-c-qos)# set qos-group 1
OS10(conf-pmap-c-qos)# do commit
```

**Show Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: p1
   Class-map (qos): c1
     set qos-group 1
```

# Interface Policy-Map

Apply policy maps to interfaces.

1    Define the interface to attach a policy-map to in CONFIGURATION mode.

```
interface ethernet node/slot/port:[subport]
```

2    Configure an input or output service-policy in INTERFACE mode.

```
service-policy {[input type {qos}] | [output type {queuing}]} policy-map-name
```

3    Save the configuration in INTERFACE mode.

```
do commit
```

**Attach Policy-Map**

```
OS10(conf)# interface ethernet 1/1/1
OS10(conf-if-eth1/1/1)# service-policy input type qos p1
OS10(conf-if-eth1/1/1)# do commit
OS10(conf-if-eth1/1/1)# service-policy output type queuing p2
OS10(conf-if-eth1/1/1)# do commit
```

**Show Policy-Map**

```
OS10(conf-if-eth1/1/1)# do show policy-map
 Service-policy (qos) input: p1
   Class-map (qos): c1
     set qos-group 1
 Service-policy (queuing) output: p2
   Class-map (qos): c2
     set qos-group 2
```

**Show QoS Interface**

```
OS10(conf-if-eth1/1/1)# do show qos interface ethernet 1/1/1
   Interface ethernet 1/1/1
   unknown-unicast-storm-control : Disabled
   multicast-storm-control : Disabled
   broadcast-storm-control : Disabled
   flow-control-rx : Disabled
   flow-control-tx : Disabled
     Service-policy (Input)(qos): p1
     Service-policy (Output)(queuing): p2
```

# Control-Plane Policy-Map

Apply the policies you created to the control-plane.

1   Enter CONTROL-PLANE configuration mode from CONFIGURATION mode.
    ```
    control-plane
    ```
2   Apply the service policy and create a name for the policy-map in CONTROL-PLANE mode.
    ```
    service-policy input policy-map-name
    ```
3   Save the configuration in CONTROL-PLANE mode.
    ```
    do commit
    ```

**Attach Policy-Map to Control-Plane**

```
OS10(config)# control-plane
OS10(conf-control-plane)# service-policy input p1
OS10(conf-control-plane)# do commit
```

**View Control-Plane Service Policy**

```
OS10(conf-control-plane)# do show qos control-plane
  Service-policy (Input)(control-plane): p1
```

# System Policy-Map

Apply the policies you created to the system.

*   Enter SYSTEM-QOS mode from CONFIGURATION mode.
    ```
    system qos
    ```
*   Configure an input service-policy and create a name for the policy-map in SYSTEM-QOS mode.
    ```
    service-policy {[input type {qos}] | [output type {queuing}]} policy-map-name
    ```
*   Save the configuration in SYSTEM-QOS mode.
    ```
    do commit
    ```

**Attach Policy Map to System**

```
OS10(conf-sys-qos)# service-policy input type qos p1
OS10(conf-sys-qos)# do commit
OS10(conf-sys-qos)# service-policy output type queuing p2
OS10(conf-sys-qos)# do commit
```

**View Service Policies**

```
OS10(conf-sys-qos)# do show qos system
 Service-policy (input)(qos): p1
 Service-policy (output)(queuing): p2
```

# 802.1p Priorities on Ingress Traffic

By default, OS10 does not honor 802.1p priorities on ingress traffic. Honoring 802.1p means assigning a traffic-class ID implicitly based on incoming packets. You can use the `trust` command only under the ingress QoS policy-type, and under the reserved class-map name `class-trust` to enable honoring of 802.1p priorities on ingress traffic.

The table shows the recommended priority values for traffic-class mappings.

| Priority | Available Traffic Classes | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 (default) | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Priority | Available Traffic Classes | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 |
| 3 | 0 | 0 | 0 | 1 | 1 | 2 | 3 | 3 |
| 4 | 0 | 1 | 1 | 2 | 2 | 3 | 4 | 4 |
| 5 | 0 | 1 | 1 | 2 | 2 | 3 | 4 | 5 |
| 6 | 0 | 1 | 2 | 3 | 3 | 4 | 5 | 6 |
| 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

1  Create a class-map named *class-trust* in CONFIGURATION mode.

```
class class-trust
```

2  Return to CONFIGURATION mode.

```
exit
```

3  Create a policy-map and configure a name for the policy-map in CONFIGURATION mode.

```
policy-map type qos policy-map-name
```

4  Associate the class-map named *class-trust* with the policy-map in POLICY-MAP-CLASS-MAP mode.

```
class class-trust
```

5  Honor 802.1p (dot1p) priorities on ingress traffic in POLICY-MAP-CLASS-MAP mode.

```
trust {dot1p} [fallback]
```

- `dot1p` — Sets the dynamic classification to `trust dot1p`.
- `fallback` — (Optional) Honor trusting 802.1p (dot1p) only if other match criteria in this policy-map fails to qualify for a packet.

6  Save the configuration in POLICY-MAP mode.

```
do commit
```

### Honor 802.1p Priorities on Ingress Traffic

```
OS10(config)# class-map class-trust
OS10(conf-cmap-qos)# do commit
OS10(conf-cmap-qos)# exit
OS10(config)# policy-map policy-trust
OS10(conf-pmap-qos)# class class-trust
OS10(conf-pmap-c-qos)# trust dot1p
OS10(conf-pmap-c-qos)# do commit
```

### View Policy-Map

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: policy-trust
   Class-map (qos): class-trust
     trust dot1p
```

# DSCP Queue Selection

By default, OS10 does not honor DSCP values on ingress traffic. Honoring DSCP means assigning a traffic-class ID implicitly based on the incoming packets. You can use the `trust` command under the reserved class-map name `class-trust` to enable honoring DSCP.

The table shows the default DSCP values to queue mapping.

| DSCP/CP Hex Range (XXX)xxx | DSCP Definition | Traditional IP Precedence | Internal Queue ID | DSCP/CP Decimal (8–queue) |
|---|---|---|---|---|
| 111XXX | | Network Control | 7 | 56–63 |
| 110XXX | | Internetwork Control | 6 | 48–55 |
| 101XXX | EF (expedited forwarding) | CRITIC/ECP | 5 | 40–47 |

DELLEMC

| DSCP/CP Hex Range (XXX)xxx | DSCP Definition | Traditional IP Precedence | Internal Queue ID | DSCP/CP Decimal (8–queue) |
|---|---|---|---|---|
| 100XXX | AF4 (assured forwarding) | Flash Override | 4 | 32–39 |
| 011XXX | AF3 | Flash | 3 | 24–31 |
| 010XXX | AF2 | Immediate | 2 | 16–23 |
| 001XXX | AF1 | Priority | 1 | 8–15 |
| 000XXX | BE (best effort) | Best Effort | 0 | 0–7 |

1 Create a class-map named *class-trust* in CONFIGURATION mode.

```
class class-name
```

2 Return to CONFIGURATION mode.

```
exit
```

3 Create a policy-map and configure a name for the policy-map in CONFIGURATION mode.

```
policy-map [type qos] policy-map-name
```

4 Associate the class-trust class-map with the policy-map in POLICY-MAP-CLASS-MAP mode.

```
class class-trust
```

5 Honor incoming IP packets to classify this packet to a traffic-class ID in POLICY-MAP mode.

```
trust {diffserv} [fallback]
```

- `diffserv` — Sets the dynamic traffic-class to trust DSCP. If a policy-map has trust (dot1p or diffserv) enabled and has ACL-based classification, only trust-based classification is used.
- `fallback` — (Optional) Honor trusting DSCP only if other match criteria in this policy-map fails to qualify for a packet. If a policy-map has trust (dot1p or fallback) enabled and has ACL-based classification, the ACL-based classification receives priority over the trust-based classification — trust is the fallback mechanism for ACL classification conflicts.

6 Save the configuration in POLICY-MAP mode.

```
do commit
```

**Honor DSCP Priority on Ingress Traffic**

```
OS10(config)# class-map class-trust
OS10(conf-cmap-qos)# do commit
OS10(conf-cmap-qos)# exit
OS10(config)# policy-map policy-trust
OS10(conf-pmap-qos)# class class-trust
OS10(conf-pmap-c-qos)# trust diffserv
OS10(conf-pmap-c-qos)# do commit
```

**View Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy (qos) input: policy-trust
   Class-map (qos): class-trust
     trust diffserv
```

# Strict Priority Queuing

Queues are only for egress QoS policy-types. You can enable priorities to dequeue all packets from the assigned queue before servicing any other queues. When more than one queue is assigned strict priority, the highest number queue receives the highest priority. You can configure strict priority to any number of queues. By default, all queues schedule traffic per WDRR.

You can use the `priority` command to assign the priority to a single unicast queue — this configuration supersedes the `bandwidth percent` configuration. A queue with priority enabled can starve other queues in the same port-pipe.

1 Create a class-map and configure a name for the class-map in CONFIGURATION mode.

```
class-map type queuing class-map-name
```

2   Configure a match criteria in CLASS-MAP mode.

    ```
    match qos-group qos-group-id
    ```

3   Return to CONFIGURATION mode and save the configuration.

    ```
    exit
    do commit
    ```

4   Define a policy-map and create a policy-map name CONFIGURATION mode.

    ```
    policy-map type queuing policy-map-name
    ```

5   Create a QoS class and configure a name for the policy-map in POLICY-MAP mode.

    ```
    class class-map-name
    ```

6   Set the scheduler as the strict priority in POLICY-MAP-CLASS-MAP mode.

    ```
    priority
    ```

7   Return to CONFIGURATION mode and save the configuration.

    ```
    exit
    do commit
    ```

8   You can now apply the policy-map to the interface (INTERFACE mode) or all interfaces (SYSTEM-QOS mode).

    ```
    system qos
    ```

    OR

    ```
    interface ethernet node/slot/port[:subport]
    ```

9   Enter the output service-policy in SYSTEM-QOS mode or INTERFACE mode.

    ```
    service-policy {output} type {queuing} policy-map-name
    ```

10  Save the configuration in SYSTEM-QOS mode or INTERFACE mode.

    ```
    do commit
    ```

### Enable Strict Priority on Class Map

```
OS10(config)# class-map type queuing magnum
OS10(conf-cmap-queuing)# match qos-group 7
OS10(conf-cmap-queuing)# exit
OS10(config)# policy-map type queuing solar
OS10(conf-pmap-queuing)# class magnum
OS10(conf-pmap-c-que)# priority
OS10(conf-pmap-c-que)# exit
OS10(conf-pmap-queuing)# exit
OS10(config)# system qos
OS10(conf-sys-qos)# service-policy output solar
OS10(conf-sys-qos)# do commit
```

### View QoS System

```
OS10(conf-sys-qos)# do show qos system
 Service-policy (output)(queuing): solar
```

### Enable Strict Priority on Interface

```
OS10(config)# interface ethernet 1/1/5
OS10(conf-if-eth1/1/5)# service-policy output type queuing solar
OS10(conf-if-eth1/1/5)# do commit
```

### View Policy-Map

```
OS10(conf-if-eth1/1/5)# do show policy-map
 Service-policy(queuing) output: solar
   Class-map (queuing): magnum
     priority
```

DELLEMC

# CoS or Dot1p Classification

Classification allows you to manage network traffic by separating it into different categories. Packets are identified and categorized into traffic classes. You can use the `match` command to apply a match to place network traffic with specific traffic classes.

You can classify QoS traffic based on class of service (CoS) or 802.1p (dot1p) values. You cannot have two match statements with the same match criterion. If you enter two match statements with the same match criterion, the second statement overwrites the first statement.

1   Create a class-map and *qos* type in CONFIGURATION mode.

    ```
    class-map type qos class-map-name
    ```

2   Add the match criteria for CoS values in CLASS-MAP mode. Configure dot1p values of incoming packets to match in single, comma-delimited, or hyphenated range (0 to 7).

    ```
    match cos [cos-value | cos-range]
    ```

3   Return to CONFIGURATION mode.

    ```
    exit
    ```

4   Enter a policy-map name and *qos* type in CONFIGURATION mode.

    ```
    policy-map type qos policy-map-name
    ```

5   Associate the policy-map with a class-map in POLICY-MAP mode.

    ```
    class class-map-name
    ```

6   Enter a queue number for matched flow as qos-group ID in POLICY-MAP-CLASS-MAP mode (0 to 7).

    ```
    set qos-group queue-number
    ```

7   Save the configuration in POLICY-MAP-CLASS-MAP mode.

    ```
    do commit
    ```

**Configure CoS Classification**

```
OS10(config)# class-map type qos bluedot1p
OS10(conf-cmap-qos)# match cos 6
OS10(conf-cmap-qos)# exit
OS10(config)# policy-map type qos red
OS10(conf-pmap-qos)# class bluedot1p
OS10(conf-pmap-c-qos)# set qos-group 5
OS10(conf-pmap-c-qos)# do commit
```

**View Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: red
   Class-map (qos): bluedot1p
     set qos-group 5
```

# DSCP Classification

Classify traffic based on the DSCP value. The value of the QoS group represents the queue to which you direct a specific class of traffic. You cannot have two match statements with the same match criterion. If you enter two match statements with same filter-type, the second statement overwrites the first statement.

1   Create a class-map and enter the type as `qos` in CLASS-MAP mode.

    ```
    class-map type qos class-map-name
    ```

2   Apply an L3 DSCP match criteria and configure the DSCP values to match from incoming packets in CLASS-MAP mode. Enter DSCP values in single, comma-delimited or hyphenated range (0 to 63).

    - For IPv4 protocol use `match ip dscp [dscp-value | dscp-list]`
    - For IPv6 protocol use `match ipv6 dscp [dscp-value | dscp-list]`
    - For IPv4 and IPv6 protocols use `match ip-any dscp [dscp-value | dscp-list]`

> (i) **NOTE:** Use the `match not` command to match all valid values other than the configured values.

3   Return to CONFIGURATION mode.

    `exit`

4   Enter a policy-map name and type as `qos` in CONFIGURATION mode.

    `policy-map type qos policy—map-name`

5   Associate a policy-map with a class-map in POLICY-MAP mode.

    `class class-map-name`

6   Enter a queue number for matched flow as the qos-group ID in POLICY-MAP-CLASS-MAP mode (0 to 7).

    `set qos-group queue-number`

7   Save the configuration in POLICY-MAP-CLASS-MAP mode.

    `do commit`

### Configure DSCP Classification

```
OS10(config)# class-map type qos bluedscp
OS10(conf-cmap-qos)# match ip dscp 50
OS10(conf-cmap-qos)# exit
OS10(config)# policy-map type qos reddscp
OS10(conf-pmap-qos)# class bluedscp
OS10(conf-pmap-c-qos)# set qos-group 7
OS10(conf-pmap-c-qos)# do commit
```

### View Policy-Map

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: reddscp
   Class-map (qos): bluedscp
     set qos-group 7
```

# MAC Classification

Classify the QoS traffic based on the MAC access-list. For a match-all class-map, configure only one access-list as a match filter. You cannot apply any other match criteria. For a match-any class-map, configure up to five access-list and/or other match criteria.

For more information, see MAC Address ACLs.

1   Define a class-map type as `qos` in CLASS-MAP mode.

    `class—map type qos class—map-name`

2   Apply the match criteria for the MAC access-group in CLASS-MAP mode.

    `match mac access-group name access-group-name`

3   Save the configuration and return to CONFIGURATION mode.

    `do commit`
    `exit`

4   Define a policy-map type as `qos` and configure the policy-map name in CONFIGURATION mode.

    `policy-map type qos policy—map-name`

5   Associate the policy-map with a class-map in POLICY-MAP mode.

    `class class-map-name`

6   Enter a queue number for the matched flow as qos-group ID in POLICY-MAP-CLASS-MAP mode (0 to 7).

    `set qos-group queue-number`

7   Save the configuration in POLICY-MAP-CLASS-MAP mode.

    `do commit`

### Configure MAC Access Group Classification

```
OS10(config)# class-map type qos blueacl
OS10(conf-cmap-qos)# match mac access-group acl1
OS10(conf-cmap-qos)# do commit
OS10(conf-cmap-qos)# exit
```

```
OS10(config)# policy-map type qos redacl
OS10(conf-pmap-qos)# class blueacl
OS10(conf-pmap-c-qos)# set cos 6
OS10(conf-pmap-c-qos)# do commit
```

**View Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: redacl
   Class-map (qos): blueacl
     set cos 6
```

# IP Access-Group Classification

Classify the QoS traffic based on an IP access-group QoS policy. For a match-all class-map, configure only one access-lists as a match filter. You cannot apply any other match criteria. For a match-any class-map, configure up to five access-lists and/or other match criteria.

1   Define a class-map type as `qos` in CONFIGURATION mode.

    `class—map type qos class—map-name`

2   Apply the match criteria for an IPv4-specific QoS policy in CLASS-MAP mode.

    `match ip access-group name name`

3   Save the configuration and return to CONFIGURATION mode.

    ```
    do commit
    exit
    ```

4   Define a policy-map type as `qos` and create a name for the policy-map in CONFIGURATION mode.

    `policy-map type qos policy—map-name`

5   Associate a policy-map with a class-map in POLICY-MAP mode.

    `class class-map-name`

6   Enter a queue number for the matched flow as qos-group ID in POLICY-MAP-CLASS-MAP mode (0 to 7).

    `set qos-group queue-number`

7   Save the configuration in POLICY-MAP-CLASS-MAP mode.

    `do commit`

**Configure IP Access-Group Classification**

```
OS10(config)# class-map type qos accgrp
OS10(conf-cmap-qos)# match ip access-group name ag1
OS10(conf-cmap-qos)# do commit
OS10(conf-cmap-qos)# exit
OS10(config)# policy-map type qos ag2
OS10(conf-pmap-qos)# class accgrp
OS10(conf-pmap-c-qos)# set qos-group 6
OS10(conf-pmap-c-qos)# do commit
```

**View Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: ag2
   Class-map (qos): accgrp
     set qos-group 6
```

# IP Precedence Classification

Classify the QoS traffic based on an IP header precedence field. If DSCP-based classification — DSCP as match criterion — is used in a class-map, IP precedence cannot be used as another match criterion.

1   Create a class-map and type `qos` in CONFIGURATION mode.

    `class—map type qos class—map-name`

2   Apply the L3 precedence match criteria for the QoS policy in CLASS-MAP configuration mode. Configure the match IP precedence value as single, comma-delimited or hyphenated range (0 to 7).

- For IPv4 protocol use `match ip precedence [`*`precedence-value`* `|` *`precedence-list`*`]`
- For IPv6 protocol use `match ipv6 precedence [`*`precedence-value`* `|` *`precedence-list`*`]`

ⓘ | **NOTE: Use the `match not` command to match all values except the values configured.**

3  Save the configuration and return to CONFIGURATION mode.

```
do commit
exit
```

4  Enter a policy-map name and type `qos` in CONFIGURATION mode.

```
policy-map type qos policy—map-name
```

5  Associate a policy-map with a class-map in POLICY-MAP mode.

```
class class-map-name
```

6  Enter a queue number for matched flow as qos-group ID in POLICY-MAP-CLASS-MAP mode (0 to 7).

```
set qos-group queue-number
```

7  Save the configuration in POLICY-MAP-CLASS-MAP mode.

```
do commit
```

### Configure IP Precedence Classification

```
OS10(config)# class-map type qos bluedscp
OS10(conf-cmap-qos)# match ip precedence 5
OS10(conf-cmap-qos)# do commit
OS10(conf-cmap-qos)# exit
OS10(config)# policy-map type qos reddscp
OS10(conf-pmap-qos)# class bluedscp
OS10(conf-pmap-c-qos)# set qos-group  6
OS10(conf-pmap-c-qos)# do commit
```

### View Policy-Map

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: reddscp
   Class-map (qos): bluedscp
     set qos-group 6
```

# VLAN Classification

Classify traffic based on the VLAN ID to apply a specific QoS behavior.

1  Create a class-map of type `qos` and configure the class-map name in CONFIGURATION mode.

```
class—map type qos class—map name
```

2  Apply the match criteria as the VLAN ID in CLASS-MAP mode (1 to 4094).

```
match vlan vlan-id
```

3  Save the configuration and return to CONFIGURATION mode.

```
do commit
exit
```

4  Create a policy-map type as `qos` and configure the policy-name name in CONFIGURATION mode.

```
policy-map type qos policy—map-name
```

5  Associate a policy-map with a class-map in POLICY-MAP mode.

```
class class-name
```

6  Enter a queue number for the matched flow as qos-group ID in POLICY-MAP-CLASS-MAP mode (0 to 7).

```
set qos-group queue-number
```

7  Save the configuration in CONFIGURATION mode.

```
do commit
```

### Configure VLAN Classification

```
OS10(config)# class-map type qos bluevlan
OS10(conf-cmap-qos)# match vlan 1
```

```
OS10(conf-cmap-qos)# do commit
OS10(conf-cmap-qos)# exit
OS10(config)# policy-map type qos redvlan
OS10(conf-pmap-qos)# class bluevlan
OS10(conf-pmap-c-qos)# set qos-group 6
OS10(conf-pmap-c-qos)# do commit
```

**View Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: redvlan
   Class-map (qos): bluevlan
     set qos-group 6
```

# Mark Traffic

Marking allows you to add/modify a 802.1p priorities to a L2 header, or add a DSCP value to L3 header. You can mark classified traffic with a traffic-class ID (qos-group Id) as well. Since traffic-class ID (qos-group ID) and queue ID have a one-to-one mapping, you can use qos-group and queue interchangeably. Marking packet fields allows you to identify the traffic type based on the configured QoS information for a specific packet.

| Marking Field Names | Description |
|---|---|
| CoS (or dot1P Values) | Use the `set cos dot1p-values` command to mark the CoS field (0 to 7). |
| DSCP | Use the `set dscp dscp-values` command to mark the DSCP field (0 to 63). |
| QoS Group | Use the `set qos-group queue-number` command to mark the QoS Group field (0 to 11). |

# Mark CoS

To tag an incoming packet with 802.1p priorities, or modify incoming packets you can mark class of service (CoS). The `set cos` command is only supported under the ingress QoS policy type `qos`.

1  Create a policy-map of type `qos` and configure a name for the policy-map in CONFIGURATION mode.
   ```
   policy-map type qos policy-map-name
   ```
2  Configure a QoS class for classified traffic in POLICY-MAP mode (up to 32 characters).
   ```
   class class-name
   ```
3  Configure marking for CoS in POLICY-MAP-CLASS-MAP mode (0 to 7).
   ```
   set cos dot1p-value
   ```
4  Configure marking for QoS group in POLICY-MAP-CLASS-MAP mode.
   ```
   set qos group queue-number
   ```
5  Save the configuration in POLICY-MAP-CLASS-MAP mode.
   ```
   do commit
   ```

**Mark CoS**

```
OS10(config)# policy-map type qos platinum
OS10(conf-pmap-qos)# class diamond
OS10(conf-pmap-c-qos)# set cos 5
OS10(conf-pmap-c-qos)# set qos-group 7
OS10(conf-pmap-c-qos)# do commit
```

**View Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: platinum
   Class-map (qos): diamond
```

```
    set cos 5
    set qos-group 7
```

# Mark DSCP

To tag an incoming packet with a DSCP value, or modify incoming packets, you can configure marking for DSCP. The `set dscp` command is only supported under the ingress QoS policy type `qos`.

1   Create a policy-map of type `qos` and configure a name for the policy-map in CONFIGURATION mode.

    ```
    policy-map type qos policy-map-name
    ```

2   Configure a QoS class for classified traffic in POLICY-MAP mode (up to 32 characters).

    ```
    class class-name
    ```

3   Configure marking for DSCP in POLICY-MAP mode (0 to 63).

    ```
    set dscp dscp-value
    ```

4   Save the configuration in POLICY-MAP-CLASS-MAP mode.

    ```
    do commit
    ```

**Mark DSCP**

```
OS10(config)# policy-map type qos platinum
OS10(conf-pmap-qos)# class diamond
OS10(conf-pmap-c-qos)# set dscp 50
OS10(conf-pmap-c-qos)# set qos-group 7
OS10(conf-pmap-c-qos)# do commit
```

**View Policy-Map**

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy(qos) input: platinum
   Class-map (qos): diamond
     set dscp 50
     set qos-group 7
```

# Mark Group

To tag an incoming packet with qos-group type, you can configure marking for the QoS group. The `set qos-group` command is only supported under ingress qos type or control-plane. If the class-map type is control-plane, the qos-group corresponds to CPU queues 0 to 11. If the class-map type is qos, the qos-group corresponds to data queues 0 to 7. The maximum number of data queues supported in OS10 are 8, and the maximum number of control traffic queues are 12.

1   Create a policy-map with the type `qos` in CONFIGURATION mode.

    ```
    policy-map type qos policy-map-name
    ```

2   Configure a QoS class in POLICY-MAP mode.

    ```
    class class-name
    ```

3   Configure marking for the QoS group in POLICY-MAP-CLASS-MAP mode.

    ```
    set qos-group queue-number
    ```

4   Save the configuration in POLICY-MAP-CLASS-MAP mode.

    ```
    do commit
    ```

**Mark QoS Group**

```
OS10(config)# policy-map type qos platinum
OS10(conf-pmap-qos)# class diamond
OS10(conf-pmap-c-qos)# set qos-group 7
OS10(conf-pmap-c-qos)# do commit
```

**View QoS Policy-Map**

```
OS10(conf-pmap-c)# do show policy-map
 Service-policy(qos) input: platinum
```

DELLEMC

```
      Class-map (qos): diamond
         set qos-group 7
```

**Mark Control-Plane**

```
OS10(config)# class-map type control-plane copp
OS10(conf-cmap-control-plane)# exit
OS10(config)# policy-map type control-plane copp1
OS10(conf-pmap-control-plane)# class copp
OS10(conf-pmap-c)# set qos-group 2
OS10(conf-pmap-c)# police cir 100 pir 100
OS10(conf-pmap-c)# do commit
```

**View Control-Plane Policy-Map**

```
OS10(conf-pmap-c)# do show policy-map
 Service-policy(control-plane) input: copp1
   Class-map (control-plane): copp
      set qos-group 2
      police cir 100 bc 100 pir 100 be 100
```

# Meter Traffic

Metering applies to shaping and policing network traffic and ensures better service for traffic. OS10 includes congestion management tools to raise the priority of a flow by queuing and servicing queues in different ways. The queue management mechanism used for congestion avoidance raises the priority by dropping traffic from lower-priority flows before traffic from higher-priority flows. Policing and shaping provides priority to a flow by limiting the throughput of other flows.

You can configure a guaranteed bandwidth percentage by examining for the egress out flows on the queue. For example, if you have identified three different flows which are egressing out of an interface, you can configure the bandwidth ratio for the flows as 3:2:1 which means:

- 1st flow — 50
- 2nd flow — 33.3
- 3rd flow — 16.66

The configuration is then:

- Bandwidth percent 50 = bandwidth percent 3
- Bandwidth percent 33 = bandwidth percent 2
- Bandwidth percent 17 = bandwidth percent 1

# Bandwidth Allocation

You can allocate relative bandwidth to limit large flows and show a preference for smaller flows. Allocate the relative amount of bandwidth to non-priority queues when priorities queues are consuming maximum link bandwidth. Configure the bandwidth percentage to guarantee allocation for non-priority queues. If the `bandwidth percent` command is present, you cannot configure the `priority` command. The `bandwidth percent` command is only supported on the egress QoS policy type.

1. Create a class-map of type `queuing` and configure a name for the class-map in CONFIGURATION mode.
   ```
   class-map type queuing class-map-name
   ```
2. Apply the match criteria for the QoS group in CLASS-MAP mode.
   ```
   qos-group queue-number
   ```
3. Save the configuration and return to CONFIGURATION mode.
   ```
   do commit
   exit
   ```
4. Create a policy-map of type `queuing` and configure a policy-map name in CONFIGURATION mode.
   ```
   policy-map type queuing policy-map-name
   ```
5. Configure a queuing class in POLICY-MAP mode.
   ```
   class class-name
   ```

6    Assign a bandwidth percent (1 to 100) to non-priority queues in POLICY-MAP-CLASS-MAP mode.

     `bandwidth percent value`

7    Save the configuration in POLICY-MAP-CLASS-MAP mode.

     `do commit`

### Configure Bandwidth Allocation

```
OS10(config)# class-map type queuing solar
OS10(conf-cmap-queuing)# match qos-group 5
OS10(conf-cmap-queuing)# do commit
OS10(conf-cmap-queuing)# exit
OS10(config)# policy-map lunar
OS10(config)# policy-map type queuing lunar
OS10(conf-pmap-queuing)# class solar
OS10(conf-pmap-c-que)# bandwidth percent 80
OS10(conf-pmap-c-que)# do commit
```

### View Class-Map

```
OS10(conf-cmap-queuing)# do show class-map
 Class-map (queuing): solar (match-any)
   Match: qos-group 5
```

### View Policy-Map

```
OS10(conf-pmap-c-que)# do show policy-map
 Service-policy (queuing) output: solar
   Class-map (queuing): lunar
     bandwidth percent 80
```

# Policy-Based Rate Shaping

Rate shaping buffers traffic exceeding the specified rate until the buffer is exhausted. If any stream exceeds the configured bandwidth on a continuous basis, it can consume all the buffer space that is allocated to the port.

1    Enter the policy-map type as `queuing` and configure a policy-map name in CONFIGURATION mode.

     `policy-map type queuing policy-map-name`

2    Enter a class name to apply to the shape rate in POLICY-MAP-QUEUEING mode (up to 32 characters).

     `class class-name`

3    (Optional) If you need rate shaping on a specific queue, match the corresponding qos-group in the class-map. If you do not configure the `match qos-group` command, rate shaping applies to all queues.

     `match qos-group queue-number`

4    Enter a minimum and maximum shape rate value in POLICY-MAP-QUEUEING-CLASS mode.

     `shape {min {kbps | mbps | pps}min-value} {max {kbps | mbps | pps}max-value}`

     - 0 to 40000000 — kilobits per second (kbps)
     - 0 to 40000 — megabits per second (mbps)
     - 1 to 268000000 — in packets per second (pps)

5    Save the configuration in POLICY-MAP-QUEUEING-CLASS mode.

     `do commit`

### Policy-Based Shaping

```
OS10(config)# policy-map type queuing master
OS10(conf-pmap-queuing)# class first
OS10(conf-pmap-c-que)# shape min pps 10 max pps 40
OS10(conf-pmap-c-que)# do commit
```

### View Policy-Map

```
OS10(conf-pmap-c-que)# do show policy-map
 Service-policy(queuing) output: master
```

```
Class-map (queuing): first
   shape min pps 10 max pps 40
```

# Policy-Based Rate Policing

You can configure traffic rate-limiting in packets per second (pps) for a QoS input policy, and a rate policing value in kilobits per second (kbps) or pps. Committed rate guarantees bandwidth for traffic entering or leaving the interface under normal network conditions.

When traffic propagates at an average rate that is less than or equal to the committed rate, it is green colored or coded. For a system that does not have ingress buffers, OS10 performs rate-limiting on the incoming traffic stream. The traffic rate above the configured committed rate is tail dropped (which means if the queue is full the packets are dropped) to guarantee a fixed bandwidth for an ingress traffic flow.

When the transmitted traffic falls below the committed rate, the bandwidth, which is not used by any traffic that is traversing the network, aggregates to form the committed burst size. Traffic is green-coded up to the point at which the unused bandwidth does not exceed the committed burst size.

Peak rate is the maximum rate for traffic arriving or exiting an interface under normal traffic conditions. Peak burst size indicates the maximum size of unused peak bandwidth that is aggregated. This aggregated bandwidth enables brief durations of burst traffic that exceeds the peak rate and committed burst.

1   Create the policy-map type as `qos` and configure a name for the policy-map in CONFIGURATION mode.

`policy-map type qos policy-map-name`

2   Enter a class name to apply the shape rate in POLICY-MAP mode.

`class class-map-name`

3   Configure traffic policing on incoming traffic in POLICY-MAP-CLASS-MAP mode.

`police {cir committed-rate [bc committed-burst-size]} {pir peak-rate [be peak-burst-size]}`

·   `cir committed-rate` — Enter a committed rate value in kilobits per second (kbps) (0 to 40000000).
·   `bc committed-burst-size` — (Optional) Enter a committed burst size in packets for control plane and kbps (16 to 200000, default 200).
·   `pir peak-rate` — Enter a peak-rate value in kbps (0 to 40000000).
·   `be peak-burst-size` — (Optional) Enter a peak burst size in kbps (16 to 200000, default 200).

4   (Optional) Configure traffic policing for a specific queue in POLICY-MAP-CLASS-MAP mode. Queue number range is from 0 to 7 for qos policy map and 0 to 11 for control-plane policy map.

`set qos-group queue-number`

5   Save the configuration in POLICY-MAP-CLASS-MAP mode.

`do commit`

### Configure Policy-Based Rate Policy

```
OS10(config)# policy-map type qos galaxy
OS10(conf-pmap-qos)# class bigbang
OS10(conf-pmap-c-qos)# police cir 5 bc 30 pir 20 be 40
OS10(conf-pmap-c-qos)# do commit
```

### Configure Rate Policing on Specific Queue

```
OS10(config)# policy-map bronze
OS10(conf-pmap-qos)# class silver
OS10(conf-pmap-c-qos)# set qos-group 7
OS10(conf-pmap-c-qos)# police cir 5 pir 30
OS10(conf-pmap-c-qos)# do commit
```

### View Policy-Map

```
OS10(conf-pmap-c-qos)# do show policy-map
 Service-policy (qos) input: galaxy
   Class-map (qos): bigbang
     police cir 5 bc 30 pir 20 be 40
```

```
Service-policy (qos) input: bronze
  Class-map (qos): silver
    police cir 5 bc 100 pir 30 be 100
```

# Control Plane Policing (CoPP)

CoPP increases security on the system by protecting the route processor from unnecessary traffic and giving priority to important control plane and management traffic. CoPP uses a dedicated control plane configuration through the QoS CLIs to provide filtering and rate-limiting capabilities for the control plane packets.

If the rate of control packets towards the CPU is higher than it can handle, CoPP provides a method to selectively drops some of the control traffic so the CPU can process high-priority control traffic. You can use CoPP to rate-limit traffic through each CPU port queue of the NPU.

CoPP applies policy actions on all control-plane traffic. The control-plane class map does not use any match criteria. Create policy maps to enforce rate-limiting or rate policing on control-plane traffic. You can use the `control-plane` command to attach the CoPP service policies directly to the control-plane.

The default rate limits apply to 12 CPU queues and the protocols mapped to each CPU queue. The control packet type to CPU ports control queue assignment is fixed. The only way you can limit the traffic towards the CPU is choose a low priority queue, and apply rate-limits on that queue to find a high rate of control traffic flowing through that queue.

| CPU Queue | Protocols Mapped to Control Processor Queues | Rate Limit (in pps) for S6000 and S4000 | Rate Limit (in pps) for S3000 |
|---|---|---|---|
| 0 | | 600 | 1300 |
| 1 | | 1000 | 300 |
| 2 | | 300 | 300 |
| 3 | VLT NDS | 1300 | 400 |
| 4 | ICMPv4 ICMPv6 | 2000 | 2000 |
| 5 | ARP-REQ ICMP-V6–RS-NS | 400 | 300 |
| 6 | SSH, TELNET, TACACS, NTP, FTP, ARP-REPLY, ICMP-V6–RA-NA | 600 | 600 |
| 7 | RSTP, PVST, MSTP, LACP | 400 | 400 |
| 8 | DOT1X, LLDP | 600 | 400 |
| 9 | BGP-V4, OSPF-V4, OSPF-V6 | 600 | 600 |
| 10 | DHCP-V4, DHCP-V6, SERVICEABILITY, VRRP | 600 | 300 |
| 11 | OPEN-FLOW | 300 | 300 |

# Configure CoPP

Rate-limiting the protocol CPU queues requires configuring control-plane type QoS policies.

- Create QoS policies (class maps and policy maps) for the desired CPU-bound queue.
- Associate the QoS policy with a particular rate-limit.
- Assign the QoS service policy to control plane queues.

By default, the `pir` and `cir` values are in pps for control plane. CoPP for CPU queues converts the input rate from kbps to pps, assuming 64 bytes is the average packet size, and applies that rate to the corresponding queue – 1 kbps is roughly equivalent to 2 pps.

1    Create a class-map of type `control-plane` and configure a name for the class-map in CONFIGURATION mode.
     `class-map type control-plane class-map-name`

2   Save the configuration and return to CONFIGURATION mode.

```
do commit
exit
```

3   Create an input policy-map to assign the QoS policy to the desired service queues in CONFIGURATION mode.

```
policy-map type control-plane policy-map-name
```

4   Associate a policy-map with a class-map in POLICY-MAP mode.

```
class class-name
```

5   Configure marking for a specific queue number in POLICY-MAP-CLASS-MAP mode (0 to 11).

```
qos group queue-number
```

6   Configure rate policing on incoming traffic in POLICY-MAP-CLASS-MAP mode.

```
police {cir committed-rate | pir peak-rate}
```

   · `cir committed-rate` — Enter a committed rate value in pps (0 to 4000000).

   · `pir peak rate` — Enter a peak-rate value in pps (0 to 40000000).

7   Save the configuration in POLICY-MAP-CLASS-MAP mode.

```
do commit
```

### Create QoS Policy for CoPP

```
OS10(config)# class-map type control-plane copp
OS10(conf-cmap-control-plane)# do commit
OS10(conf-cmap-control-plane)# exit
OS10(config)# policy-map type control-plane copp1
OS10(conf-pmap-control-plane)# class copp
OS10(conf-pmap-c)# set qos-group 2
OS10(conf-pmap-c)# police cir 100 pir 100
OS10(conf-pmap-c)# do commit
```

### View Policy-Map

```
OS10(conf-pmap-c)# do show policy-map
 Service-policy(control-plane) input: copp1
   Class-map (control-plane): copp
     set qos-group 2
     police cir 100 bc 100 pir 100 be 100
```

# Assign Service Policy

Controlling traffic and rate the protocol CPU queues requires configuring QoS policies. To enable CoPP, you need to apply the defined policy-map to control-plane configuration mode.

1   Enter CONTROL-PLANE mode from CONFIGURATION mode.

```
control-plane
```

2   Define a service-policy of type `input` and configure a name for the service policy in CONTROL-PLANE mode.

```
service-policy input service-policy-name
```

3   Save the configuration in CONTROL-PLANE mode.

```
do commit
```

### Assign Control-Plane Service-Policy

```
OS10(config)# control-plane
OS10(conf-control-plane)# service-policy input copp1
OS10(conf-control-plane)# do commit
```

### View Control-Plane Service-Policy

```
OS10(conf-control-plane)# do show qos control-plane
 Service-policy (input): copp1
```

# View CoPP Configuration

Use the `show` commands to display the protocol traffic assigned to each control-plane queue and the current rate-limit applied to each queue. You can also use the `show` command output to verify the CoPP configuration.

**View CoPP Configuration**

```
OS10# show qos control-plane
 Service-policy (input): pmap1
```

**View CMAP1 Configuration**

```
OS10# show class-map type control-plane cmap1
 Class-map (control-plane): cmap1 (match-any)
```

**View CoPP Service Policy**

```
OS10# show policy-map type control-plane

 Service-policy(control-plane) input: pmap1
   Class-map (control-plane): cmap1
     set qos-group 6
     police cir 200 bc 100 pir 200 be 100
```

**View CoPP Information**

```
OS10# show control-plane info

Queue      Rate Limit(in pps)   Protocols
0          600
1          1000
2          300
3          1300
4          2000                 VLT NDS
5          400                  ARP_REQ IPV6_ICMP_REQ
6          400                  ARP_RESP IPV6_ICMP IPV6_ICMP_RESP IPV4_ICMP SSH TELNET TACACS NTP
FTP
7          400                  RSTP PVST MSTP LACP
8          600                  DOT1X LLDP
9          600                  IPV6_OSPF IPV4_BGP IPV4_OSPF
10         600                  IPV6_DHCP IPV4_DHCP SERVICEABILITY
11         300                  OPEN_FLOW
```

**View CoPP Statistics**

```
OS10# show control-plane statistics
Queue    Packets    Bytes       Dropped Packets   Dropped Bytes
0        0          0           0                 0
1        0          0           0                 0
2        0          0           0                 0
3        0          0           0                 0
4        0          0           0                 0
5        2          172         0                 0
6        0          0           0                 0
7        32048      2180484     0                 0
8        14140      2569184     0                 0
9        0          0           0                 0
10       0          0           0                 0
11       0          0           0                 0
```

# WRED and ECN

The weighted random early detection (WRED) congestion avoidance mechanism drops packets to prevent buffering resources from being consumed. Network traffic is a mixture of packets of different traffic types or flows, and the rate of some types of traffic is greater than others.

The packet buffer resources (ingress and egress buffers) are consumed by only one or a few types of traffic, leaving no space for other types. Apply WRED threshold values to a policy-map so that configured traffic is prevented from consuming large amounts of BTM resources.

Configure WRED parameters for a queue and configure the minimum threshold drop rate using the `random-detect` command. The minimum threshold is the allowed buffer space for the specified traffic — for example, 1000 KB on egress. If the 1000 KB is consumed, packets drop randomly at an exponential rate until the maximum threshold is reached — this is the "early detection" part of WRED.

Before queuing the packet, OS10 assigns a color (also called drop precedence or DP) — yellow or green — to each packet using the `color {yellow | green}` command based on the packet's DSCP value. DSCP is a 6-bit field (bits 2 to 4 of the ToS field), and OS10 uses the first three bits (LSB) of this field (DP) to determine the drop precedence. The last three bits of the DSCP field are the drop precedence bits.

If the maximum threshold (for example 2000 KB) is reached, all incoming packets drop until the buffer space reduces to below 2000 KB of the specified traffic.
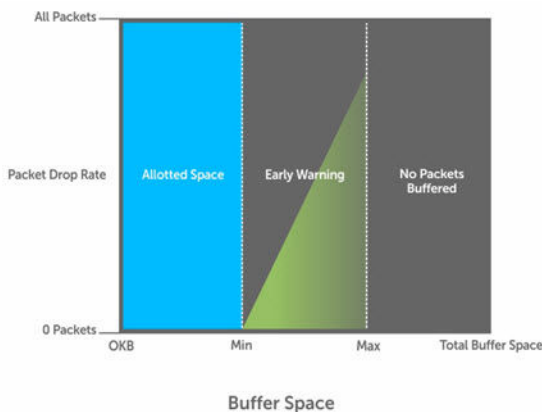


## Explicit Congestion Notification (ECN)

When you use ECN in conjunction with WRED, packets are marked as per RFC3168 instead of dropping them. Devices on a network respond to congestion before a queue overflows and packets drop, enabling improved queue management. You can configure ECN using the `ecn` command.

When a packet reaches the device with ECN enabled for WRED, the average queue size is computed. To measure the average queue size, a user-configurable weight factor is used. Use the `weight value` command to configure the weight for the WRED average queue size. The weight factor is set to zero by default.

## WRED Queue Management

Queues are allocated with shared buffers. The minimum threshold specifies the number of packets in a queue before the queue considers discarding packets. The probability of discard increases until the queue depth reaches the maximum threshold. After a queue depth exceeds the maximum threshold, all other packets that attempt to enter the queue are discarded.

1  Enter the policy-map name and type in CONFIGURATION mode.
```
policy-map policy-map-name [type {qos | queuing | control-plane}]
```

2    Enter WRED parameters for a queue and enter a minimum and maximum threshold drop rate in POLICY-MAP mode (1 to 12480 kbps).

```
random-detect minimum-threshold threshold-value maximum-threshold threshold-value
```

- Enter the drop probability and enter the maximum drop rate in POLICY-MAP mode (1 to 100).

```
drop probability percent
```

- Assign a yellow or green color drop precedence to the traffic in POLICY-MAP mode. `yellow` configures a medium drop precedence, and `green` configures a low drop precedence (default).

```
color {yellow | green}
```

- Configure explicit congestion notification (ECN) in POLICY-MAP mode.

```
ecn
```

- Enter the weight factor to compute average-queue size in POLICY-MAP mode (1 to 15, default 0).

```
weight value
```

3    Save the configuration in POLICY-MAP mode.

```
do commit
```

**Configure WRED Queue Management**

```
OS10(config)# policy-map type queuing gold
OS10(conf-pmap-queuing)# class silver
OS10(conf-pmap-c-que)# random-detect minimum-threshold 10 maximum-threshold 100 drop-
probability 50 color yellow ecn weight 10
OS10(conf-pmap-c-que)# do commit
```

# View Configured WRED Profiles

You can view WRED profiles and their threshold values. If you configure WRED profiles to a queue, you view the count of GREEN, YELLOW or RED drop packet counts for the interface.

**View Configured WRED Profile**

```
OS10# show policy-map interface
  Service-policy (qos) input:  p1
    Class (qos):  c1
      set qos-group 1

    Class (qos):  c2
      set qos-group 4

    Class (qos):  c3
      set qos-group 7
```

# View WRED Statistics

You can display egress-queue statistics of both transmitted and dropped packets and bytes. If the Packets counter is non-zero, the corresponding Drop Bytes counters display as "NA".

**View WRED Statistics**

```
OS10# show queuing statistics interface ethernet 1/1/1 wred
Interface ethernet1/1/1 (All queues)
Description      Packets        Bytes
Output          2811           418309
Dropped         0              0
Green Drop      0              0
Yellow Drop     0              0
Red Drop        0              0
```

**View WRED Statistics When Packets Counter is Non-Zero**

```
OS10# show queuing statistics interface ethernet 1/1/1 wred
Interface ethernet1/1/1 (All queues)
Description      Packets        Bytes
```

```
Output          0          0
Dropped         0          0
Green Drop      123        NA
```

# View QoS Configuration

You can view QoS configuration information related to active class-maps, policy-maps, and match criteria.

- Display the ingress or egress QoS configuration details in EXEC mode.

  `show qos {control-plane | system | interface interface}`

  - `control-plane` — View all QoS control-plane information.
  - `system` — View all QoS system information.
  - `interface ethernet node/slot/port:[subport]` — View the QoS configuration for the specified Ethernet interface node, slot, and port identification number.

- Display the configuration details of all existing class-maps in EXEC mode.

  `show class-map type {control-plane | qos | queuing}class-map-name`

  - `type` — Enter the policy-map type (`qos`, `queuing`, or `control-plane`).
  - `qos` — Displays all policy-maps in the `qos` type.
  - `queuing` — Displays all policy-maps in the `queuing` type.
  - `control-plane` — Displays all policy-maps in the `control-plane` type.
  - `class-map-name` — Enter the QoS class-map name to display the details of a specific policy-map.

- Display information on all existing policy-maps in EXEC mode.

  `show policy-map type {control-plane | qos | queuing}[policy-map-name]`

  - `type` — Enter the policy-map type (`qos`, `queuing`, or `control-plane`).
  - `control-plane` — View all policy-maps in the `control-plane` type.
  - `qos` — View all policy-maps in the `qos` type.
  - `queuing` — View all policy-maps in the `queuing` type.
  - `policy-map name` — Enter the QoS policy-map name to display the details of a specific policy-map.

### View QoS Control-Plane

```
OS10# show qos control plane
    Service-policy (Input): p1
```

### View QoS System

```
OS10# show qos system
    Service-policy (Input): p1
    Service-policy (Output): p2
```

### View QoS Interface Information

```
OS10# show qos interface ethernet 1/1/5
```

### View QoS Class-Map

```
OS10# show class-map type qos c1
    Class-map (qos):  c1 (match-all)
      Match(not): ip-any dscp 10
```

### View QoS Policy-Map

```
OS10# show policy-map interface
  Service-policy (qos) input:  p1
    Class (qos):  c1
      set qos-group 1

    Class (qos):  c2
```

```
        set qos-group 4

   Class (qos):  c3
        set qos-group 7
```

# View QoS Statistics

Display egress-queue statistics of both transmitted and dropped packets and bytes.

- View the number of packets and bytes on the egress-queue profile on a specific interface in EXEC mode.

  `show qos interface ethernet node/slot/port[:subport] queue`
- View the number of packets and bytes on the egress-queue profile on a specific queue in EXEC mode.

  `show queuing statistics interface ethernet node/slot/port[:subport] queue number`

### View Number of Packets and Bytes in Egress Queue Profile

```
OS10# show queuing statistics interface ethernet 1/1/1
Interface ethernet1/1/1 (All queues)
Description      Packets      Bytes
Output           2811         418309
Dropped          0            0
Green Drop       0            0
Yellow Drop      0            0
Red Drop         0            0
```

### View Number of Packets and Bytes on Specific Queue

```
OS10# show queuing statistics interface ethernet 1/1/1 queue 3
Interface ethernet 1/1/1 Queue 3
Description Packets  Bytes
Output          0      0
Dropped         0      0
```

# QoS Commands

## bandwidth

Assigns a percentage of weight to the queue.

| | |
|---|---|
| Syntax | `bandwidth percent value` |
| Parameters | `percent value` — Enter the bandwidth percentage value to the queue (1 to 100). |
| Default | Not configured |
| Command Mode | POLICY-MAP |
| Usage Information | If you configure this command, you cannot use the `priority` command for the class. |
| Example | `OS10(conf-pmap-que)# bandwidth percent 70` |
| Supported Releases | 10.2.0E or later |

# class

Creates a QoS class for a type of policy-map.

| | |
|---|---|
| Syntax | `class class-name` |
| Parameters | `class-name` — Enter a name for the class-map (up to 32 characters). |
| Default | Not configured |
| Command Mode | POLICY-MAP-QUEUEING |
| Usage Information | If you define a class-map under a policy-map, the type (`qos`, `queuing`, or `control-plane`) is the same as the policy-map. You must create this map in advance. The only exception to this rule is when the policy-map type is `trust`, where the class type must be `qos`. |
| Example | `OS10(conf-pmap-qos)# class c1` |
| Supported Releases | 10.2.0E or later |

# class-map

Creates a QoS class-map which filters traffic to match packets to the corresponding QoS policy created for your network.

| | |
|---|---|
| Syntax | `class-map [type {qos | queuing | control-plane}] [{match-any | match-all}] class-name` |
| Parameters | • `type` — Enter a class-map type.<br>• `control-plane` — Enter a `control-plane` type class-map.<br>• `qos` — Enter a `qos` type class-map.<br>• `queuing` — Enter a `queueing` type class-map.<br>• `match-all` — Define all the class-map filters. Packets must meet any of the match criteria to qualify as a member of the class.<br>• `match-any` — Define a class-map filter. Packets must meet all the match criteria to qualify as a member of the class.<br>• `class-name` — Enter a class name for the class map (up to 32 characters). |
| Defaults | • `qos` — class-map type<br>• `match-any` — class-map filter |
| Command Mode | CLASS-MAP-QOS |
| Usage Information | Apply `match-any` or `match-all` class-map filters to `control-plane`, `qos`, and `queuing` type class-maps. |
| Example | `OS10(config)# class-map type qos match-all c1`<br>`OS10(conf-cmap-qos)#` |
| Command History | 10.2.0E or later |

# clear qos statistics

Clears all QoS related statistics.

| | |
|---|---|
| Syntax | `clear qos statistics` |
| Parameters | None |
| Default | Not configured |
| Command Mode | EXEC |
| Usage Information | This command clears all QoS related statistics. |
| Example | `OS10# clear qos statistics` |
| Supported Releases | 10.2.0E or later |

# clear qos statistics type

Clears all queue counters for the control-plane, qos, and queueing.

| | |
|---|---|
| Syntax | `clear qos statistics type {{qos | queuing | control-plane} [interface ethernet node/slot/port[:subport]]}` |
| Parameters | <ul><li>`qos` — Clears `qos` type statistics.</li><li>`queuing` — Clears `queueing` type statistics.</li><li>`control-plane` — Clears `control-plane` type statistics.</li><li>`interface ethernet node-id/slot/port-id [:subport]` — Clears QoS statistics for an Ethernet interface configured for `qos`, `queuing`, or `control-plane`.</li></ul> |
| Default | Not configured |
| Command Mode | EXEC |
| Usage Information | This command clears all queue counters for QoS, queuing, control-plane, and interface information. |
| Example | `OS10# clear qos statistics type qos interface ethernet 1/1/5` |
| Example (Control-Plane) | `OS10# clear qos statistics type control-plane interface ethernet 1/1/7` |
| Example (Queuing) | `OS10# clear qos statistics type queuing interface ethernet 1/1/2` |
| Supported Releases | 10.2.0E or later |

# control-plane

Enters Control-Plane mode.

| | |
|---|---|
| Syntax | `control-plane` |
| Parameters | None |

| | |
|---|---|
| Default | Not configured |
| Command Mode | CONTROL-PLANE |
| Usage Information | If you attach an access-list to the `class-map` type of control-plane, the access-list ignores the `permit` and `deny` keywords. The `no` form of this command is not allowed. |
| Example (Class-Map) | `OS10(config)# class-map type control-plane match-any c1`<br>`OS10(conf-cmap-control-plane)#` |
| Example (Policy-Map) | `OS10(config)# policy-map type control-plane p1`<br>`OS10(conf-pmap-control-plane)#` |
| Supported Releases | 10.2.0E or later |

# match

Configures a match criteria for the QoS policy.

| | |
|---|---|
| Syntax | `match {cos cos-number | ip [access-group name name | dscp dscp-value | precedence value] | mac access-group acl-name | not [ip | cos] vlan vlan-id}`<br>`[set dscp dscp-value]` |
| Parameters | • `cos cos-number` — Enter a queue number for the CoS match criteria (0 to 7).<br>• `ip` — Enter the IPv4 match criteria.<br>• `access-group name name` — (Optional) Enter the IPv4 access-group name.<br>• `dscp dscp-value` — (Optional) Enter a DSCP value for Layer 3 DSCP match criteria (0 to 63).<br>• `precedence value` — (Optional) Enter a precedence value for Layer 3 precedence match criteria (0 to 7).<br>• `mac access-group name name` — Enter an access-group name for the MAC access-list match criteria (up to 140 characters).<br>• `set dscp dscp-value` — Enter a DSCP value for marking the DSCP packets (0 to 63).<br>• `not` — Enter the IP or CoS to negate the match criteria.<br>• `vlan vlan-id` — Enter a VLAN number for VLAN match criteria (1 to 4094). |
| Default | Not configured |
| Command Mode | CLASS-MAP |
| Usage Information | In a `match-any` class, you can enter multiple match criteria. In a `match-all` class, if the match case is `access-group`, no other match criteria is allowed. If you attach the access-list to `class-map type control-plane`, the access-list ignores the `permit` and `deny` keywords. |
| Example | `OS10(conf-cmap-qos)# match ip access-group name ag1` |
| Supported Releases | 10.2.0E or later |

# match cos

Matches a cost of service (CoS) value to L2 dot1p packets.

| | |
|---|---|
| Syntax | `match [not] cos cos-value` |

**Parameters**

- *cos-value* — Enter a CoS value (0 to 7).
- not — Enter not to cancel the match criteria.

**Default**      Not configured

**Command Modes**      CLASS-MAP

**Usage Information**      You cannot have two match statements with the same filter-type. If you enter two match statements with the same filter-type, the second statement overwrites the first statement.

**Example**
```
OS10(conf-cmap-qos)# match cos 3
```

**Supported Releases**      10.2.0E or later

# match dscp

Configures a DSCP value as a match criteria for a class-map.

**Syntax**
```
match [not] {ip | ipv6 | ip-any } dscp [dscp-list | dscp-list]
```

**Parameters**

- not — (Optional) Enter to cancel a previously applied match criteria.
- ip — Enter the IPv4 protocol as the match protocol.
- ipv6 — Enter the IPv6 protocol as the match protocol.
- ip-any — Enter both IPv4 and IPv6 as the match protocol.
- dscp *dscp-list* | *dscp-list* — Enter a DSCP value in single numbers, comma separated, or a hyphenated range (0 to 63).

**Default**      Not configured

**Command Mode**      CLASS-MAP

**Usage Information**      You cannot enter two match statements with the same filter-type. If you enter two match statements with the same filter-type, the second statement overwrites the first statement. The match-all option in a class-map does not support ip-any. Select either ip or IPv6 for the match-all criteria. If you select ip-any, you cannot select ip or ipv6 for the same filter type.

**Example**
```
OS10(conf-cmap-qos)# match ip-any dscp 17-20
```

**Supported Releases**      10.2.0E or later

# match precedence

Configures a match precedence rule based on the 3–bits Differentiate Serve value for a QoS policy.

**Syntax**
```
match [not] {ip | ipv6 | ip-any} precedence precedence-list
```

**Parameters**

- not — Cancel a previously applied match precedence rule.
- ip — Enter IPv4 as the match precedence rule.
- ipv6 — Enter IPv6 as the match precedence rule.
- ip-any — Enter both IPv4 and IPv6 as the match precedence rule.

- precedence *precendence-list* — Enter a precedence-list value (0 to 7).

| | |
|---|---|
| Default | Not configured |
| Command Mode | CLASS-MAP |
| Usage Information | You cannot enter two match statements with the same filter-type. If you enter two match statements with the same filter-type, the second statement overwrites the first statement. |
| Example | `OS10(conf-cmap-qos)# match not ipv6 precedence 3` |
| Supported Releases | 10.2.0E or later |

## match qos-group

Configures a match criteria for a QoS group.

| | |
|---|---|
| Syntax | `match qos-group` *qos—group-id* |
| Parameters | *qos-group-id* — Enter a QoS group identification number (0 to 11). |
| Default | Not configured |
| Command Mode | CLASS-MAP |
| Usage Information | You can only configure this command when the class-map type is `queuing`. You cannot enter two match statements with the same filter-type. If you enter two match statements with the same filter-type, the second statement overwrites the first statement. When class-map type is `queuing`, the qos-group corresponds to data queues (0 to 7). |
| Example | `OS10(conf-cmap-queuing)# match qos-group 1` |
| Supported Releases | 10.2.0E or later |

## match vlan

Configures a match criteria based on the VLAN identification number.

| | |
|---|---|
| Syntax | `match vlan` *vlan-id* |
| Parameters | *vlan-id* — Enter a VLAN ID number (1 to 4094). |
| Default | Not configured |
| Command Mode | CLASS-MAP |
| Usage Information | You cannot enter two match statements with the same filter-type. If you enter two match statements with the same filter-type, the second statement overwrites the first statement. |
| Example | `OS10(conf-cmap-qos)# match vlan 100` |
| Supported Releases | 10.2.0E or later |

# police

Configures traffic policing on incoming traffic.

| | |
|---|---|
| Syntax | `police {cir committed-rate [bc committed-burst-size]} {pir peak-rate [be peak-burst-size]}` |
| Parameters | • `cir committed-rate` — Enter a committed rate value in kilo bits per second (0 to 4000000).<br>• `bc committed-burst-size` — (Optional) Enter a committed burst size in packets for the control plane and kilobytes (16 to 200000).<br>• `pir peak-rate` — Enter a peak-rate value in kilo bits per second (0 to 40000000).<br>• `be peak-burst-size` — (Optional) Enter a peak burst size in kilo bytes (16 to 200000). |
| Defaults | • The `bc committed-burst-size` value is 200 KB for `control plane` and 100 KB for all other class-map types.<br>• The `be peak-burst-size` value is 200 KB for `control plane` and 100 KB for all other class-map types. |
| Command Mode | POLICY-MAP-CLASS-MAP |
| Usage Information | If you do not provide the peak-rate `pir` values, the committed-rate `cir` values are taken as the `pir` values. Only the ingress QoS policy type supports this command. For control-plane policing, the rate values are in pps. |
| Example | `OS10(conf-pmap-c-qos)# police cir 5 bc 30 pir 20 be 40` |
| Supported Releases | 10.2.0E or later |

# policy-map

Enters QoS Policy-Map mode and creates or modifies a QoS policy-map.

| | |
|---|---|
| Syntax | `policy-map policy-map-name [type {qos | queuing | control-plane}]` |
| Parameters | • `policy-map-name` — Enter a class name for the policy-map (up to 32 characters).<br>• `type` — Enter the type of policy-map:<br>  • `qos` — Create a `qos` type policy-map.<br>  • `queuing` — Create a `queueing` type policy-map.<br>  • `control-plane` — Create a `control-plane` type policy-map. |
| Defaults | `qos` = class-map type and `match-any` = class-map filter |
| Command Mode | CONFIGURATION |
| Usage Information | The `no` version of this command deletes a policy-map. |
| Example | `OS10(config)# policy-map p1` |
| Example (Queuing) | `OS10(config)# policy-map type queuing p1` |
| Supported Releases | 10.2.0E or later |

DELLEMC

# priority

Sets the scheduler as a strict-priorty.

| | |
|---|---|
| Syntax | `priority` |
| Parameters | None |
| Default | WRR — when priority is mentioned, it moves to SP with default level 1 |
| Command Mode | POLICY-MAP-CLASS-MAP |
| Usage Information | If you use this command, bandwidth is not allowed. Only the egress QoS policy type supports this command. |
| Example | `OS10(conf-pmap-que)# priority` |
| Supported Releases | 10.2.0E or later |

# random-detect

Configures WRED parameters for the queue.

| | |
|---|---|
| Syntax | `random-detect minimum-threshold threshold-value maximum-threshold threshold-value drop-probability value percentage [weight value] [color {green | yellow}] [ecn]` |
| Parameters | • `minimum threshold threshold value` — Enter the minimum drop rate (1 to 12480 kbps).<br>• `maximum threshold threshold value` — Enter the maximum drop rate (1 to 12480 kbps).<br>• `drop probability percentage` — Enter a drop probability rate in percentage (1 to 100).<br>• `weight value` — Enter the weight value (1 to 15, default 0).<br>• `color` — Enter a color drop precedence.<br>• `green` — Enter a medium drop precedence (default).<br>• `yellow` — Enter a low drop precedence.<br>• `ecn` — Enter the explicit congestion notification (ECN). |
| Default | ECN disabled |
| Command Mode | CONFIG-POLICY-MAP-CLASS-MAP |
| Usage Information | Use this command to configure WRED parameters for the queue. |
| Example | `OS10(conf-pmap-c-que)# random-detect minimum-threshold 10 maximum-threshold 100 drop-probability 50 color yellow ecn weight 10` |
| Supported Releases | 10.2.0E or later |

# service-policy

Configures the input and output service policies.

| | |
|---|---|
| Syntax | `service-policy {input | output} [type {qos | queuing}] policy-map-name` |

**Parameters**

- input — Assign a QoS policy to the interface input.
- output — Assign a QoS policy to the interface output.
- qos — Assign a qos type policy-map.
- queuing — Assign the queuing type policy-map.
- *policy-map-name* — Assign the policy map name (up to 32 characters).

**Default**          Not configured

**Command Mode**     INTERFACE

**Usage Information**  Attach only one policy map to the interface input and output for each qos and queuing policy-map type. Attach four service-policies to the system QoS — one each for qos and queueing type policy-maps. When you configure service policies at the interface level and system level, the interface-level policy takes precedence over the system-level policy.

**Example**          OS10(conf-if-eth1/1/7)# service-policy input type qos p1

**Supported Releases**  10.2.0E or later

# set cos

Sets a cost of service (CoS) value to mark L2 802.1x (dot1p) packets.

**Syntax**           set cos *cos-value*

**Parameters**       *cos-value* — Enter a CoS value (0 to 7).

**Default**          Not configured

**Command Mode**     POLICY-MAP-CLASS-MAP

**Usage Information**  You cannot enter two set statements with the same action-type. If you enter two set statements with the same action-type, the second statement overwrites the first. When class-map type is qos, the qos-group corresponds to data queues 0 to 7.

**Example**          OS10(conf-pmap-c-qos)# set cos 6

**Supported Releases**  10.2.0E or later

# set dscp

Sets the drop precedence for incoming packets based on their DSCP value and color map profile.

**Syntax**           set dscp *dscp-value* [color {red | yellow}]

**Parameters**
- *dscp value* — Enter a DSCP value (0 to 63).
- color — (Optional) — Enter to apply a color map profile.
- red — Enter to mark the packets to drop.
- yellow — Enter to mark the packets to deliver to the egress queue.

**Default**          Not configured

**Command Mode**     POLICY-MAP-CLASS-MAP

| | |
|---|---|
| Usage Information | The `QoS` ingress QoS policy type only supports this command. The packets marked as `color yellow` deliver to the egress queue. The egress queue transmits the packets with the available bandwidth. If bandwidth is not available, the packets drop. All packets marked as `color red` drop. When class-map type is `qos`, the qos-group corresponds to data queues 0 to 7. |
| Example | `OS10(conf-pmap-c-qos)# set dscp 10 color yellow` |
| Supported Releases | 10.2.0E or later |

## set qos-group

Configures marking for the QoS-group queues.

| | |
|---|---|
| Syntax | `set qos-group queue-number` |
| Parameters | `queue-number` — Enter a queue number (0 to 7). |
| Default | Not configured |
| Command Mode | POLICY-MAP-CLASS-MAP |
| Usage Information | The `qos` or `control-plane` ingress QoS policy type only supports this command. When class-map type is `control-plane`, the qos-group corresponds to CPU queues 0 to 11, and when the class-map type is `qos`, the qos-group corresponds to data queues 0 to 7. |
| Example | `OS10(conf-pmap-c-qos)# set qos-group 7` |
| Supported Releases | 10.2.0E or later |

## shape

Shapes the outgoing traffic rate.

| | |
|---|---|
| Syntax | `shape {min {kbps | mbps | pps}min-value [burst-size]}{max {kbps | mbps | pps max-value [max-burst-size]}` |
| Parameters | • `min` — Enter the minimum committed rate in unit (kbps, mbps, or pps).<br>• `kbps` — Enter the committed rate unit in kilobits per second (0 to 40000000).<br>• `mbps` — Enter the committed rate unit in megabits per second (0 to 40000).<br>• `pps` — Enter the committed rate unit in packets per second (1 to 268000000).<br>• `burst-size` — Enter the burst size in kilobytes per packet (0 to 10000 or 1 to 1073000).<br>• `max` — Enter the maximum peak rate in kbps, mbps, or pps.<br>• `max-burst-size` — Enter the burst size in kilobytes per packets (0 to 10000 or 1 to 1073000). |
| Default | 50 kb or 200 packets |
| Command Mode | POLICY-MAP-CLASS-MAP |
| Usage Information | Only the ingress QoS policy type supports this command. You must enter both the minimum and maximum values for this command. If you enter the rate value in pps, the burst provided is in packets. If you enter the rate in kbps or mbps, the burst is provided in kb. If you enter the minimum rate in pps, you must also enter the maximum rate in pps. |

| Example | `OS10(conf-pmap-c-que)# shape min pps 10 max pps 40` |

| Supported Releases | 10.2.0E or later |

# show class-map

Displays configuration details of all existing class-maps.

| Syntax | `show class-map [type {control-plane | qos | queuing} class-map-name]` |

| Parameters | |

- `type` — Enter the policy-map type (`qos`, `queuing`, or `control-plane`).
- `qos` — Displays all policy-maps of `qos` type.
- `queuing` — Displays all policy-maps of `queuing` type.
- `control-plane` — Displays all policy-maps of `control-plane` type.
- `class-map-name` — Displays the QoS class-map-name of a class-map.

| Default | Not configured |

| Command Mode | EXEC |

| Usage Information | This command displays all class-maps of `qos`, `queuing`, or `control-plane` type. The `class-map-name` parameter displays all details of a configured class-map name. |

| Example | ```
OS10# show class-map type qos c1
    Class-map (qos):  c1 (match-all)
      Match(not): ip-any dscp 10
``` |

| Supported Releases | 10.2.0E or later |

# show control-plane info

Displays the control-plane queue mapping and rate limit.

| Syntax | `show control-plane info` |

| Parameters | None |

| Default | Not configured |

| Command Mode | EXEC |

| Usage Information | Monitors the statistics for the control plane and troubleshoot CoPP. |

| Example | ```
OS10# show control-plane info
Queue  Rate Limit(in pps) Protocols
0      600
1      1000
2      300
3      1300
4      2000              VLT NDS
5      400               ARP_REQ IPV6_ICMP_REQ
6      400               ARP_RESP IPV6_ICMP IPV6_ICMP_RESP IPV4_ICMP SSH
TELNET TACACS NTP FTP
7      400               RSTP PVST MSTP LACP
8      600               DOT1X LLDP
9      600               IPV6_OSPF IPV4_BGP IPV4_OSPF
``` |

```
         10    600                    IPV6_DHCP IPV4_DHCP SERVICEABILITY
         11    300                    OPEN_FLOW
```

**Supported Releases**    10.2.0E or later


# show control-plane statistics

Displays counters of all the CPU queue statistics.

| | |
|---|---|
| Syntax | `show control-plane info` |
| Parameters | None |
| Default | Not configured |
| Command Mode | EXEC |
| Usage Information | This command displays control-plane statistics for all counters. |
| Example | |

```
OS10# show control-plane statistics
Queue Packets    Bytes       Dropped Packets Dropped Bytes
0     0          0           0               0
1     0          0           0               0
2     0          0           0               0
3     0          0           0               0
4     0          0           0               0
5     2          172         0               0
6     0          0           0               0
7     32048      2180484     0               0
8     14140      2569184     0               0
9     0          0           0               0
10    0          0           0               0
11    0          0           0               0
```

**Supported Releases**    10.2.0E or later


# show policy-map

Displays information on all existing policy-maps.

| | |
|---|---|
| Syntax | `show policy-map type {control-plane | qos | queuing}] [`*`policy-map-name`*`]` |
| Parameters | • `type` — Enter the policy-map type (`qos`, `queuing`, or `control-plane`).<br>• `qos` — Displays all policy-maps of `qos` type.<br>• `queuing` — Displays all policy-maps configured of `queuing` type.<br>• `control-plane` — Displays all policy-maps of `control-plane` type.<br>• *`policy-map-name`* — Displays the QoS policy-map name details. |
| Default | Not configured |
| Command Mode | EXEC |
| Usage Information | This command displays all existing policy-map information. The *`policy-map-name`* parameter displays details of the policy-map name. |
| Example | |

```
OS10# show policy-map
 Service-policy(qos) input: p1
   Class-map (qos): c1
```

```
            set qos-group 1
      Service-policy(qos) input: p2
        Class-map (qos): c2
            set qos-group 2
```

**Supported Releases**    10.2.0E or later

# show qos control-plane

Displays the QoS configuration applied to control plane.

**Syntax**              `show qos control-plane`

**Parameters**          None

**Default**             Not configured

**Command Mode**        EXEC

**Usage Information**   Monitors statistics for the control plane and troubleshoots CoPP.

**Example**
```
OS10# show qos control-plane
      Service-policy (Input): p1
```

**Supported Releases**    10.2.0E or later

# show qos interface

Displays the QoS configuration applied on a specific interface.

**Syntax**              `show qos interface ethernet `*`node/slot/port[:subport]`*

**Parameters**          *`node/slot/port[:subport]`* — Enter the Ethernet interface node/slot/port information.

**Default**             Not configured

**Command Mode**        EXEC

**Usage Information**   This command displays QoS interface information.

**Example**
```
OS10# show qos interface ethernet 1/1/10
Ethernet 1/1/10
    unknown-unicast-storm-control : 100 pps
    multicast-storm-control : 200 pps
    broadcast-storm-control : Disabled
    flow-control-rx: Enabled
    flow-control-tx: Disabled
      Service-policy (Input)(qos): p1
```

**Supported Releases**    10.2.0E or later

# show qos system

Displays the QoS configuration applied to the system.

**Syntax**              `show qos system`

**Parameters**          None

| Default | Not configured |
| --- | --- |
| Command Mode | EXEC |
| Usage Information | View and verify system level service policy configuration information. |
| Example | ```
OS10# show qos system
Service-policy (Input) (qos) : policy1
Service-policy (Output)(queuing) : policy2
``` |
| Supported Releases | 10.2.0E or later |

# show queuing statistics

Displays QoS queuing statistics information.

| Syntax | `show queuing statistics interface ethernet node/slot/port[:subport] [queue number]` |
| --- | --- |
| Parameters | • `node/slot/port[:subport]` — Enter the Ethernet interface node/slot/port information.<br><br>• `queue number` — Enter the QoS queue number (0 to 7). |
| Default | Not configured |
| Command Mode | EXEC |
| Usage Information | Use this command to view all queuing counters. WRED counters are available only at the port level. |
| Example | ```
OS10# show queuing statistics interface ethernet 1/1/1
Interface ethernet1/1/1 (All queues)
Description Packets  Bytes
Output          0        0
Dropped         0        0
Green Drop      0        0
Yellow Drop     0        0
Red drop        0        0
``` |
| Example (Queue) | ```
OS10# show queuing statistics interface ethernet 1/1/1 queue 3
Interface ethernet1/1/1 Queue 3
Description Packets  Bytes
Output          0        0
Dropped         0        0
``` |
| Supported Releases | 10.2.0E or later |

# system qos

Configures system-level service policies.

| Syntax | `system qos` |
| --- | --- |
| Parameters | None |
| Default | Not configured |
| Command Mode | SYSTEM-QOS |
| Usage Information | This command enters System QoS mode. |

| Example | ```
OS10(config)# system qos
OS10(config-sys-qos)#
``` |
|---|---|

| Supported Releases | 10.2.0E or later |
|---|---|

# trust

Sets the dynamic classification to trust.

| Syntax | `trust {dot1p | diffserv} [fallback]` |
|---|---|
| Parameters | • `diffserv` — Set the dynamic classification to `trust DSCP`.<br>• `dot1p` — Set the dynamic classification to `trust Dot1p`.<br>• `fallback` — (Optional) Honor trusting dot1p or DSCP only if other match criteria in this policy map does not qualifies for a packet. |

| Default | Disabled |
|---|---|
| Command Mode | POLICY-MAP-CLASS-MAP |
| Usage Information | The ingress QoS policy type and `class-trust` support this command. |
| Example | `OS10(conf-pmap-c-qos)# trust dot1p` |

| Supported Releases | 10.2.0E or later |
|---|---|