CONFIGURATION GUIDE

# Extreme SLX-OS QoS and Traffic Management Configuration Guide, 18x.1.00

## Supporting the ExtremeSwitching SLX 9030 Switches

*9035682*
*November 2018*

# Contents

# Preface



This section discusses the conventions used in this guide, ways to provide feedback, additional help, and other Extreme Networks® publications.

# Conventions

This section discusses the conventions used in this guide.

## Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
**A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.**

DANGER
***A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.***

## Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

| Format | Description |
|---|---|
| **bold** text | Identifies command names. |
| | Identifies keywords and operands. |
| | Identifies the names of GUI elements. |
| | Identifies text to enter in the GUI. |
| *italic* text | Identifies emphasis. |
| | Identifies variables. |
| | Identifies document titles. |

| Format | Description |
|---|---|
| `Courier font` | Identifies CLI output.<br>Identifies command syntax examples. |

## Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

| Convention | Description |
|---|---|
| **bold** text | Identifies command names, keywords, and command options. |
| *italic* text | Identifies a variable. |
| [ ] | Syntax components displayed within square brackets are optional.<br>Default responses to system prompts are enclosed in square brackets. |
| { **x** \| **y** \| **z** } | A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. |
| **x** \| **y** | A vertical bar separates mutually exclusive elements. |
| < > | Nonprinting characters, for example, passwords, are enclosed in angle brackets. |
| ... | Repeat the previous element, for example, *member*[*member*...]. |
| \ | Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash. |

# Documentation and Training

To find Extreme Networks product guides, visit our documentation pages at:

| | |
|---|---|
| Current Product Documentation | www.extremenetworks.com/documentation/ |
| Archived Documentation (for earlier versions and legacy products) | www.extremenetworks.com/support/documentation-archives/ |
| Release Notes | www.extremenetworks.com/support/release-notes |
| Hardware/Software Compatibility Matrices | https://www.extremenetworks.com/support/compatibility-matrices/ |
| White papers, data sheets, case studies, and other product resources | https://www.extremenetworks.com/resources/ |

## Open Source Declarations

Some software files have been licensed under certain open source licenses. More information is available at: www.extremenetworks.com/support/policies/open-source-declaration/.

## Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For more information, visit www.extremenetworks.com/education/.

# Getting Help

If you require assistance, contact Extreme Networks using one of the following methods:

| | |
|---|---|
| Extreme Portal | Search the GTAC (Global Technical Assistance Center) knowledge base, manage support cases and service contracts, download software, and obtain product licensing, training, and certifications. |
| The Hub | A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC. |
| Call GTAC | For immediate support: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact |

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

## Subscribing to Service Notifications

You can subscribe to email notifications for product and software release announcements, Vulnerability Notices, and Service Notifications.

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form with your information (all fields are required).
3. Select the products for which you would like to receive notifications.

   NOTE
   You can modify your product selections or unsubscribe at any time.

4. Click **Submit**.

# Providing Feedback to Us

Quality is our first concern at Extreme Networks, and we have made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you! We welcome all feedback but especially want to know about:

- Content errors or confusing or conflicting information.
- Ideas for improvements to our documentation so you can find the information you need faster.
- Broken links or usability issues.

If you would like to provide feedback to the Extreme Networks Information Development team, you can do so in two ways:

- Use our short online feedback form at https://www.extremenetworks.com/documentation-feedback/.

- Email us at documentation@extremenetworks.com.

Please provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

# About This Document



## Supported hardware and software

The following platforms are supported by this release:

- ExtremeSwitching SLX 9030 Series

## What's new in this document

The SLX-OS 18x.1.00 release is the first SLX-OS release that supports SLX 9030 devices.

## Regarding Ethernet interfaces and chassis devices

Many features can apply to either single-slot (1 RU) or multi-slot (chassis) devices.

The Ethernet interface configuration and output examples in this document may appear as either 0/*X* or *N*/*X* assignments, where *N* is an integer greater than 0.

Be aware of the interface configuration options of your particular device.

In addition, some legacy show outputs may reflect output from a variety of devices, including chassis devices.

# Traffic Policing



# Rate limiting and traffic policing overview

The purpose of rate limiting is to control the amount of bandwidth consumed by an individual flow or an aggregate of flows.

Rate limiting is applicable to both inbound and outbound traffic, where packets are dropped that are above the configured committed rates. Rate shaping controls traffic bursts applicable to egress traffic by buffering and queueing of excess packets above the committed rate.

Traffic policing refers to class-based rate limiters applicable to ingress and egress traffic.

Rate limiting on the SLX platform implements the single-rate, three-color mechanism based on RFC 2697 and the two-rate, three-color mechanism based on RFC 4115.

# Service policies — policy maps, class maps, and policers

Traffic policing is accomplished by applying a service policy to an interface.

A service policy consists of a policy map that specifies traffic policing and QoS parameters based on matching associated class maps.

One service policy can be applied per interface per direction.

## Policy maps

A policy map includes a set of class maps and policer values for the classified traffic. The policy map configuration allows you to specify policers in a single location that can be applied to multiple ports and to make changes to that policy.

The traffic policing policies must be defined using a policy map.

You can configure up to 1024 policy maps per system, but only one policy map can be specified per service policy.

Refer to Policy maps on page 12 for more information.

## Class maps

A class map is used to determine the traffic properties subject to policing. The port-based rate limit, applied using the default class map, is applicable to all traffic and can be used for ingress and egress service policies. An IP standard or extended ACL are also supported to classify traffic for ingress-only service policies.

The default and match access-group class maps can be combined in a single policy map.

Refer to Class maps on page 14 for more information.

## Service policy configuration rules

You must follow these binding rules when configuring or deleting a service policy:

- All policy map and class map names used in a service policy must be unique among all maps of that type.
- You can bind a service policy to multiple interfaces.
- A service policy can only be bound to physical ports and port-channel interfaces (LAGs). It cannot be bound to virtual interfaces.
- A service policy cannot be bound to an interface if a class map is not associated with the policy map referenced in the service policy.
- If a service policy is bound to an interface, and the policy class map lacks mandatory policer attributes (such as the CIR settings), the traffic on that interface is treated as conformed traffic. The packets on that interface are marked as green, no meter is allocated, and no statistics are available.
- BUM storm control and input service-policy features can coexist with each other. You can enable both at the same time on a given interface. BUM storm control has the highest precedence.

# Policy maps

Policy maps allow you to set a policy in a single location that affects multiple ports and to make changes to that policy.

The policy map configuration includes a set of class maps and QoS parameters.

A policy map allows you to specify policers in a single location that can be applied to multiple ports and to make changes to that policy.

The traffic policing policies must be defined using a policy map. One policy map can be specified per service policy. You can configure up to 1024 policy maps per system.

## Policy map configuration rules

Follow these rules when configuring traffic policing:

- A policer map (policy map or class map) name must be unique among all maps of that type.
- A policer name must begin with an alphabetic character ( a-z or A-Z) . An underscore, hyphen, and numeric values 0-9 can be used in the body of the name but not as the first character.
- You can configure a maximum of 1024 policy maps.
- ACL-based class maps and default class maps can be used in a single policy map.
- ACL-based, storm-control, and port-based rate limiting can coexist on the same interface. ACL-based rate limiting has the highest priority, followed by storm-control rate limiting, and then port-based rate limiting.
- For an ingress or egress service policy, one default class map can be specified per policy map.
- Broadcast, unknown unicast, and multicast (BUM) policies are counted separately.

- You cannot delete a policy map if is referenced in an active service policy (applied on an interface).

## QoS shaping rate

You can specify the shaping rate per port attached to the policy map to smooth the traffic that egresses an interface. This configuration is allowed only for egress traffic.

## Committed information rate and committed burst size

The committed information rate (CIR) bucket is defined by two separate parameters: the CIR rate, and the committed burst size (CBS) rate.

The CIR is the maximum number of bits a port is allowed to receive or send during a one-second interval. The rate of the traffic that matches the traffic policing policy cannot exceed the CIR. The CIR represents a portion of an interface's line rate (bandwidth), expressed in bits per second (bps), and it cannot be larger than the port's line rate. CIR-defined traffic that does not use the CIR available to it accumulates credits until the credit reaches to the CBS. These credits can be used later in circumstances where it temporarily exceeds the CIR.

When traffic exceeds the bandwidth that has been reserved for it by the CIR defined in its policy, it becomes subject to the CBS rate. The CBS rate provides a rate higher than the CIR to traffic that exceeded the CIR. The bandwidth in the CBS rate, as expressed in bytes, is accumulated during periods when policy-defined traffic does not use the full CIR available to it. Traffic is allowed to pass through the port for a short period of time at the CBS rate.

## Excess information rate and excess burst size

When inbound or outbound traffic exceeds the bandwidth available for the defined CIR and CBS, it is either dropped or made subject to the conditions set in the excess information rate (EIR) and excess burst size (EBS).

The EIR bucket provides an option for traffic that has exceeded the conditions set by policy for the CIR bucket. The EIR and EBS operate like the CIR and CBS except that they only act upon traffic that has been passed to the EIR bucket because it could not be accommodated by the CIR bucket. Like the CIR, the EIR provides an initial bandwidth allocation to accommodate inbound and outbound traffic. Like the CBS, the bandwidth available for burst traffic from the EBS is subject to the amount of bandwidth that is accumulated during periods of time when traffic that has been allocated by the EIR policy is not used. When inbound or outbound traffic exceeds the bandwidth available (the accumulated credits or tokens), it is dropped.

## Traffic policing behavior

Consider these behaviors when configuring traffic policing:

- Policer actions are applicable only to data traffic.
- When a Layer 2 control protocol is not enabled on an interface, those packets are dropped at ingress and are subject to ingress policing.
- If the configured CBS value is less than 2*(default MTU) value, then 2*(default MTU) is programmed as the CBS in the hardware. For example, if you configure CBS at 2000 bytes and the default MTU on an interface is 1548 bytes, when a policy map is applied on this interface, the CBS programmed in the hardware is 2*MTU (3096 bytes). If you update the MTU value, the CBS value is not be updated.
- If the CBS and EBS values are not configured, then these values are derived from the CIR and EIR respectively. The burst size calculation is as follows: Burst size (CBS or EBS) = (1.2 × information rate (CIR or EIR)) ÷ 8

- You have the responsibility to configure rate limit threshold values on an interface based on interface speed. No validation is performed for user-configured values against the interface speed.
- Because CIR is a mandatory policer attribute, you cannot delete the CIR parameter. If you want to delete the CIR attribute, use the **no police** command in policy-map-class sub-mode, which deletes all policer attributes.
- The policer operates in color blind mode; the color is evaluated at ingress and egress policers independently. This may result in packets that are marked as yellow in ingress policer to be evaluated as green at egress policer depending on policer settings.
- Packet drops caused by any action other than the ACL are included in the policer counter.
- Layer 3 control packets get policed at the egress side.

# Class maps

A class map is used to determine the traffic properties subject to policing.

An ACL can be used for match criteria while port-based policing is only implemented to match any criteria. Class maps can be used in a policy map to apply policing and QoS policies to a particular class. You can also define a default class map that matches any criteria.

The default class map, is applicable to all traffic and can be used for ingress and egress service policies. In addition, an IP standard or extended ACL are supported to classify traffic for ingress-only service policies.

## Class map configuration rules

Follow these rules when configuring class maps:

- A class map name must be unique among all maps of this type.
- A class map name must begin with an alphabetic character from a to z or from A to Z.
- Underscores, hyphens, and numeric characters 0 through 9 can be used in the body of the name but not as the first character.
- A class map cannot be deleted if it is referenced in a policy map.
- A class map cannot be deleted from a policy map when the policy map is bound to an active service policy.
- You can configure a maximum of 8K class maps.
- The default and match access-group class maps can be combined in a single policy map.
- Only one default class map can be specified per policy map.

## Class map policer configuration parameters

Use the values in the following table when setting the CIR, CBS, EIR, and EBS parameters.

**TABLE 1** Map parameters for rate limiting

| Parameter | Values | Range | Increments of |
|---|---|---|---|
| cir - Committed information rate | bits per second | 22000 through 300000000000 | Starts at 22000 then is rounded up to next achievable rate. |
| cbs - Committed burst size | Bytes per second | 1250 through 37500000000 | 1 Byte |
| eir - Excess information rate | bits per second | 22000 through 300000000000 | Starts at 22000 then is rounded up to next achievable rate. |
| ebs - Excess burst size | Bytes per second | 1250 through 37500000000 | 1 Byte |

NOTE
The parameters cir and eir are configured in bits per second, cbs and ebs are configured in Bytes per second.

The possible combinations when entering policer values are:

```
device(config-policymap-class)# police cir 600000000
device(config-policymap-class)# police cir 700000000 cbs 8000000000
device(config-policymap-class)# police cir 7000000000 cbs 70000000 eir 500000000
device(config-policymap-class)# police cir 7000000000 cbs 70000000 eir 500000000 ebs 90000000
device(config-policymap-class)# police cir 700000 eir 800000
device(config-policymap-class)# police cir 700000 eir 800000 ebs 6000000
```

Follow these rules when configuring the class map policier parameters:

- The CIR value must be specified; all other parameters are optional.
- You should configure the rate (CIR/EIR) and burst size (CBS/EBS) based on the interface speed.
- Default values will be calculated if not specified by the user.
- Configured values take priority over default values.
- If you only specify the CIR value, a default value is calculated and set to CBS value.
- If you specify the values of both CIR and CBS, the configured value takes priority over the default values in the policer.
- Should the CIR value be updated, the configured CBS value is retained, and the default value is not restored.
- If you want to revert to the default CBS value, you must first remove the configured CBS value.
- To disable the learning of MAC addresses for stream matching of the ACL-based rate limiting entries, the CIR and EIR values must be 0.

  The MAC address entries in the MAC-address table which are already learnt are not be flushed when you configure the CIR or EIR value as 0. You must explicitly clear the entries in MAC-address table by using the **clear mac-address-table dynamic** command.

# Traffic policer configuration rules for class maps

The following rules apply to configuring traffic policing for classified traffic in a policy map:

- A service policy map or class map name must be unique among all maps of that type.
- You cannot delete a service policy map or class map if it is active on an interface.
- Operational values that are programmed in the hardware are displayed as part of **show policy-map interface ethernet** *slot/port* command.
- A policer name must begin with an alphabetic character from a to z or from A to Z. Underscores, hyphens, and numeric characters 0 through 9 are permitted, except as the first character of the name.
- The configurable CIR and EIR ranges start from 22,000 bits per second (bps) and are rounded up to the next achievable rate.
- Percentage values are not supported as a policer parameter.
- Policer actions are not supported.
- If a service policy map is applied to an interface and no policer attributes are present in that service policy map, then ingress and egress packets on that interface are marked as green (conforming).
- If the configured CBS value is less than 2*(default MTU) value, then 2*(default MTU) is programmed as the CBS in the hardware. For example, if you configure CBS at 2000 bytes and the default MTU on an interface is 1548 bytes, when a policy map is applied on this interface, the CBS programmed in the hardware is 2*MTU (3096 bytes). If you update the MTU value, the CBS value is not be updated.

- If CBS and EBS values are not configured, then these values are derived from CIR and EIR values, respectively. Burst size calculation is as follows: Burst size (CBS or EBS) = (1.2 × information rate (CIR or EIR)) ÷ 8.
- If you do not configure EIR and EBS, then the single rate, two-color scheme is applied. Packets are marked as either green or red.
- You must configure rate limit threshold values on an interface based on interface speed.
- No validation is performed for user-configured values against interface speed.
- You can configure up to 1024 service policy maps. Broadcast, unknown unicast, and unknown multicast policies are counted separately.

## Default class map traffic policing

The default class map (port-based) policer feature controls the amount of bandwidth consumed by an individual flow or aggregate of flows.

The default class map is a port-based policer feature that controls the inbound (ingress) and outbound (egress) traffic rate on an individual port according to criteria that you define.

### *Default class map traffic policing considerations and limitations*

- You can configure up to 1024 policy maps.
- You can configure one default class map per policy map.
- You can configure up to 8k policers (ingress - egress).
- Traffic filtered by an ACL is not subject to default service policy policer.
- Match default class map service policy is supported on ingress and egress interfaces.
- BUM storm control and input service-policy features can coexist with each other. You can enable both at the same time on a given interface. BUM storm control has the highest precedence.
- ACL-based, storm-control, and port-based rate limiting can coexist on the same interface. ACL-based rate limiting has the highest priority, followed by storm-control rate limiting, and then port-based rate limiting.
- Control protocols are not rate-limited by the default class map service policy.
- The default class service policy does support remarking or internal queue assignment.
- Metering is performed on the packet as received on the wire. For example, including IPG and preamble, excluding CRC.
- Only hit counters are supported. FWD and DROP counters are not supported.

# Single rate three color marker

Single rate three color marker (SrTCM) meters an IP packet stream and marks its packets either green, yellow, or red.

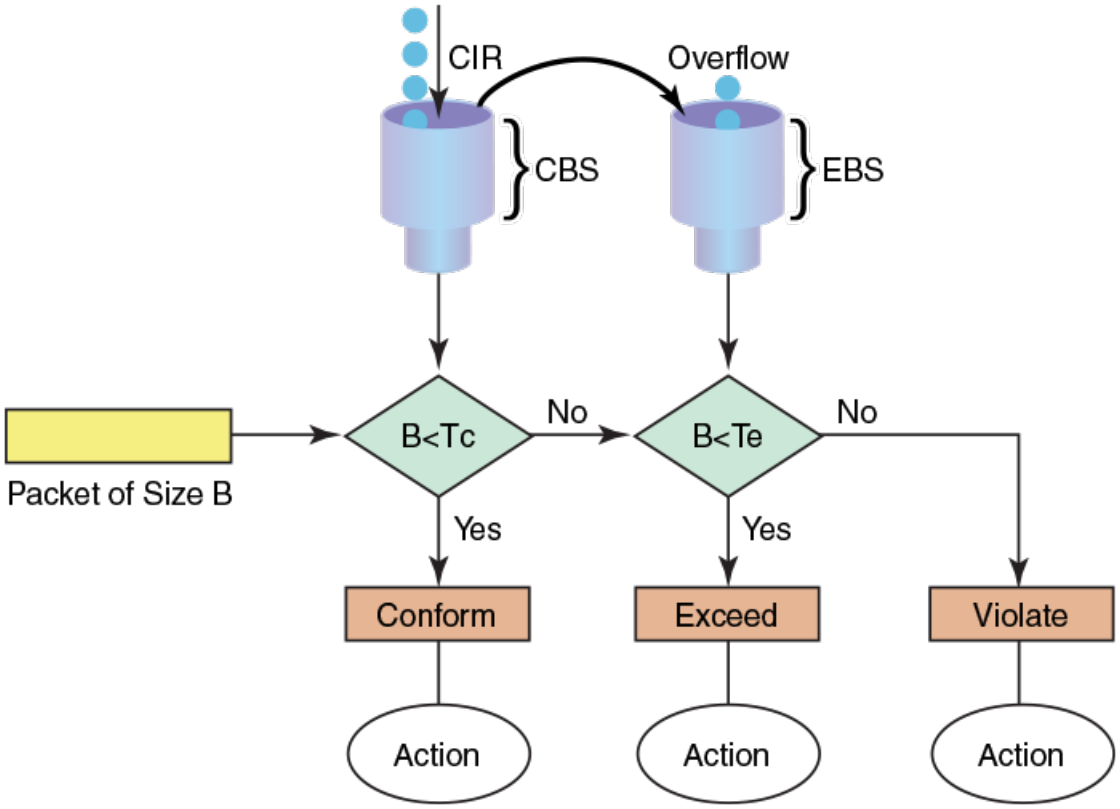
Single-rate traffic contract has there are three parameters, the, CIR, CBS and EBS. associated with this type of contract:

Marking is based on CIR and two associated burst sizes, CBS and EBS. Packets are marked:

- Green - if it does not exceed the CBS
- Yellow - if it does exceed the CBS, but not the EBS
- Red - otherwise

The SrTCM is useful for ingress policing of a service, where only the length, not the peak rate, of the burst determines service eligibility.

**FIGURE 1** Single rate three color marker




## Implementation

SrTCM traffic policing is implemented by tracking the current burst size using token-buckets, and discarding packets that exceed the CIR. In SrTCM the three color scheme has any incoming burst classified as either conforming (green, under CBS), exceeding (yellow, over CBS but under EBS), or violating (red, over EBS).

Every arriving packet is first compared to CBS and then to the EBS to determine the next action. There is a single flow of the tokens that fills the CBS bucket first and then continues to filling the EBS bucket. The second bucket is only filled if there was enough idle time to let the first bucket fill up completely.

The drawback of single-rate traffic contracts is that the service provider should be cautions assigning CIR bandwidth, by offering less bandwidth than it can service at any moment. The reason for this is that not all customers send traffic simultaneously, so network links may effectively become underutilized even at weak spots.

# Two-rate, three-color marker

The two-rate, three-color marker (TrTCM) meters an IP packet stream and marks its packets either green, yellow, or red.

There are four main parameters in a dual-rate traffic contract. CIR, CBS, PIR, and EBS.

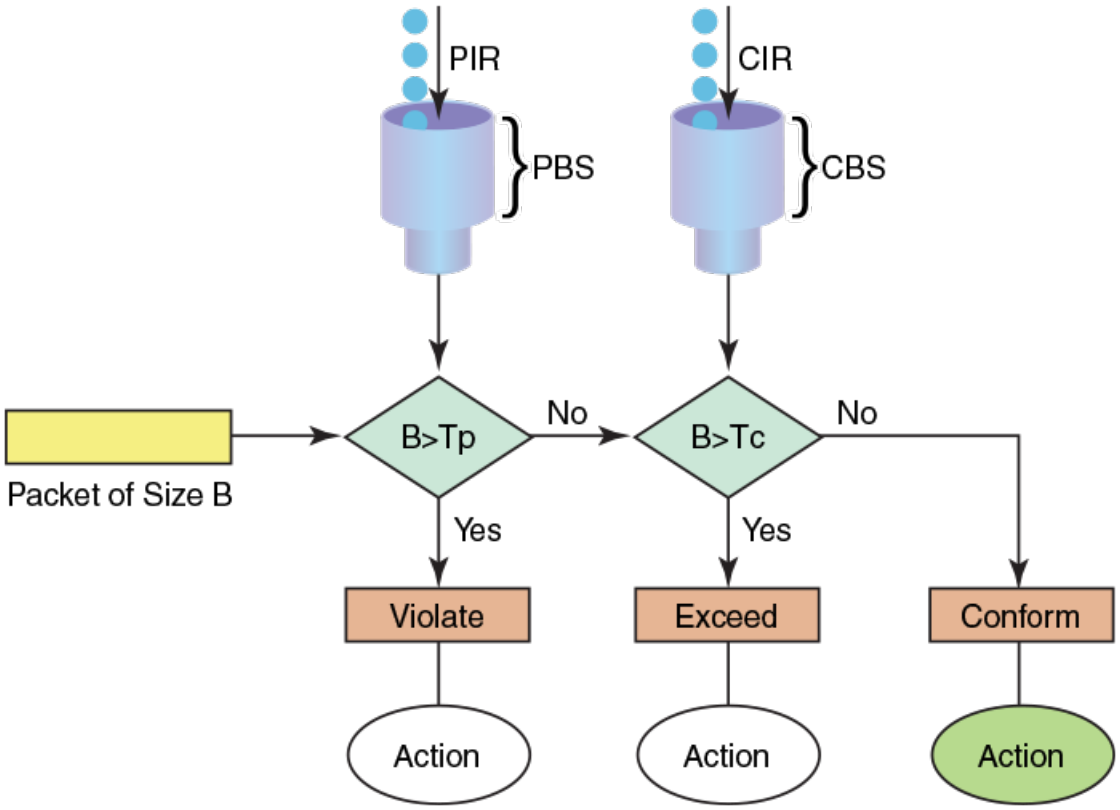
Marking is based on CIR. Packets are marked:

- Green - if it does not exceed the CIR

- Yellow - if it exceeds the CIR
- Red - if it exceeds the peak information rate (PIR)

The TrTCM is useful for ingress policing of a service, where a peak rate needs to be enforced separately from a committed rate.

**FIGURE 2** Two-rate, three-color marker




## Implementation

A dual-rate traffic contract supplies customers with two sending rates but only guarantees the smaller one. In case of congestion in the network, it discards traffic that exceeds the committed rate more aggressively and signals the customer to slow down to the committed rate. This principle was first widely implemented in Frame-Relay networks, but could be easily replicated using any packet-switching technology. There are four main parameters in a dual-rate traffic contract.

# Match access-group — class map policing

Access groups are used for Layer 2 and Layer 3 ACL-based ingress rate limit and denial of service (DoS) mitigation.

ACL-based rate limiting is built on top of ACL and policer features. It rate limits the following traffic:

- Layer 2 traffic that matches the permit conditions in Layer 2 access lists.
- Layer 3 traffic that matches the permit conditions specified in an IPv4 access list.

The ACL-based policer feature controls the amount of bandwidth consumed by an individual flow or aggregate of inbound flows by limiting the traffic rate on an individual port according to criteria defined by the **match access-group** class map. This ACL-based rate limiting feature can serve as a hardware solution to prevent DoS attacks.

# Match access-group - class map policing rules and limitations

Consider these rules and limitations when you are configuring **match access-group** class map policing:

- You can configure:
  - 1,024 policy maps
  - 6,144 ACL Content Addressable Memory (CAM) entries for use with rate limiting
  - 2,048 ACLs with rate limiting for each user

    NOTE
    The number of Ternary Content Addressable Memory (TCAM) entries for use with rate limiting and ingress policers are dependent on the hardware TCAM profile that is used.

- For protection against:
  - PING attacks
  - TCP Reset attacks
  - TCP SYN attacks
  - UDP attack
- Layer 2 MAC and Layer 3 IPv4 ACL-based rate limiting are supported.
- ACL-based rate limiting is applicable only to ingress traffic.
- There is one policer per ACL, it applies to all the rules for that ACL
- Control protocols are rate-limited if they match the configured ACL clause.
- When a **match access-group** class map rate limit is applied to a LAG logical port, and all LAG ports belong to the same tower, then MAX CIR value is the interface speed × number of physical ports. For example: if 0/1, 0/2 are LAG member ports, then MAX CIR will be 2 × 10Gbps.
- When a **match access-group** class map rate limit is applied to LAG logical port, MAX rate on that port is the number of the tower in that LAG × CIR. For example: if 0/1, 0/2 , 0/4, 0/5 are LAG member ports, then MAX rate is 3 × CIR.

# Precedence for the ACL and rate limiting features

When working with ACL features, their precedence are as follows:

OpenFlow > rACL > PBR > ACL > VLAN RL

When working with RL features, their precedence are as follows:

ACL RL > BUM RL > VLAN RL (port>system) > Port RL

All ACL and ACL RL features reside in one of two TCAM databases as shown in the following table.

**TABLE 2** TCAM databases and features

| Database | Feature |
|---|---|
| TCAM User | Layer 3 ACL |
| | Layer 2 ACL |
| | Layer 3 ACL rate limiting |

**TABLE 2** TCAM databases and features (continued)

| Database | Feature |
|---|---|
| | Layer 2 ACL rate limiting |
| TCAM Control (Ctrl) | Layer 3 Ctrl |
| | Layer 2 Ctrl |
| | VLAN rate limiting |
| | Port rate limiting |

For intra database features, priority is based on the entry strength or ordering, for example, first come first served basis. For inter database features, when there is a hit in both databases, the device first look at following actions:

- For non-conflicting actions, the actions are merged.
- For actions with the same strengths, the action from the User database takes precedence.

# Considerations and limitations for Layer 2 ACL-based rate limiting

- ACL-based rate limiting is applicable for ingress traffic only. Egress traffic is blocked by the SLX-OS device.
- Layer 2 ACL rate limiting takes precedence over BUM rate limiting.
- You can create a policy map with the class-map types of default, VLAN or BD, Layer 2 ACL, and Layer 3 ACL rate limiting.
- There is one policer for each ACL, and it applies to all rules for that ACL.
- Control protocols are rate limited if they match the configured ACL clause.
- ACL-based, storm-control, and port-based rate limiting can coexist on the same interface. ACL-based rate limiting has the highest priority, followed by storm-control rate limiting, and then port-based rate limiting.
- Packets drops caused by any action other than ACL rate limiting are included in the policer counters.
- Traffic that matches deny or hard-drop rules is not subject to rate limiting.
- Metering is performed on the packet size as received on the wire (including IPG, Preamble and SOF, ignoring FCS/CRC).
- A policy map can be applied to a physical port as well as LAG interfaces.
- Multiple class maps with user-defined ACLs can be added in a policy map. However, once a matching ACL clause is found, the device does not evaluate subsequent ACL clauses and subsequently rate limits those ACLs.
- The configured rate in bits per second (bps) is rounded up to the next achievable rate.
- Statistics display only two colors. Conform includes Green and Yellow color packets. Violate includes dropped or a RED color packet.
- You cannot delete a policy map if it is active on the interface.
- You cannot delete a class map from a policy map when the policy map is active on the interface.
- If a policy map is applied to an interface and no policer attributes are present in this policy map, then ingress and egress packets on this interface are marked as green (conforming).
- A policy map can be bound to multiple interfaces.
- A policy map cannot be bound to interface if a class map is not associated with this policy map.
- All LAG members within the same tower are governed by the same policer. One policer is for each tower.
- When ACL rate limiting is applied to a LAG logical port, and all LAG ports belong to the same tower, then the maximum CIR value is the interface speed * number of physical ports. For example, if Ethernet 1/1 and 1/2 are LAG member ports, then the maximum CIR is 1 * 10Gbps.
- Layer 2 ACL rate limiting can be configured at the interface or system level.

Interface ACL rate limiting has a higher priority than global ACL rate limiting.

The TCAM is shared between the interface and global ACL rate limiting. When a policy map is bound for interface and global ACL rate limiting, the TCAM can accommodate the following maximum number of class maps:

- L3 ACL RL = 2K class maps
- L3 ACL RL Global = 512 class maps
- L2 ACL RL = 512 class maps
- L2 ACL RL Global = 256 class maps

There is a software restriction of 32 class maps per policy map. The TCAM maximum number of class maps is inside a policy map whose ACL rules would be programmed to the TCAM. This number is derived based on the priority ranges as described in Precedence for the ACL and rate limiting features.

- The number of ACL entries are limited by the TCAM size. Refer to scalability numbers in the release notes.
- To disable the learning of MAC addresses for stream matching of the ACL-based rate limiting entries, the CIR and EIR values must be 0.

  The MAC address entries in the MAC-address table which are already learned will not be flushed when you configure the CIR or EIR value as 0. You must explicitly clear the entries in MAC-address table by using the **clear mac-address-table dynamic** command.

# Configuring Layer 2 ACL rate limiting

Perform the following steps to configure rate limiting based on a Layer 2 ACL.

1. In privileged EXEC mode, access global configuration mode.

   ```
   device# configure terminal
   ```

2. Create the Layer 2 ACL.

   ```
   device(config)# mac access-list extended m1
   ```

3. Add the permit rule for the ACL.

   ```
   device(conf-macl-ext)# permit any any vlan 100
   ```

4. Access global configuration mode.

   ```
   device(conf-macl-ext)# exit
   ```

5. Create the class map and access its configuration mode.

   ```
   device(config)# class-map c1
   ```

6. Add a match statement to the class map.

   ```
   device(config-classmap)# match access-group m1
   ```

7. Access global configuration mode.

   ```
   device(config-classmap)# exit
   ```

8. Create the policy map and access its configuration mode.

   ```
   device(config)# policy-map p1
   ```

9. Add the class map to the policy map and access the policy-class configuration mode.

```
device(config-policymap)# class c1
```

10. Configure the class map.

```
device(config-policymap-class)# police cir 400000
```

This step configures the committed information rate for the class map.

11. Access global configuration mode.

```
device(config-policymap-class)# exit
device(config-policymap)# exit
```

12. Access the interface configuration mode for the interface that you want to apply the policy map.

```
device(config)# interface ethernet 0/1
```

13. Apply the policy map to the interface.

```
device(conf-if-eth-0/1)# service-policy in p1
```

The following example shows the configuration in the previous steps.

```
device# configure terminal
device(config)# mac access-list extended m1
device(conf-macl-ext)# permit any any vlan 100
device(conf-macl-ext)# exit
device(config)# class-map c1
device(config-classmap)# match access-group m1
device(config-classmap)# exit
device(config)# policy-map p1
device(config-policymap)# class c1
device(config-policymap-class)# police cir 400000
device(config-policymap-class)# exit
device(config-policymap)# exit
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# service-policy in p1
```

### *Displaying Layer 2 ACL and policy map information*

To display the Layer 2 ACL bindings, use the **show access-list mac** command.

To display the policy-map bindings and rate-limiting statistics, use the **show policy-map interface** command.

## ACL-based rate limiting use cases

The following use cases describe four common DoS attacks and how to protect against them using ACL-based rate limiting.

Use case 1: Protection against TCP SYN attacks on page 23 and Configuring use case 1: Protection against TCP SYN attacks on page 33.
Use case 2: Protection against TCP RST attacks on page 23 and Configuring use case 2: Protection against TCP RST attacks on page 35.
Use case 3: Protection against ping flood attacks on page 23 and Configuring use case 3: Protection against ping flood attacks on page 38.
Use case 4: Protection against UDP flood attacks on page 23 and Configuring use case 4: Protection against UDP flood attacks on page 40.

## Use case 1: Protection against TCP SYN attacks

A TCP SYN attack, also known as a SYN flood, is a form of denial-of-service (DoS) attack where an attacker sends a series of SYN requests to a system in an attempt to consume enough server resources so that the system is unresponsive to other traffic.

TCP SYN attacks disrupt normal traffic by exploiting the way TCP connections are established. These attacks attempt to exhaust the target system's half open TCP queue, which is a limited resource to service new connection requests. The attacker creates a random source address for each packet and a SYN flag is set in each packet to request to open a new connection, The TCP IP stack of the victim responds to the spoofed IP with SYN ACK and waits for a return ACK from the sender which never comes.

Refer to Configuring use case 1: Protection against TCP SYN attacks on page 33.

## Use case 2: Protection against TCP RST attacks

A TCP RST (reset) attack is meant to abnormally terminate legitimate TCP connections by sending a random packet with the RST bit set.

In the packet stream of a TCP connection, each packet contains a TCP header and every header contains an RST bit. If this bit is set to 1, it instructs the receiving computer to immediately terminate the TCP connection. Following this instruction, the sending computer does not forward any more packets through the connection's ports, and discards any further packets it receives with headers indicating they should be sent to that connection.

A TCP reset terminates a TCP connection instantly.

Refer to Configuring use case 2: Protection against TCP RST attacks on page 35.

## Use case 3: Protection against ping flood attacks

A ping flood is a DoS attack that is based on sending the targeted system an overwhelming number of ICMP Echo Request (ping) packets.

The attack uses the ping flood option, which sends ICMP packets as fast as possible without waiting for replies. In a successful attack, the target system responds to the ping requests with ICMP Echo Reply packets, consuming both outgoing bandwidth as well as incoming bandwidth. If the target system is slow enough, it is possible to consume enough of its CPU cycles for a user to notice a significant slowdown.

Refer to Configuring use case 3: Protection against ping flood attacks on page 38.

## Use case 4: Protection against UDP flood attacks

A User Datagram Protocol (UDP) flood is a brute force DoS attack where a large number of UDP packets are sent by the attacker to random ports on a remote host.

In a UDP attack, the targeted system is forced to reply to the UDP packets with ICMP Destination Unreachable packets, eventually leading the target system becomes unreachable to other clients. The targeted system responds to a UDP flood by:

**Checking for the application listening at that port > Seeing that no application listens at that port > Replies with an iCMP Destination Unreachable packet**

The attacker may also spoof the IP address of the UDP packets, ensuring that the excessive ICMP return packets do not reach them, and anonymize their network location.

Refer to Configuring use case 4: Protection against UDP flood attacks on page 40.

### *Configuring all the use cases for ACL traffic filtering*

You can configure all four use cases and apply them to an ingress port by following these high level steps.

1. Create an ACL, with criteria that matches the potential attack.
   - A standard ACL table provides the option to filter only based on source address information.
   - An extended ACL table provides the option to filter based on most of the fields in the packet header.
2. Create a Class Map, associate to that ACL.
3. Create a Policy Map using the Class Map created in step 2, and assign a Policer.
4. Associate that Policy Map to an ingress port.

Refer to Configuring and applying all four use cases for ACL-based traffic filtering on page 43.

# VLAN-based rate limiting

VLAN-based rate limiting provides specific bandwidth for the inbound traffic on the VLAN on a physical port, port channel, and system-wide.

Ingress traffic on both tagged and untagged VLAN are rate limited. A packet can be classified for QoS policing by using the VLAN ID match criteria. A class map is configured to match this match criteria before the QoS policing action is taken. Each class map can match on a VLAN ID. Multiple class maps can reside within a policy map. When the system is configured to use VLAN-based rate limiting, the traffic received on this interface is classified, policed, and marked according to the policy map attached to the VLAN to which the packet belongs.

Separate ACLs matching traffic based on VLAN can exist on the device. VLAN-based rate limiting can coexist with existing Layer 2 MAC ACLs.

## Configuration considerations and limitations for VLAN-based rate limiting

- VLAN-based rate limiting is applicable for ingress traffic only. Egress traffic is blocked by the SLX-OS device.
- There is one policer per ACL, it applies to all the rules for that ACL.
- Currently, 1,024 TCAM entries are shared between VLAN rate limiting and DAI features on a first-come first serve basis.
- Packets drops caused by any action other than ACL rate limiting are included in policer counters.
- Only a permit clause in an ACL rule is subject to rate limit traffic calculations. A deny clause does not result in a policing action.
- Metering is performed at Layer 1 on the packet size as received on the wire (including IPG, Preamble and SOF, ignoring FCS/CRC).
- A policy map can be applied to a physical port, LAG interface, as well as a system.
- Multiple class maps with user ACLs can be added in a policy map, however, once a matching ACL clause is found, the device does not evaluate subsequent ACL clauses and subsequently rate limits those ACLs.
- A configured rate in bps is rounded up to next achievable rate.
- The device does not support the specifying of actions to perform on traffic with a color-class priority, such as having packet DSCP priority, traffic class (internal queue assignment), or traffic class (internal queue assignment) set to specific values.
- Statistics display only two colors. Conform includes Green and Yellow color packets. Violate includes dropped or a RED color packet.
- You cannot delete a policy map if it is active on the interface.
- You cannot delete a class map from a policy map when the policy map is active on the interface.

- Configure CIR and EIR in multiples of 22 kbps. If it is below 22 kbps, then 22 kbps is programmed in the device.
- If a police map is applied to an interface and no Policer attributes are present in that policy map, then ingress and egress packets on that interface is marked as green (conforming).
- If you does not configure EIR, then the Single rate Three Color scheme (SrTCM) is applied.
- You must configure rate limit threshold values on an interface based on interface speed.
- No validation is performed for user-configured values against interface speed.
- A policy map cannot be bound to interface if class map is not associated with that policy map.
- All VLAN ports within the same tower is governed by the same policer. A policer is per tower based. One ACL entry per VLAN is per tower.

# Configuring VLAN-based rate limiting

Perform the following steps to apply a VLAN for traffic filtering and policing.

1. In privileged EXEC mode, access global configuration mode.

```
device# configure terminal
```

2. Create the class map and access its configuration mode.

```
device(config)# class-map c2
```

3. Add a match statement to the VLAN.

```
device(config-classmap)# match vlan 500
```

4. Access global configuration mode.

```
device(config-classmap)# exit
```

5. Create the policy map and access its configuration mode.

```
device(config)# policy-map p2
```

6. Add the class map to the policy map and access the policy-class configuration mode.

```
device(config-policymap)# class c2
```

7. Configure the class map.

```
device(config-policymap-class)# police cir 100000 cbs 6000
```

   This step configures the committed information rate for the class map.

8. Access global configuration mode.

```
device(config-policymap-class)# exit
device(config-policymap)# exit
```

9. Access the interface configuration mode for the interface that you want to apply the policy map.

```
device(config)# interface ethernet 0/1
```

10. Apply the policy map to the interface.

```
device(conf-if-eth-0/1)# service-policy in p2
```

11. Access global configuration mode.

```
device(conf-if-eth-0/1)# exit
```

12. Optionally, apply the policy map globally.

```
device(config)# qos service-policy in p2
```

The following example provides the configuration in the previous steps.

```
device# configure terminal
device(config)# class-map c2
device(config-classmap)# match vlan 500
device(config-classmap)# exit
device(config)# policy-map p2
device(config-policymap)# class c2
device(config-policymap-class)# police cir 100000 cbs 6000
device(config-policymap-class)# exit
device(config-policymap)# exit
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# service-policy in p2
device(conf-if-eth-0/1)# exit
device(config)# qos service-policy in p2
```

## *Displaying information related to VLAN-based rate limiting*

To display rate limiting policies implemented in the configured policy maps, and policer confirm, exceeded, or violate counters, use the show **policy-map** command.

The following example displays the binding of the policy maps.

```
device# show policy-map
Number of policy maps : 1
Policy-Map p2
  Bound To: Eth 0/1(in)
....
```

The following example displays the interface-specific policy-map information.

```
device# show policy-map interface ethernet 0/1 in

Ingress Direction :
Policy-Map p2
Class c2
matches 7867567 packets 1007048576 bytes
Police cir 1000000
       Stats:
         Operational cir:1010000 cbs:149999 eir:0 ebs:0
         Conform Byte:1180928 Exceed Byte:0 Violate Byte:1005867648
```

To clear the policer counters for the interface, use the **clear policy-map-counters interface ethernet** command. If you do not specify an interface, all rate limit counters are cleared on all interfaces.

The following example displays the system-specific policy-map information.

```
device# # show policy-map system map-name pm1

Ingress Direction :
Policy-Map pm1
Class cm1
matches 480661 packets 61524608 bytes
Police cir 100000
Stats:
 Operational cir:109000 cbs:14999 eir:0 ebs:0
  Conform Byte:265088 Exceed Byte:0 Violate Byte:0
```

To display the policy-map configuration information, use the **show running-config policy-map** command.

```
device# show running-config show running-config policy-map
policy-map p2
 class c2
  police cir 100000 cbs 6000
 !
!
```

To display the class map information, use the **show running-config class-map** command.

```
device# show running-config class-map
class-map c2
 match vlan 500
```

# Bridged-domain based rate limiting

Bridged-domain based rate limiting applies to a specific logical interfaces (LIFs) and is performed at each virtual switching instance representing the AC or LIFs.

For more information on configuring these instances, refer to the *Extreme SLX-OS Layer 2 Switching Configuration Guide*.

## Configuration considerations and limitations for bridge-domain based rate limiting

- BD based rate limiting coexists and works in parallel with all other ACL and rate-limiting features.
- BD based rate limiting is applicable for ingress traffic only. Egress traffic is blocked by the SLX-OS device.
- There is one policer per ACL, it applies to all the rules for that ACL.
- Currently, 1,024 TCAM entries are shared between BD rate limiting and DAI features on a first-come first serve basis.
- Packets drops caused by any action other than ACL rate limiting are included in policer counters.
- Only a permit clause in an ACL rule is subject to rate limit traffic calculations. A deny clause does not result in a policing action.
- Metering is performed at Layer 1 on the packet size as received on the wire (including IPG, Preamble and SOF, ignoring FCS/CRC).
- A policy map can be applied to a physical port, LAG interface, as well as a system.
- Multiple class maps with user ACLs can be added in a policy map, however, once a matching ACL clause is found, the device does not evaluate subsequent ACL clauses and subsequently rate limits those ACLs.
- A configured rate in bps is rounded up to next achievable rate.
- The device does not support the specifying of actions to perform on traffic with a color-class priority, such as having packet DSCP priority, traffic class (internal queue assignment), or traffic class (internal queue assignment) set to specific values.
- Statistics display only two colors. Conform includes Green and Yellow color packets. Violate includes dropped or a RED color packet.
- You cannot delete a policy map if it is active on the interface.
- You cannot delete a class map from a policy map when the policy map is active on the interface.
- Configure CIR and EIR in multiples of 22 kbps. If it is below 22 kbps, then 22 kbps is programmed in the device.
- If a police map is applied to an interface and no Policer attributes are present in that policy map, then ingress and egress packets on that interface is marked as green (conforming).
- If the CBS and EBS values are not configured, then these values are derived from CIR and EIR values, respectively. The burst size calculation is as follows: Burst size (CBS or EBS) =1.2*information rate (CIR/EIR)/8.

- If the configured CBS value is less than 2*MTU value, then 2*MTU is programmed as the CBS in the hardware.
- If you does not configure EIR, then the Single rate Three Color scheme (SrTCM) is applied.
- You must configure rate limit threshold values on an interface based on interface speed.
- No validation is performed for user-configured values against interface speed.
- A policy map cannot be bound to interface if class map is not associated with that policy map.
- All VLAN ports within the same tower is governed by the same policer. A policer is per tower based. One ACL entry per VLAN is per tower.

# Configuring bridge-domain based rate limiting

Perform the following steps to apply a bridge domain for traffic filtering and policing.

1. In privileged EXEC mode, access global configuration mode.

```
device# configure terminal
```

2. Create the class map and access its configuration mode.

```
device(config)# class-map BD-1000
```

3. Add a match statement to the bridge domain.

```
device(config-classmap)# match bridge-domain 1000
```

4. Access global configuration mode.

```
device(config-classmap)# exit
```

5. Create the policy map and access its configuration mode.

```
device(config)# policy-map QOS-BD
```

6. Add the class map to the policy map and access the policy-class configuration mode.

```
device(config-policymap)# class BD-1000
```

7. Configure the class map.

```
device(config-policymap-class)# police cir 100000 cbs 6000
```

   This step configures the committed information rate for the class map.

8. Access global configuration mode.

```
device(config-policymap-class)# exit
device(config-policymap)# exit
```

9. Access the interface configuration mode for the interface that you want to apply the policy map.

```
device(config)# interface ethernet 0/4
```

10. Apply the policy map to the interface.

```
device(conf-if-eth-0/4)# service-policy in QOS-BD
```

11. Access global configuration mode.

```
device(conf-if-eth-0/4)# exit
```

12. Optionally, apply the policy map globally.

```
device(config)# qos service-policy in QOS-BD
```

The following example provides the configuration in the previous steps.

```
device# configure terminal
device(config)# class-map BD-1000
device(config-classmap)# match bridge-domain 1000
device(config-classmap)# exit
device(config)# policy-map QOS-BD
device(config-policymap)# class BD-1000
device(config-policymap-class)# police cir 100000 cbs 6000
device(config-policymap-class)# exit
device(config-policymap)# exit
device(config)# interface ethernet 0/4
device(conf-if-eth-0/4)# service-policy in QOS-BD
device(conf-if-eth-0/4)# exit
device(config)# qos service-policy in QOS-BD
```

# Displaying information related to bridge-domain based rate limiting

To display rate limiting policies implemented in the configured policy maps, and policer confirm, exceeded, or violate counters, use the show **policy-map** command.

The following example displays the binding of the policy maps.

```
device# show policy-map
Number of policy maps : 1
Policy-Map QOS-BD
  Bound To: Eth 0/4(in)
....
```

The following example displays the interface-specific policy-map information.

```
device# show policy-map interface ethernet 0/4 in

Ingress Direction :
Policy-Map QOS-BD
Class BD-1000
matches 7867567 packets 1007048576 bytes
Police cir 1000000
       Stats:
         Operational cir:1010000 cbs:149999 eir:0 ebs:0
         Conform Byte:1180928 Exceed Byte:0 Violate Byte:1005867648
```

To clear the policer counters for the interface, use the **clear policy-map-counters interface ethernet** command. If you do not specify an interface, all rate limit counters are cleared on all interfaces.

The following example displays the system-specific policy-map information.

```
device# # show policy-map system map-name QOS-BD

Ingress Direction :
Policy-Map QOS-BD
Class BD-1000
matches 480661 packets 61524608 bytes
Police cir 100000
Stats:
 Operational cir:109000 cbs:14999 eir:0 ebs:0
  Conform Byte:265088 Exceed Byte:0 Violate Byte:0
```

To display the policy-map configuration information, use the **show running-config policy-map** command.

```
device# show running-config show running-config policy-map
policy-map QOS-BD
 class BD-1000
  police cir 100000 cbs 6000
 !
!
```

To display the class map information, use the **show running-config class-map** command.

```
device# show running-config class-map
class-map BD-1000
 match bridge-domain 1000
```

# Rate limiting scaling limits

The following list provides the rate limiting scaling limits:

- Number of policy-maps in a system—1,024
- Number of class-maps in a system—32,712
- Number of distinct policer instances (CIR, CBS, EIR, or EBS) in a chip—1,023
- Number of class-maps in a policy map—4,096
- Number of supported policers in a system—32,712
- The number of supported port, BUM, ACL, VLAN, and BD-based rate limiting entries in a chip depends on the configured TCAM profile and the number of resource available for the subtype.

  NOTE
  The TCAM entries allocated for the rate limiting subtypes remains the same as the 17r.1.02 release.
  The 4,096 policers (class maps) are not applicable to port channels. Only 1,215 policers are reserved for port channels.

# Configuring traffic policing

Follow these tasks to configure traffic policing.

## Configuring a class map using an ACL

To configure a classification or class map by using an ACL, follow these steps.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create an IP access list to define the traffic.
   a) Create and name a standard IP access list and enter IP ACL configuration mode.

```
device(config)# ip access-list standard ip_acl
```

   b) Allow traffic from a specific IP address.

```
device(conf-ipacl-std)# permit host 10.10.10.0
```

   c) Exit IP ACL configuration mode to global configuration mode.

```
device(conf-ipacl-std)# exit
```

   For details on creating access lists, refer to the *Extreme SLX-OS Security Configuration Guide* for the device.
3. Verify the IP ACL.

```
device(config)# do show running-config | include ip_acl
ip access-list standard ip_acl
```

4. Create and name a class map.

```
device(config)# class-map class_1
```

5. Provide match criteria for the class.

```
device(config-classmap)# match access-group ip_acl
```

6. Return to privileged EXEC mode.

```
device(config-classmap)# end
```

7. Verify the class configuration.

```
device# show running-config | include class
...
class-map cee
class-map class_1
class-map default
```

8. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

## *Class map using an ACL configuration example*

```
device# configure terminal
device(config)# ip access-list standard IP_acl
device(conf-ipacl-std)# permit host 10.10.10.0
device(conf-ipacl-std)# exit
device(config)# do show running-config | include ip_acl
device(config)# class-map class_1
device(config-classmap)# match access-group ip_acl
device(config-classmap)# end
device# show running-config | include class
device# copy running-config startup-config
```

# Configuring a policy map

Add a class map to a policy map and set policing parameters to the class map.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create and name a policy map.

```
device(config)# policy-map policy_2
```

3. Add a class map to the policy map.

```
device(config-policymap)# class default
```

4. Create a policy map class police instance and set the committed information rate (cir), committed burst rate (cbs), excess information rate (eir), and the excess burst rate (ebs).

```
device(config-policymap-class)# police cir 3000000 cbs 375000000 eir 300000000 ebs 37500000
```

5. Return to privileged EXEC mode.

```
device(config-policymap-class)# end
```

6. Verify the configuration.

```
device# show policy-map detail policy_2

Policy-Map policy_2
    Class default
      Police cir 3000000 cbs 375000000 eir 300000000 ebs 37500000

  Bound To:None
```

7. Save the configuration.

```
device# copy running-config startup-config
```

## *Policy map configuration example*

```
device# configure terminal
device(config)# policy-map policy_2
device(config-policymap)# class default
device(config-policymap-class)# police cir 3000000 cbs 375000000 eir 300000000 ebs 37500000
device(config-policymap-class)# end
device# show policy-map detail policy_2
device# copy running-config startup-config
```

# Configuring port-based traffic policing

Follow these steps to associate the policy map with the Interface. By associating the policy map, the policing parameters are applied to the port.

Use an ingress or egress policy map that has been created and populated with policing parameters.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode

```
device(config)# interface ethernet 0/3
```

3. Attach an input policy map.
4. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# end
```

5. Verify the configuration.

```
device# show policy-map interface ethernet 0/3

Ingress Direction :
 Policy-Map policy_2
    Class default
      Police cir 1000000000 eir 1000000000 classification-type color remark-profile default
        Stats:
          Operational cir:1000000000 cbs:149999999 eir:1000000000 ebs:149999999
          Conform Byte:0 Exceed Byte:0 Violate Byte:0
```

### *Port-based traffic policing configuration example*

```
device# configure terminal
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policy_2
device(conf-if-eth-0/3)# end
device# show policy-map interface ethernet 0/3 in
```

## Configuring ACL-based rate limiting

You can configure ACL-based rate limiting for protection against TCP SYN, TCP RST, ping flood, and UDP flood attacks. In the following use cases for these attacks, you configure the ACL that is used to protect against the attack, bind the ACL to an interface, and configure and apply the ACL traffic filtering.

### *Configuring use case 1: Protection against TCP SYN attacks*

Follow these steps to configure an ACL that can be used to protect against TCP SYN DoS attacks.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create an extended IP ACL.

```
device(config)# ip access-list extended acl1
2015/04/01-13:18:15, [SSMD-1400], 2315, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 is created.
```

   The system message is generated when you create an ACL. If you are configuring an existing ACL, no message is generated.

3. Configure the extended ACL to permit TCP traffic from any source to any destination while filtering packets for which the **sync** (synchronize) flag is set.

```
device(conf-ipacl-ext)# permit tcp any any sync
2015/04/01-13:22:16, [SSMD-1404], 2316, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 rule sequence number 10 is added.
```

4. Return to privileged EXEC mode.

```
device(conf-ipacl-ext)# end
```

5. Verify the ACL.

```
device# show running-config ip access-list extended acl1
ip access-list extended acl1
 seq 10 permit tcp any any sync
```

### Protection against TCP SYN attacks - ACL configuration example

```
device# configure terminal
device(config)# ip access-list extended acl1
device(conf-ipacl-ext)# permit tcp any any sync
device(conf-ipacl-ext)# end
device# show running-config ip access-list extended acl1
```

## *Configuring use case 1: Bind the TCP SYN ACL to an interface*

To protect against TCP SYN DoS attacks, bind ACL-based protection against TCP SYN attacks to an interface.

You have configured an extended Layer 3 ACL-based rate limit matching TCP SYN.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a class map.

```
device(config)# class-map aclFilter
```

   The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

3. While in class map mode associate the class map with an ACL.

```
device(config)# match access-group acl1
```

   The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

4. Return to privileged EXEC mode.

```
device(config-classmap)# end
```

5. Verify the class map to ACL association.

```
device# show running-config class-map aclFilter
 class-map aclFilter
  match access-group acl1
!
```

6. Create a policy map with a policer.

```
device(config)# policy-map policyAclFilter
```

   A policy map is used to apply policer and QoS attributes to a particular interface.

7. Associate a class map with the policy map.

```
device(config-policymap)# class aclFilter
```

   Each policy map can have different class maps. Each class map in the policy map can be associated to separate policing and QoS parameters.

8. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
```

   CIR and EIR are in increments of 22000 bps.

9. Return to privileged Exec mode.

```
device(config-policymap-class-police)# end
```

10. Verify the configuration.

```
device# show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class aclFilter
      Police cir 220000 cbs 50000 eir 36000 ebs 400000

  Bound To:None
```

11. Enter interface configuration mode.

```
device(config)# interface ethernet 0/3
```

12. Bind the policy map to the port.

```
device(conf-if-eth-0/3)# service-policy in policyAclFilter
2016/02/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 configured on interface Ethernet 0/33 at Ingress by FbQos_9_11.
```

13. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# end
```

14. Verify the configuration.
15. Save the configuration.

```
device# copy running-config startup-config
```

### ACL-based protection against TCP SYN attacks applied to an interface configuration example

## *Configuring use case 2: Protection against TCP RST attacks*

Follow these steps to configure an ACL that can be used to protect against TCP RST DoS attacks.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create or invoke an extended IP ACL.

```
device(config)# ip access-list extended acl1
2015/04/01-13:18:15, [SSMD-1400], 2315, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 is created.
```

   The system message is generated when you create an ACL. If you are configuring an existing ACL, no message is generated.

3. Configure the extended ACL to permit TCP traffic from any source to any destination while filtering packets for which the **rst** flag is set.

```
device(conf-ipacl-ext)# permit tcp any any rst
2015/04/01-13:22:16, [SSMD-1404], 2316, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 rule sequence number 10 is added.
```

4. Return to privileged EXEC mode.

```
device(conf-ipacl-ext)# end
```

5. Verify the ACL.

```
device# show running-config ip access-list extended acl1
ip access-list extended acl1
 seq 10 permit tcp any any rst
```

### Protection against TCP RST attacks - ACL configuration example

```
device# configure terminal
device(config)# ip access-list extended acl1
device(conf-ipacl-ext)# permit tcp any any rst
device(conf-ipacl-ext)# exit
device# show running-config ip access-list extended acl1
```

## *Configuring use case 2: Bind the TCP RST ACL to an interface*

To protect against TCP RST DoS attacks, bind an extended Layer 3 ACL based rate limit matching TCP RST to an interface.

A TCP RST matching ACL has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a class map.

```
device(config)# class-map aclFilter
```

   The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

3. While in class map mode associate the class map with an ACL.

```
device(config)# match access-group acl1
```

   The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

4. Return to privileged EXEC mode.

```
device(config-classmap)# end
```

5. Verify the class map to ACL association.

```
device# show running-config class-map aclFilter
 class-map aclFilter
  match access-group acl1
!
```

6. Create a policy map with a policer.

```
device(config)# policy-map policyAclFilter
```

A policy map is used to apply policer and QoS attributes to a particular interface.

7. Associate a class map with the policy map.

```
device(config-policymap)# class aclFilter
```

Each policy map can have different class maps. Each class map in the policy map can be associated to separate policing and QoS parameters.

8. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
```

CIR and EIR are in increments of 22000 bps.

9. Return to privileged EXEC mode.

```
device(config-policymap-class-police)# end
```

10. Verify the configuration.

```
device# show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class aclFilter
      Police cir 220000 cbs 50000 eir 36000 ebs 400000

  Bound To:None
```

11. Enter interface configuration mode.

```
device(config)# interface ethernet 0/3
```

12. Bind the policy map to the port.

```
device(conf-if-eth-0/3)# service-policy in policyAclFilter
2016/02/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 configured on interface Ethernet 0/3 at Ingress by FbQos_9_11.
```

13. Verify the configuration.

```
device# show policy-map detail policyAclFilter
Policy-Map policyAclFilter
 Class aclFilter
 Police cir 220000 cbs 50000 eir 36000 ebs 400000
 Bound To: Et 03(in)
```

14. Save the configuration.

```
device# copy running-config startup-config
```

### ACL-based protection against TCP RST attacks applied to an interface configuration example

```
device# configure terminal
device(config)# class-map aclFilter
device(config)# match access-group acl1
device(config-classmap)# end
device# show running-config class-map aclFilter
device(config)# policy-map policyAclFilter
device(config-policymap)# class aclFilter
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

## *Configuring use case 3: Protection against ping flood attacks*

Follow these steps to configure an ACL that can be used to protect against ping flood attack.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create or invoke an extended IP ACL.

   ```
   device(config)# ip access-list extended acl1
   2015/04/01-13:18:15, [SSMD-1400], 2315, SW/device | Active | DCE, INFO, device, IPv4 access list
   acl1 is created.
   ```

   The system message is generated when you create an ACL. If you are configuring an existing ACL, no message is generated.

3. Configure the extended ACL to filter ICMP packets.

   ```
   device(conf-ipacl-ext)# permit icmp any any
   2015/04/02-11:44:45, [SSMD-1404], 2501, SW/device | Active | DCE, INFO, device, IPv4 access list
   acl1 rule sequence number 10 is added.
   ```

4. Return to privileged EXEC mode.

   ```
   device(conf-ipacl-ext)# end
   ```

5. Verify the ACL.

   ```
   device# show running-config ip access-list extended acl1
   ip access-list extended acl1
    seq 10 permit icmp any any
   ```

### Protection against ping attacks - ACL configuration example

```
device# configure terminal
device(config)# ip access-list extended acl1
device(conf-ipacl-ext)# permit icmp any any
device(conf-ipacl-ext)# end
device# show running-config ip access-list extended acl1
```

## *Configuring use case 3: Bind the ping flood attack ACL to an interface*

To protect against ping flood DoS attacks, bind an extended Layer 3 ACL-based rate limit to filter ICMP packets and bind it to an interface.

You have configured an extended Layer 3 ACL-based rate limit to filter ICMP packets.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a class map.

```
device(config)# class-map aclFilter
```

   The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

3. While in class map mode associate the class map with an ACL.

```
device(config)# match access-group acl1
```

   The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

4. Return to privileged EXEC mode.

```
device(config-classmap)# end
```

5. Verify the class map to ACL association.

```
device# show running-config class-map aclFilter
 class-map aclFilter
  match access-group acl1
!
```

6. Create a policy map with a policer.

```
device(config)# policy-map policyAclFilter
```

   A policy map is used to apply policer and QoS attributes to a particular interface.

7. Associate a class map with the policy map.

```
device(config-policymap)# class aclFilter
```

   Each policy map can have different class maps. Each class map in the policy map can be associated to separate policing and QoS parameters.

8. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
```

   CIR and EIR are in increments of 22000 bps.

9. Return to privileged EXEC mode.

```
device(config-policymap-class-police)# end
```

10. Verify the configuration.

```
device# show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class aclFilter
      Police cir 22000 cbs 50000 eir 36000 ebs 400000

  Bound To:None
```

11. Bind the policy map to the port.

```
device(conf-if-eth-0/3)# service-policy in policyAclFilter
2016/02/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 configured on interface Ethernet 0/3 at Ingress by FbQos_9_11.
```

12. Verify the configuration.

```
device# show policy-map detail policyAclFilter
Policy-Map policyAclFilter
 Class aclFilter
 Police cir 220000 cbs 50000 eir 36000 ebs 400000
 Bound To: Eth 0/3(in)
```

13. Save the configuration.

```
device# copy running-config startup-config
```

### ACL-based protection against ping attacks applied to an interface configuration example

```
device# configure terminal
device(config)# class-map aclFilter
device(config)# match access-group acl1
device(config-classmap)# end
device# show running-config class-map aclFilter
device(config)# policy-map policyAclFilter
device(config-policymap)# class aclFilter
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

## *Configuring use case 4: Protection against UDP flood attacks*

Follow these steps to configure an ACL that can be used to protect against UDP flood attacks.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create or invoke an extended IP ACL.

```
device(config)# ip access-list extended acl1
2015/04/01-13:18:15, [SSMD-1400], 2315, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 is created.
```

The system message is generated when you create an ACL. If you are configuring an existing ACL, no message is generated.

3. Configure the extended ACL to filter UDP packets.

```
device(conf-ipacl-ext)# permit udp any any
2015/04/02-11:44:45, [SSMD-1404], 2501, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 rule sequence number 10 is added.
```

4. Return to privileged EXEC mode.

```
device(conf-ipacl-ext)# end
```

5. Verify the ACL.

```
device(config)# do show running-config ip access-list extended acl1
ip access-list extended acl1
 seq 10 permit udp any any
```

### Protection against UDP flood attacks - ACL configuration example

```
device# configure terminal
device(config)# ip access-list extended acl1
device(conf-ipacl-ext)# permit udp any any
device(conf-ipacl-ext)# end
device# show running-config ip access-list extended acl1
```

## *Configuring use case 4: Bind the UDP ACL to an interface*

A UDP flood attack is a brute force type of DoS attack where a large number of UDP packets are sent to random ports on the targeted system

You have configured an extended Layer 3 UDP ACL.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a class map.

```
device(config)# class-map aclFilter
```

The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

3. While in class map mode, associate the class map with an ACL.

```
device(config)# match access-group acl1
```

The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

4. Return to privileged EXEC mode.

```
device(config-classmap)# end
```

5. Verify the class map to ACL association.

```
device# show running-config class-map aclFilter
 class-map aclFilter
  match access-group acl1
!
```

6. Create a policy map with a policer.

```
device(config)# policy-map policyAclFilter
```

   A policy map is used to apply policer and QoS attributes to a particular interface.

7. Associate a class map with the policy map.

```
device(config-policymap)# class aclFilter
```

   Each policy map can have different class maps. Each class map in the policy map can be associated to separate policing and QoS parameters.

8. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
```

   CIR and EIR are in increments of 22000 bps.

9. Return to privileged EXEC mode.

```
device(config-policymap-class-police)# end
```

10. Verify the configuration.

```
device# show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class aclFilter
      Police cir 220000 cbs 50000 eir 36000 ebs 400000

  Bound To:None
```

11. Bind the policy map to the port.

```
device(conf-if-eth-0/3)# service-policy in policyAclFilter
2016/02/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4 access list
acl1 configured on interface Ethernet 0/3 at Ingress by FbQos_9_11.
```

12. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# end
```

13. Verify the configuration.

```
device# show policy-map detail policyAclFilter
Policy-Map policyAclFilter
 Class aclFilter
 Police cir 220000 cbs 50000 eir 36000 ebs 400000
 Bound To: Eth 0/3(in)
```

14. Save the configuration.

```
device# copy running-config startup-config
```

### ACL-based protection against UDP flood attacks applied to an interface configuration example

```
device# configure terminal
device(config)# class-map aclFilter
device(config)# match access-group acl1
device(config-classmap)# end
device# show running-config class-map aclFilter
device(config)# policy-map policyAclFilter
device(config-policymap)# class aclFilter
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

## *Configuring and applying all four use cases for ACL-based traffic filtering*

Follow these steps to apply ACLs for traffic filtering.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create an ACL.

   ```
   device(config)# ip access-list extended acl1
   2015/04/02-13:22:39, [SSMD-1400], 2506, SW/device | Active | DCE, INFO, device, IPv4 access list
   acl1 is created.
   ```

   The system message is generated when you create an ACL. If you are configuring an existing ACL, no message is generated.

3. Configure the extended ACL to filter packets for which the **sync** (synchronize) flag is set.

   ```
   device(conf-ipacl-ext)# permit tcp any any sync
   2015/04/02-13:25:28, [SSMD-1404], 2507, SW/device | Active | DCE, INFO, device, IPv4 access list
   acl1 rule sequence number 10 is added.
   ```

   This step provides protection from TCP SYN attacks.

4. Configure the extended ACL to filter packets for which the **rst** flag is set.

   ```
   device(conf-ipacl-ext)# permit tcp any any
   rst

   2015/04/02-13:26:48, [SSMD-1404], 2508, SW/device | Active | DCE, INFO, device, IPv4 access list
   acl1 rule sequence number 20 is added.
   ```

   This step provides protection from TCP RST attacks.

5. Configure the extended ACL to filter ICMP packets.

   ```
   device(conf-ipacl-ext)# permit icmp any any
   2015/04/02-13:28:20, [SSMD-1404], 2509, SW/device | Active | DCE, INFO, device, IPv4 access list
   acl1 rule sequence number 30 is added.
   ```

   This step protects against ping flood attacks.

6. Configure the extended ACL to filter UDP packets.

   ```
   device(conf-ipacl-ext)# permit udp any any
   2015/04/02-13:30:15, [SSMD-1404], 2510, SW/device | Active | DCE, INFO, device, IPv4 access list
   acl1 rule sequence number 40 is added.
   ```

   This step protects against UDP flood attacks.

7. Return to global configuration mode.

```
device(conf-ipacl-ext)# exit
```

8. Verify the ACL.

```
device(config)# do show running-config ip access-list extended acl1
ip access-list extended acl1
 seq 10 permit tcp any any sync
 seq 20 permit tcp any any rst
 seq 30 permit icmp any any
 seq 40 permit udp any any
!
```

9. Create a class map.

```
device(config)# class-map aclFilter
```

   The class map is used to classify the traffic; different match conditions, including an ACL, can be used to match the traffic properties.

10. While in class map mode associate the class map with an ACL.

```
device(config-classmap)# match access-group acl1
```

11. Return to global configuration mode.

```
device(config-classmap)# exit
```

12. Verify the class map to ACL association.

```
device(config)# do show running-config class-map aclFilter
class-map aclFilter
 match access-group acl1
!
```

13. Create a policy map with a policer.

```
device(config)# policy-map policyAclFilter
```

   A policy map is used to apply policer and QoS attributes to a particular interface.

14. Associate a class map with the policy map.

```
device(config-policymap)# class aclFilter
```

   Each policy map can have different class maps. Each class map in the policy map can be associated to separate policing and QoS parameters.

15. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
```

   CIR and EIR are in increments of 22000 bps.

16. Return to privileged EXEC mode.

```
device(config-policymap-class-police)# end
```

17. Verify the configuration.

```
device# show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class aclFilter
      Police cir 220000 cbs 50000 eir 36000 ebs 400000

  Bound To:None
```

18. Enter global configuration mode.

```
device# configure terminal
```

19. Enter interface configuration mode.

```
device(config)# interface ethernet 0/3
```

20. Bind the policy map to the port.

```
device(conf-if-eth-0/3)# service-policy in policyAclFilter
2016/02/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device,
IPv4 access list acl1 configured on interface Ethernet 0/3 at Ingress by FbQos_9_11.
```

21. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# end
```

22. Verify the configuration.

```
device# show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class aclFilter
      Police cir 220000 cbs 50000 eir 36000 ebs 400000

  Bound To: Eth 0/3(in)
```

23. Save the configuration.

```
device# copy running-config startup-config
```

### ACL-based traffic filtering to protect from DoS attacks configuration example

```
device# configure terminal
device(config)# ip access-list extended acl1
device(conf-ipacl-ext)# permit tcp any any sync
device(conf-ipacl-ext)# permit tcp any any rst
device(conf-ipacl-ext)# permit icmp any any
device(conf-ipacl-ext)# permit udp any any
device(config)# do show running-config ip access-list extended acl1
device(config)# class-map aclFilter
device(config-classmap)# match access-group acl1
device(config-classmap)# exit
device(config)# do show running-config class-map aclFilter
device(config)# policy-map policyAclFilter
device(config-policymap)# class aclFilter
device(config-policymap-class)# police cir 220000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device# configure terminal
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

# Storm control - rate limiting broadcast, unknown unicast, and multicast traffic

A broadcast, unknown unicast, and multicast (BUM) traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance.

BUM storm control can prevent disruptions on Layer 2 physical ports. This feature is supported only at the interface level.

BUM storm control allows you to limit the amount of broadcast, unknown unicast, and multicast ingress traffic on a specified interface. All traffic received in excess of the configured rate is discarded.

## Storm control considerations and limitations

- BUM storm control applies only to ingress traffic.
- BUM can only be configured on physical interfaces.
- BUM storm control and input service-policy features can coexist with each other. You can enable both at the same time on a given interface. BUM storm control has the highest precedence.
- For LAG ports, BUM rate limiting needs to be enabled on each LAG member port.
- A single rate two color marking scheme is used.
- Metering is performed on the packet size as received on the wire (including IPG and preamble), ignoring CRC.
- If BUM traffic is also classified by an ACL, then BUM rate limiting is not effective
- The configured rate in bits per second (bps) is rounded up to next achievable rate.

# Configuring storm control on an Ethernet interface

Perform the following steps to configure BUM storm control on an Ethernet interface.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Specify the Ethernet interface for the traffic you want to control.

```
device(config)# interface ethernet 0/3
```

3. Issue the storm control ingress command to set a traffic limit for broadcast traffic on the interface..

```
device(conf-if-eth-0/3)# storm-control ingress broadcast limit-bps 400000
```

   In this example you set a control on the inbound broadcast traffic, limiting the rate to 400000 bits per second (bps).

4. Issue the storm control ingress command to set a traffic limit for unknown-unicast traffic on the interface..

```
device(conf-if-eth-0/3)# storm-control ingress unknown-unicast limit-bps 50000000
```

   In this example you set a control on the inbound unknown-unicast traffic, limiting the rate to 50000000 bps.

5. Issue the storm control ingress command to set a traffic limit for multicast traffic on the interface..

```
device(conf-if-eth-0/3)# storm-control ingress multicast limit-percent 3
```

   In this example you set a control on the inbound multicast traffic, limiting the rate to 3% of traffic.

6. Return to privileged EXEC mode.

```
device(conf-if-eth-0/3)# end
```

7. Verify the storm control configuration.

```
device# show run | include storm-control
 storm-control ingress broadcast limit-bps 400000
 storm-control ingress multicast limit-percent 3
 storm-control ingress unknown-unicast limit-bps 50000000
```

8. Save the configuration.

```
device# save running-config startup-config
```

## *BUM storm control configuration example*

```
device# configure terminal
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# storm-control ingress broadcast limit-bps 400000
device(conf-if-eth-0/3)# storm-control ingress unknown-unicast limit-bps 50000000
device(conf-if-eth-0/3)# storm-control ingress multicast limit-percent 3
device(conf-if-eth-0/3)# end
device# show storm-control
device# save running-config startup-config
```

# Quality of Service



# QoS overview

Quality of Service (QoS) provides preferential treatment to specific traffic.

By offering preferential treatment to specific traffic, other traffic may be stopped or slowed. Without QoS, the device offers best-effort service to each packet and transmits packets without any assurance of reliability, delay bounds, or throughput. Implementing QoS in a network makes performance more predictable and bandwidth utilization more effective.

## QoS features

The principal QoS features are as follows.

- Processing capability of 880 Gbps
- 12 MB buffer
- 61K cell for 208B each
- 3k unicast queues and 10 multicast queues for each ports, however only eight are used
- Support for the following:
  - Port shaping
  - Priority shaping
  - Strict priority (SP)
  - Weighted round robin (WRR)

  NOTE
  Packet Cell Packing is not supported. Different packets cannot be packed on the same cell.

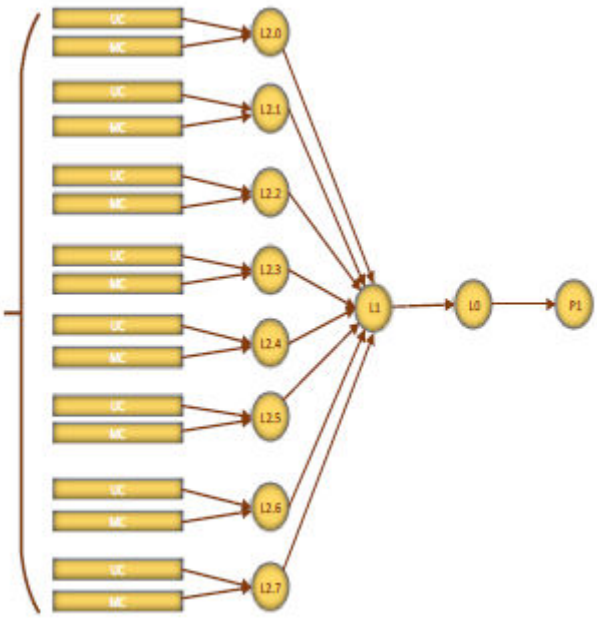
### *Egress shaping and scheduling*

QOS hierarchy for each egress port has three level of scheduling and eight priority queue pairs.

Scheduling supports:

- At L1, SP and WRR, or mixed mode for both unicast and multicast traffic
- At L0, per port shaping
- At L2, per priority shaping

**FIGURE 3** Egress shaping and scheduling




# IEEE 802.1q ToS-DSCP header fields

The Type of Service (ToS), now known as Differentiated Services (DS), defines a mechanism for assigning a priority to each IP packet as well as a mechanism to request specific treatment such as high throughput, high reliability or low latency.

The 8 bit ToS field originally defined a mechanism for assigning priority to each IP packet as well as a way to request treatment such as high throughput, high reliability or low latency.

The definition of this field was changed in RFC 2474 . The 6 bit field is now called the DS (Differentiated Services) field and the upper 6 bits contain a value called the Differentiated Services Code Point (DSCP). The remaining two least significant bits are used for Explicit Congestion Notification (ECN).

## *DSCP*

The ToS field is now used by Differentiated Services and is called the Differentiated Services Code Point (DSCP) .

DSCP values range from 0 through 63 that map in groups of 8 to the user priority values.

**TABLE 3** Default DSCP mappings

| DSCP IP precedence | User priority |
|---|---|
| 0-7 | 0 |
| 8-15 | 1 |
| 16-23 | 2 |
| 24-31 | 3 |
| 32-39 | 4 |
| 40-47 | 5 |
| 48-55 | 6 |
| 56-63 | 7 |

# Congestion control

Congestion control covers features that define how the system responds when congestion occurs or active measures taken to prevent the network from entering a congested state.

Sustained, large queue buildups generally indicate congestion in the network and can affect application performance through increased queueing delays and frame loss. Queues can begin filling up for a number of reasons, such as over-subscription of a link or back pressure from a downstream device. When queues begin filling up and all buffering is exhausted, frames are dropped. This has a detrimental effect on application throughput. Congestion control techniques are used to reduce the risk of queue overruns without adversely affecting network throughput.

Congestion control covers features that define how the system responds when congestion occurs or active measures taken to prevent the network from entering a congested state. These features include link level flow control (LLFC), Weighted random early detection (WRED), transient buffer congestion detection.

# Scheduling

Scheduling arbitrates among multiple queues waiting to transmit a frame.

The device supports Strict Priority (SP) scheduling, Weighted fair queue traffic scheduling (WFQ), and mixed SP and WFQ scheduling.

## *Scheduling types*

**TABLE 4** Scheduling comparisons

| Scheduling type | Description |
|---|---|
| SP (Strict priority) | SP handles the scheduling of the packets following a priority-based model where packets are classified and placed into different queues with different priorities. Packets are sent from the head of a given queue for processing only if the queues with higher priorities are empty. |
| WRR (Weighted round robin) | WRR addresses the priority queue problem in which one queue can starve other queues that are not as high a priority. WRR does this by allowing at least one packet to be removed from each queue containing packets in each scheduling turn. This scheme is best used with server queues with different processing capacities. |
| WFQ (Weighted fair queueing) | In WFQ big packets do not get more scheduling time than smaller packets, as the WFQ foci is on bits and not packets as in WRR. |
| DWRR (Deficit weighted round robin) | DWRR is a modified WRR scheduling type that addresses the limitations of WRR. The algorithm handles packets with variable sizes. A maximum packet size number is subtracted from the packet length, and packets that exceed that number are held back until the next scheduling turn |
| Mixed SP and WFQ | With this type of scheduling the top scheduler inputs are SP and the bottom scheduler inputs are WFQ . Usually it is the top three are SP and the bottom five are WFQ. |

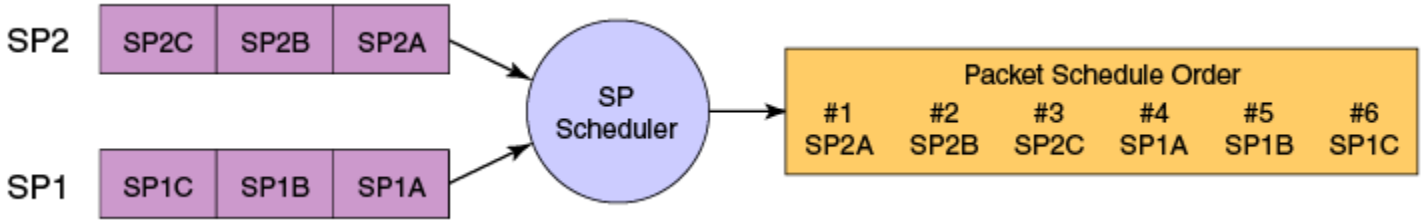
## *QoS strict priority egress traffic scheduling*

Egress traffic scheduling allows you to selectively manage traffic based on the forwarding queue to which it is mapped.

Strict priority scheduling (SP) scheduling is used to facilitate support for latency sensitive traffic. A strict priority scheduler drains all frames queued in the highest-priority queue before continuing on to service lower-priority traffic classes.

The following figure displays the frame scheduling order for an SP scheduler servicing two SP queues. The higher-numbered queue, SP2, has a higher priority.

**FIGURE 4** Strict priority schedule — two queues




The disadvantage of strict priority-based scheduling is that lower-priority traffic can be starved of any access.

The devices classify packets into one of eight internal priorities. For each egress port, there are 8 Virtual output queues (VOQ) allocated on each ingress TM core to support 8 priorities. SP queue input values map to traffic classes and range from 0 through 7. These are:

- 0 - No strict priority queue.
- 1 - Traffic Class 7 strict priority queue.
- 2 - Traffic Class 6 through 7 strict priority queues.
- 3 - Traffic Class 5 through 7 strict priority queues.
- 4 - Traffic Class 4 through 7 strict priority queues.
- 5 - Traffic Class 3 through 7 strict priority queues.
- 6 - Traffic Class 2 through 7 strict priority queues.
- 7 - Traffic Class 1 through 7 strict priority queues.

When configuring egress traffic scheduling you use credit request and grant mechanisms to perform QoS. The credit size is 1024B.

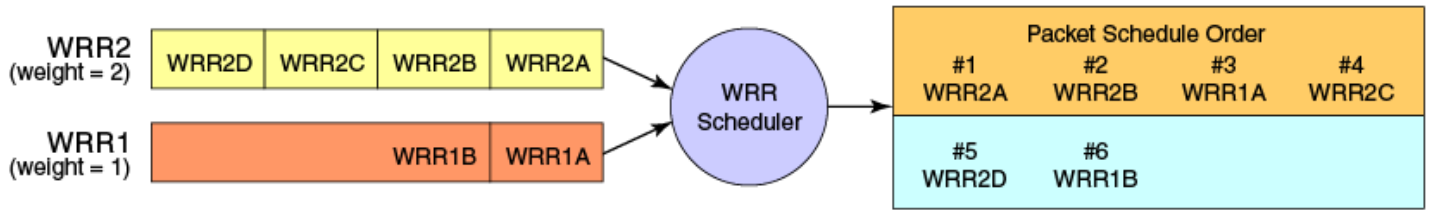
## *Weighted round robin egress traffic scheduling*

In the weighted round robin (WRR) destination-based scheduling enabled scheme, some weight-based bandwidth is allocated to all queues.

WRR scheduling is used to facilitate controlled sharing of the network bandwidth. WRR assigns a weight to each queue; that value is then used to determine the amount of bandwidth allocated to the queue. The round robin aspect of the scheduling allows each queue to be serviced in a set order, sending a limited amount of data before moving onto the next queue and cycling back to the highest-priority queue after the lowest-priority queue is serviced.

The following figure displays the frame scheduling order for a WRR scheduler servicing two WRR queues. The higher-numbered queue is considered higher priority (WRR2), and the weights indicate the network bandwidth should be allocated in a 2:1 ratio between the two queues. In this figure WRR2 receives 66 percent of the bandwidth and WRR1 receives 33 percent. The WRR scheduler tracks the extra bandwidth used and subtracts it from the bandwidth allocation for the next cycle through the queues. In this way, the bandwidth utilization statistically matches the queue weights over longer time periods.

FIGURE 5 WRR schedule — two queues




Deficit Weighted Round Robin (DWRR) is an improved version of WRR. DWRR remembers the excess used when a queue goes over its bandwidth allocation and reduces the queue's bandwidth allocation in the subsequent rounds. This way the actual bandwidth usage is closer to the defined level when compared to WRR.

## *Fair queue egress traffic scheduling*

There are two types of fair queue egress traffic scheduling, weighted fair queue (WFQ) and mixed strict priority (SP) and WFQ.

### Weighted fair queue

With WFQ destination-based scheduling enabled, some weight-based bandwidth is allocated to all queues. With this scheme, the configured weight distribution is guaranteed across all traffic leaving an egress port and an input port is guaranteed allocation in relationship to the configured weight distribution.

You can specify weighted for each VOQ if in WFQ mode.

### Mixed SP and WFQ egress traffic scheduling

This scheme provides a mixture of SP for the three highest priority queues and WFQ for the five remaining priority queues.

## *Multicast queue scheduling*

A fixed mapping from multicast traffic class to equivalent unicast traffic class is applied to select the queue scheduling behavior.

The multicast traffic classes are numbered from 0 to 7; higher numbered traffic classes are considered higher priority. The Multicast traffic class equivalence mapping table below presents the multicast traffic class with the equivalence mapping applied.

Once the multicast traffic class equivalence mapping has been applied, then scheduling and any scheduler configuration are inherited from the equivalent unicast traffic class. Refer to the table below for details on exact mapping equivalencies.

TABLE 5 Multicast traffic class equivalence mapping

| Multicast traffic class | Equivalent unicast traffic class |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |

Unicast ingress and egress queueing utilizes a hybrid scheduler that simultaneously supports SP+WRR service and multiple physical queues with the same service level. Multicast adds additional multicast expansion queues. Because multicast traffic classes are equivalent to unicast service levels, they are treated exactly as their equivalent unicast service policies.

## *Control protocol packet prioritization*

Certain control packets are handled with certain priorities by default and hence those priorities cannot be lowered with any of the QoS configuration commands.

The following table lists the protocol packets that are internally and automatically prioritized for IPv4, Layer 2, and IPv6.

**TABLE 6** Default prioritized protocol table

| Protocol Packets |
|---|
| ARP |
| BFD (Bidirectional Forwarding Detection) |
| BGP / BGP in 6to4 |
| BGP / BGP over GRE |
| BOOTP/DHCP |
| GARP |
| IGMP |
| IPv4 Router Alert |
| IPv4/L2 |
| IPv6 |
| LACP |
| ND6 / ND6 in 6to4 |
| OSPF / OSPF in 6to4 |
| OSPF / OSPF over GRE |
| RIP |
| RIPNG |
| STP/RSTP/BPDU |
| VRRP |
| VRRPE |

### Enhanced control packet prioritization

The Traffic Manager (TM) allows prioritization and scheduling of packets destined for the CPU to guarantee optimal control packet processing and to reduce protocol flapping.

The TM achieves physical separation of CPU-bound data and control packets. The hierarchical structure supports four sets of eight priority queues. The four sets are as follows:

- Protocol set - Protocol packets that are prioritized by the network processor.
- Management set - Packets destined for the router; for example, Ping and Telnet.
- Flow set - Flow-driven packets to the CPU; for example, Unknown DA, DPA, Layer 2 broadcast and multicast, Multicast packets.
- Snoop set - CPU copy packets; for example, Regular Layer 2 SA learning, sFlow, ACL Logging, RPF Logging. The **rl-cpu-copy** command defines the rate shaping value for the snoop queues.

Protocol packets are mapped into different queues. Each queue has it own rate limit and is assigned one of the three priority groups. Each priority group is scheduled based on the different weight. The application queues without buffering the packets does not compete for the scheduling resource and allows the following:

- The time critical packets receive more scheduling to meet the timing requirement.
- The time insensitive packets are guaranteed the minimum bandwidth to avoid the resource starvation.
- No application can flood the CPU path and block other applications that have been processed.

The following table lists the group priority of the applications.

**TABLE 7** Protocol packet priority groups

| Group priority | Protocol |
|---|---|
| High | BFD, LACP, UDLD, LLDP, CTP, 802.1X |
| Medium | OSPF, BGP, VRRP, PIM, IGMP, STP, MCAST |
| Low | Management, ARP, SFLOW, DHCP, others |

# Configuring flow-based QoS

Follow these high level steps to configure flow-based QoS.

1. Configure a class map to classify traffic according to the traffic properties required for your flow-based QoS needs.
2. Configure a policy map and associate it to the class map.

   NOTE
   Policy maps can be bound in both the ingress and egress directions.

3. Add the QoS action to be applied on the type of flow determined by the class map.
4. Bind the policy map to a specific interface.

## *Match access-group — class map policing*

Access groups are used for Layer 2 and Layer 3 ACL-based ingress rate limit and denial of service (DoS) mitigation.

ACL-based rate limiting is built on top of ACL and policer features. It rate limits the following traffic:

- Layer 2 traffic that matches the permit conditions in Layer 2 access lists.
- Layer 3 traffic that matches the permit conditions specified in an IPv4 access list.

The ACL-based policer feature controls the amount of bandwidth consumed by an individual flow or aggregate of inbound flows by limiting the traffic rate on an individual port according to criteria defined by the **match access-group** class map. This ACL-based rate limiting feature can serve as a hardware solution to prevent DoS attacks.

### Match access-group - class map policing rules and limitations

Consider these rules and limitations when you are configuring **match access-group** class map policing:

- You can configure:
  - 1,024 policy maps
  - 6,144 ACL Content Addressable Memory (CAM) entries for use with rate limiting
  - 2,048 ACLs with rate limiting for each user

NOTE
The number of Ternary Content Addressable Memory (TCAM) entries for use with rate limiting and ingress policers are dependent on the hardware TCAM profile that is used.

- For protection against:
  - PING attacks
  - TCP Reset attacks
  - TCP SYN attacks
  - UDP attack
- Layer 2 MAC and Layer 3 IPv4 ACL-based rate limiting are supported.
- ACL-based rate limiting is applicable only to ingress traffic.
- There is one policer per ACL, it applies to all the rules for that ACL
- Control protocols are rate-limited if they match the configured ACL clause.
- When a **match access-group** class map rate limit is applied to a LAG logical port, and all LAG ports belong to the same tower, then MAX CIR value is the interface speed × number of physical ports. For example: if 0/1, 0/2 are LAG member ports, then MAX CIR will be 2 × 10Gbps.
- When a **match access-group** class map rate limit is applied to LAG logical port, MAX rate on that port is the number of the tower in that LAG × CIR. For example: if 0/1, 0/2 , 0/4, 0/5 are LAG member ports, then MAX rate is 3 × CIR.

## *Policy maps*

Policy maps allow you to set a policy in a single location that affects multiple ports and to make changes to that policy.

The policy map configuration includes a set of class maps and QoS parameters.

A policy map allows you to specify policers in a single location that can be applied to multiple ports and to make changes to that policy.

The traffic policing policies must be defined using a policy map. One policy map can be specified per service policy. You can configure up to 1024 policy maps per system.

### Policy map configuration rules

Follow these rules when configuring traffic policing:

- A policer map (policy map or class map) name must be unique among all maps of that type.
- A policer name must begin with an alphabetic character ( a-z or A-Z) . An underscore, hyphen, and numeric values 0-9 can be used in the body of the name but not as the first character.
- You can configure a maximum of 1024 policy maps.
- ACL-based class maps and default class maps can be used in a single policy map.
- ACL-based, storm-control, and port-based rate limiting can coexist on the same interface. ACL-based rate limiting has the highest priority, followed by storm-control rate limiting, and then port-based rate limiting.
- For an ingress or egress service policy, one default class map can be specified per policy map.
- Broadcast, unknown unicast, and multicast (BUM) policies are counted separately.
- You cannot delete a policy map if is referenced in an active service policy (applied on an interface).

### QoS shaping rate

You can specify the shaping rate per port attached to the policy map to smooth the traffic that egresses an interface. This configuration is allowed only for egress traffic.

## Ingress QoS mutation

The QoS operation on ingress traffic involves reception and processing of packets based upon priority information contained within the packet.

When packets are processed through the device, there are several opportunities to influence the processing by configuration as described in the steps below. The processes performed to map packet priority to internal priority can be described as following:

- Collect priority information from various portions of the packet header:
  - If a packet's EtherType matches 8100, derive a priority value by decoding the PCP value.
  - For IPv4 or IPv6 packets, derive priority value by decoding the DSCP bits.
  - For untagged Layer 2 packet, derive traffic class using the port's default value.
  - The derived values for PCP and DSCP are mapped using either a default map or a configured ingress decode policy map.
  - To assist the device in the decoding process described, decode map tables are defined.
- The priority values is obtained in descending order of priority, as follows:
  1. If tag exists and packet is switched, by decoding the PCP value from the tag.
  2. For IPv4 or IPv6 packets, and when the packet is routed, by decoding the DSCP field from the IP header.
  3. Physical port default value.

# Configuring QoS

QoS configuration involves multiple procedures for QoS processing as described in the following sections.

## Configuring QoS for control traffic

Configure the Traffic Manager (TM) CPU port shaper rate (all towers) to the line card (LC) CPU.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Set the TM CPU port shaper on slot 1 to 4000 Kbps with a burst size of 1KB.

```
device(config)# qos cpu slot 1 port shaper rate 4000 burst 1
```

3. Return to privileged exec mode,

```
device(config)# exit
```

4. Verify the configuration.

```
device# show run qos cpu
qos cpu slot 1 port shaper rate 4000 burst 1
```

5. Save the configuration.

```
device# copy running-config startup-config
```

### *QoS for control traffic configuration example*

```
device# configure terminal
device(config)# qos cpu slot 1 port shaper rate 4000 burst 1
device(config)# exit
device# show run qos cpu
device# copy running-config startup-config
```

# Configuring CoS-to-traffic class mappings

Follow these tasks to configure QoS CoS-to-traffic class mappings.

### *Configuring a CoS-to-traffic class mutation map*

Follow these steps to configure a QoS CoS-to-traffic class mutation map.

The ingress 802.1p priority values can be used to classify traffic to a specific traffic class (priority queue) and drop precedence. This can be done by configuring a cos-to-traffic class map.

The drop precedence is optional in all below commands.

If a CoS-to-traffic class mutation map is not defined, the default CoS is used as value of the traffic class, and 0 is used for the drop precedence.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create and name a CoS-to-traffic class map.

   ```
   device(config)# qos map cos-traffic-class cosTCMap
   ```

3. Map ingress CoS value to the CoS-to-traffic class map traffic-class and drop precedence values.

   ```
   device(config-cos-traffic-class-cosTCMap)# map cos 4 to traffic-class 3 drop-precedence 0
   device(config-cos-traffic-class-cosTCMap)# map cos 5 to traffic-class 5 drop-precedence 1
   device(config-cos-traffic-class-cosTCMap)# map cos 6 to traffic-class 6 drop-precedence 0
   device(config-cos-traffic-class-cosTCMap)# map cos 7 to traffic-class 6 drop-precedence 1
   ```

4. Return to privileged EXEC mode.

   ```
   device(config-cos-traffic-class-cosTCMap)# end
   ```

5. Verify the configuration.

   ```
   device# show qos maps cos-traffic-class

   Cos-to-Traffic Class map 'cosTCMap'
            In-Cos  : 0  1  2  3  4  5  6  7
      ---------------------------------------
      TrafficClass  : 0  1  2  3  3  6  6  6
      DropPrecedence: 0  0  0  0  0  1  0  1

      Enabled on the following interfaces:
   ```

### QoS CoS-to-traffic class map configuration example

```
device# configure terminal
device(config)# qos map cos-traffic-class cosTCMap
device(config-cos-traffic-class-cosTCMap)# map cos 4 to traffic-class 3 drop-precedence 0
device(config-cos-traffic-class-cosTCMap)# map cos 5 to traffic-class 5 drop-precedence 1
device(config-cos-traffic-class-cosTCMap)# map cos 6 to traffic-class 6 drop-precedence 0
device(config-cos-traffic-class-cosTCMap)# map cos 7 to traffic-class 6 drop-precedence 1
device(config-cos-traffic-class-cosTCMap)# end
device# show qos maps cos-traffic-class
```

# Configuring DSCP mappings

Follow the tasks below to configure DSCP mappings.

## *Configuring a DSCP-to-DSCP mutation map*

Follow these steps to create a DSCP mutation map and remap the incoming DSCP value of the ingress packet to egress DSCP values.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create the DSCP-to-DSCP mutation map by specifying a map name, which places the system in DSCP mutation mode so that you can map to traffic classes.

   ```
   device(config)# qos map dscp-mutation dscpMap
   ```

3. Map ingress DSCP values to egress DSCP values.
   a) Set the DSCP input value 24 to output as DSCP value 50.

      ```
      device(dscp-mutation-dscpMap)# map dscp 24 to dscp 50
      ```

   b) Set the DSCP input value 33 to output as DSCP value 35.

      ```
      device(dscp-mutation-dscpMap)# map dscp 33 to dscp 35
      ```

   c) Set the DSCP input value 53 to output as DSCP value 61.

      ```
      device(dscp-mutation-dscpMap)# map dscp 53 to dscp 61
      ```

   d) Set the DSCP input value 60 to output as DSCP value 40.

      ```
      device(dscp-mutation-dscpMap)# map dscp 60 to dscp 40
      ```

4. Return to privileged EXEC mode.

   ```
   device(dscp-mutation-dscpMap# end
   ```

5. Verify the configuration.

```
device# show qos map dscp-mutation dscpMap
Dscp-to-Dscp Mutation map 'dscpMap' (dscp= d1d2)
d1 :  d2 0  1  2  3  4  5  6  7  8  9
---------------------------------------
0 :    00 01 02 03 04 05 06 07 08 09
1 :    10 11 12 13 14 15 16 17 18 19
2 :    20 21 22 23 24 25 26 27 28 29
3 :    30 31 32 33 34 35 36 37 38 39
4 :    40 41 42 43 44 45 46 47 48 49
5 :    50 51 52 53 54 55 56 57 58 59
6 :    40 61 62 63

   Enabled on the following interfaces:
```

6. Save the `running-config` file to the `startup-config` file

```
device# copy running-config startup-config
```

### QoS DSCP-to-DSCP mutation map configuration example

```
device# configure terminal
device(config)# qos map dscp-mutation dscpMap
device(dscp-mutation-dscpMap)# map dscp 60 to dscp 40
device(dscp-mutation-dscpMap)# map dscp 24 to dscp 50
device(dscp-mutation-dscpMap)# map dscp 33 to dscp 35
device(dscp-mutation-dscpMap)# map dscp 53 to dscp 61
device(dscp-mutation-dscpMap# end
device# show qos map dscp-mutation dscpMap
device# copy running-config startup-config
```

## *Applying a DSCP-to-DSCP mutation map to an egress interface*

Follow these steps to apply a QoS DSCP-to-DSCP mutation map to an egress interface.

This feature allows you to take the normalized QoS in-DSCP value of the egressing IP packet, and bind it to an egress interface.

A QoS DSCP-to-DSCP mutation map has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 0/5
```

3. Enable the DSCP-to-DSCP mutation map on the interface.

```
device(conf-if-eth-0/5)# qos dscp-mutation dscpMap
```

4. Return to privileged exec mode.

```
device(conf-if-eth-0/5)# end
```

5. Verify the configuration.

```
device# show qos maps dscp-mutation

Dscp-to-Dscp Mutation map 'dscpMap+' (dscp= d1d2)
d1 :  d2 0  1  2  3  4  5  6  7  8  9
---------------------------------------
0 :     00 01 02 03 04 05 06 07 08 09
1 :     10 11 12 13 14 15 16 17 18 19
2 :     20 21 22 23 24 25 26 27 28 29
3 :     30 31 32 33 34 35 36 37 38 39
4 :     40 41 42 43 44 45 46 47 48 49
5 :     50 51 52 53 54 55 56 57 58 59
6 :     40 61 62 63

   Enabled on the following interfaces: Eth 0/5
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Apply a QoS DSCP-to-DSCP mutation map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/5
device(conf-if-eth-0/5)# qos dscp-mutation dscpMap
device(conf-if-eth-0/5)# end
device# show qos maps dscp-mutation
device# copy running-config startup-config
```

# Configuring DSCP-to-CoS mappings

Follow these tasks to configure DSCP-to-CoS mappings.

## *Configuring a DSCP-to-CoS mutation map*

Follow these steps to use the DSCP value of ingress packets to remap the egress 802.1p CoS priority values.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a named QoS DSCP-to-CoS mutation map.

```
device(config)# qos map dscp-cos dscpCosMap
```

   This also places the system in dscp-cos map mode so that you can map DSCP values to CoS values.

3. Map ingress DSCP values to egress CoS values.
   a) DSCP value 23 is set to output as CoS priority 4.

```
device(dscp-cos-dscpCosMap)# map dscp 23 to cos 4
```

   b) DSCP values 43 are set to output as CoS priority 4.

```
device(dscp-cos-dscpCosMap)# map dscp 43 to cos 5
```

   c) DSCP value 53 is set to output as CoS priority 6.

```
device(dscp-cos-dscpCosMap)# map dscp 53 to cos 6
```

   d) DSCP value 63 is set to output as CoS priority 7.

```
device(dscp-cos-dscpCosMap)# map dscp 63 to cos 7
```

4. Return to privileged EXEC mode.

```
device(dscp-cos-dscpCosMap)# end
```

5. Verify the configuration.

```
device#  show qos maps dscp-cos

Dscp-to-CoS map 'dscpCosMap' (dscp= d1d2)
d1 :  d2 0  1  2  3  4  5  6  7  8  9
---------------------------------------
0 :     00 00 00 00 00 00 00 00 01 01
1 :     01 01 01 01 01 01 02 02 02 02
2 :     02 02 02 04 03 03 03 03 03 03
3 :     03 03 04 04 04 04 04 04 04 04
4 :     05 05 05 05 05 05 05 05 06 06
5 :     06 06 06 06 06 06 07 07 07 07
6 :     07 07 07 07

   Enabled on the following interfaces:
```

6. Save the `running-config` file to the `startup-config` file

```
device# copy running-config startup-config
```

## QoS DSCP-to-CoS mutation map configuration example

```
device# configure terminal
device(config)# qos map dscp-cos dscpCosMap
device(dscp-cos-dscpCos)# map dscp 43 to cos 4
device(dscp-cos-dscpCos)# map dscp 63 to cos 6
device(dscp-cos-dscpCos)# map dscp 53 to cos 5
device(dscp-cos-dscpCos)# map dscp 23 to cos 2
device(dscp-cos-dscpCosMap)# end
device# show qos maps dscp-cos
device# copy running-config startup-config
```

### *Applying a DSCP-to-CoS mutation map to an interface*

Follow these steps to map an ingress DSCP value to an outgoing 802.1p value. This can be done by configuring a DSCP-to-CoS mutation map on the ingress interface.

A QoS DSCP-to-CoS mutation map has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 0/5
```

3. Enable the DSCP mutation map on the interface.

```
device(conf-if-eth-0/5)# qos dscp-cos dscpCosMap
```

   The dscp-mutation and dscp-traffic-class maps must also be applied on the ingress interface to be effective.

4. Return to privileged EXEC mode.

```
device(conf-if-eth-0/5)# end
```

5. Verify the configuration.

```
device#  show qos maps dscp-cos dscpCosMap
 [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]


DSCP-to-CoS Map:  dscpCosMap (dscp= d1d2)
d1 :  d2 0  1  2  3  4  5  6  7  8  9
-----------------------------------
0 :     00 00 00 00 00 00 00 01 01
1 :     01 01 01 01 01 01 02 02 02 02
2 :     02 02 02 04 03 03 03 03 03 03
3 :     03 03 04 04 04 04 04 04 04 04
4 :     05 05 05 05 05 05 05 05 06 06
5 :     06 06 06 06 06 06 07 07 07 07
6 :     07 07 07 07


   Enabled on the following interfaces: Eth 0/5
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Applying a QoS DSCP-to-CoS mutation map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/5
device(conf-if-eth-0/5)# qos dscp-cos dscpCosMap
device(conf-if-eth-0/5)# end
device# show qos maps dscp-cos dscpCosMap
device# copy running-config startup-config
```

## Configuring DSCP-to-traffic class mappings

Follow these tasks to configure DSCP-to-traffic class mappings.

## *Configuring a DSCP-to-traffic class mutation map*

Follow these steps to configure a QoS DSCP-to-traffic class map.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create and name a QoS DSCP-to-traffic class map.

```
device(config)# qos map dscp-traffic-class dscpTcMap
```

3. Define the QoS DSCP-to-traffic class values.

```
device(config-dscp-traffic-class-dscpTcMap)# map dscp 10 to traffic-class 3
device(config-dscp-traffic-class-dscpTcMap)# map dscp 40 to traffic-class 4
device(config-dscp-traffic-class-dscpTcMap)# map dscp 45 to traffic-class 5
device(config-dscp-traffic-class-dscpTcMap)# map dscp 52 to traffic-class 3
```

   The default value is used for those DSCP that are not explicitly defined.

4. Return to privileged exec mode.

```
device(config-dscp-traffic-class-dscpTcMap)# end
```

5. Verify the configuration.

```
device# show qos maps dscp-traffic-class

Dscp-to-Traffic-Class map 'dscpTcMap'
{x/y: traffic-class = x, drop-precedence = y & dscp = d1d2}
d1 :  d2  0   1   2   3   4   5   6   7   8   9
------------------------------------------------
0 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 4/2 1/0
1 :     1/0 1/0 1/0 1/0 1/0 1/0 2/0 2/0 2/0 2/0
2 :     2/0 2/0 2/0 2/0 3/0 3/0 3/0 3/0 3/0 3/0
3 :     3/0 3/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0
4 :     5/0 5/0 5/0 5/0 5/0 5/0 5/0 5/0 6/0 6/0
5 :     6/0 6/0 6/0 6/0 6/0 6/0 7/0 7/0 7/0 7/0
6 :     7/0 7/0 7/0 7/0

   Enabled on the following interfaces:
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### QoS DSCP-to-traffic class and drop precedence map configuration example

```
device# configure terminal
device(config)# qos map dscp-traffic-class dscpTcMap
device(config-dscp-traffic-class-dscpTcMap)# map dscp 10 to traffic-class 3
device(config-dscp-traffic-class-dscpTcMap)# map dscp 40 to traffic-class 4
device(config-dscp-traffic-class-dscpTcMap)# map dscp 45 to traffic-class 5
device(config-dscp-traffic-class-dscpTcMap)# map dscp 52 to traffic-class 3
device(config-dscp-traffic-class-dscpTcMap)# end
device# show qos maps dscp-traffic-class
device# copy running-config startup-config
```

## *Applying a DSCP-to-traffic class mutation map to an interface*

Follow these steps to apply a QoS DSCP-to-traffic class map to an ingress interface.

A QoS DSCP-to-traffic class map has been configured

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 0/2
```

3. Activate the QoS DSCP-to-traffic class mutation map on the interface.

```
device(conf-if-eth-0/2)# qos dscp-traffic-class dscpTcMap
```

4. Return to privileged EXEC mode.

```
device(conf-if-eth-0/2)# end
```

5. Verify the configuration

```
device# show qos maps dscp-traffic-class dscpTcMap
 [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]

  DSCP-to-TC Map: dscpTcMap (x/y: TC = x, DP = y, DSCP = d1d2)
      d1 :  d2  0   1   2   3   4   5   6   7   8   9
      ------------------------------------------------
       0 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 1/0 1/0
       1 :     3/0 1/0 1/0 1/0 1/0 1/0 2/0 2/0 2/0 2/0
       2 :     2/0 2/0 2/0 2/0 3/0 3/0 3/0 3/0 3/0 3/0
       3 :     3/0 3/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0
       4 :     4/0 5/0 5/0 5/0 5/0 5/0 5/0 5/0 6/0 6/0
       5 :     6/0 6/0 3/0 6/0 6/0 6/0 7/0 7/0 7/0 7/0
       6 :     7/0 7/0 7/0 7/0

      Enabled on the following interfaces:  Eth 0/2
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Applying a QoS DSCP-to-traffic class map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# qos dscp-traffic-class dscpTcMap
device(conf-if-eth-0/2)# end
device# show qos maps dscp-traffic-class dscpTcMap
device# copy running-config startup-config
```

## *Configuring a DSCP-to-traffic class and drop precedence mutation map*

Follow these steps to configure a QoS DSCP to traffic class and drop precedence mutation map.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create and name a QoS DSCP-to-traffic class and drop precedence mutation map.

```
device(config)# qos map dscp-traffic-class dscpTcDpMap
```

3. Define the QoS DSCP-to-traffic class and drop precedence values.

```
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 10 to traffic-class 3 drop-precedence 1
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 40 to traffic-class 4 drop-precedence 1
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 45 to traffic-class 5 drop-precedence 0
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 52 to traffic-class 3 drop-precedence 1
```

If a default DSCP-to-traffic class map is not defined, then the IP precedence bits (first 3 bits) of the DSCP are used as the traffic class for the map, and drop precedence is given a value of 0.

4. Return to privileged exec mode.

```
device(config-dscp-traffic-class-dscpTcDpMap)# end
```

5. Verify the configuration.

```
device# show qos maps dscp-traffic-class
DSCP-to-TC Map: a1 (x/y: TC = x, DP = y, DSCP = d1d2)
    d1 :  d2  0   1   2   3   4   5   6   7   8   9
    ----------------------------------------------
     0 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 1/0 1/0
     1 :     1/0 1/0 1/0 1/0 1/0 1/0 2/0 2/0 2/0 2/0
     2 :     2/0 2/0 2/0 2/0 3/0 3/0 3/0 3/0 3/0 3/0
     3 :     3/0 3/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0
     4 :     5/0 5/0 5/0 5/0 5/0 5/0 5/0 5/0 6/0 6/0
     5 :     6/0 6/0 6/0 6/0 6/0 6/0 7/0 7/0 7/0 7/0
     6 :     7/0 7/0 7/0 7/0

    Enabled on the following interfaces: >>> map a1 is not applied on any interface.
```

CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence.

6. Save the `running-config` file to the `startup-config` file

```
device# copy running-config startup-config
```

### QoS DSCP to traffic class and drop precedence mutation map configuration example

```
device# configure terminal
device(config)# qos map dscp-traffic-class dscpTcDpMap
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 10 to traffic-class 3 drop-precedence 1
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 40 to traffic-class 4 drop-precedence 1
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 45 to traffic-class 5 drop-precedence 0
device(config-dscp-traffic-class-dscpTcMap)# map dscp-value 52 to traffic-class 3 drop-precedence 1
device(dscp-traffic-class-dscpTcDpMap)# end
device# show qos maps dscp-traffic-class dscpTcDpMap
device# copy running-config startup-config
```

## *Applying a DSCP-to-traffic class and drop precedence mutation map to an interface*

Follow these steps to apply a DSCP-to-traffic class and drop precedence mutation map to an ingress interface.

A QoS DSCP-to-traffic class and drop precedence mutation map has been configured

The ingress DSCP value can be used to classify traffic in to a specific traffic class and drop precedence by applying a DSCP-to-traffic class and drop precedence mutation map on the ingress interface.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 0/2
```

3. Enable the QoS DSCP-to-traffic class and drop precedence mutation map on the interface.

```
device(conf-if-eth-0/2)# qos dscp-traffic-class dscpTcDpMap
```

4. Return to privileged exec mode.

```
device(conf-if-eth-0/2)# end
```

5. Verify the configuration

```
device# show qos maps dscp-traffic-class dscpTcDpMap
DSCP-to-TC Map: dscpTc (x/y: TC = x, DP = y, DSCP = d1d2)
    d1 :  d2  0   1   2   3   4   5   6   7   8   9
    ------------------------------------------------
     0 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 1/0 1/0
     1 :     1/0 1/0 1/0 1/0 1/0 1/0 2/0 2/0 2/0 2/0
     2 :     2/0 2/0 2/0 2/0 3/0 3/0 3/0 3/0 3/0 3/0
     3 :     3/0 3/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0 4/0
     4 :     5/0 5/0 5/0 5/0 5/0 5/0 5/0 5/0 6/0 6/0
     5 :     6/0 6/0 6/0 6/0 6/0 6/0 7/0 7/0 7/0 7/0
     6 :     7/0 7/0 7/0 7/0

    Enabled on the following interfaces: Eth 0/2
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Apply a QoS DSCP-to-traffic class map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# qos dscp-traffic-class dscpTcDpMap
device(conf-if-eth-0/2)# end
device# show qos maps dscp-traffic-class dscpTcDpMap
device# copy running-config startup-config
```

# Configuring traffic class-to-CoS mappings

## *Configuring a traffic class-to-CoS mutation map*

Follow these steps to configure QoS traffic class-to-CoS mutation map.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Configure the QoS traffic class-to-CoS mutation map.

```
device(config)# qos map traffic-class-cos CoSMap 1 1 2 3 4 4 4 3
```

   If the QoS CoS mutation map is not configured, then the default CoS mutation map is used with 1:1 mapping for the traffic class-to-PCP values.

3. Return to privileged exec mode.

```
device(config)# exit
```

4. Verify the configuration.

```
device# show qos maps traffic-class-cos

Traffic Class-to-Cos Mutation map 'CoSMap'
   TrafficClass: 0  1  2  3  4  5  6  7
   ------------------------------------
        Out-Cos: 1  1  2  3  4  4  4  3

   Enabled on the following interfaces:
```

5. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Traffic class-to-CoS mutation map configuration example

```
device# configure terminal
device(config)# qos map traffic-class-cos CoSMap 1 1 2 3 4 4 4 3
device(config)# exit
device# show qos maps traffic-class-cos
device# copy running-config startup-config
```

## *Applying a traffic class-to-CoS mutation map to an egress interface*

Follow these steps to apply a QoS traffic class-to-CoS mutation map to an egress interface.

A QoS traffic class-to-CoS mutation map has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 2/2
```

3. Apply the configured QoS traffic class-to-CoS map to the interface.

```
device(conf-if-eth-2/2)# qos traffic-class-cos tcCoSMap
```

4. Return to privileged exec mode.

```
device(conf-if-eth-2/2)# end
```

5. Verify the configuration.

```
device# show qos maps traffic-class-cos tcCoSMap

  [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]
  TC-to-CoS Map: tcCoSMap
              In-TC: 0  1  2  3  4  5  6  7
      ------------------------------------
      Out-CoS(DP=0): 0  1  2  3  4  2  6  7
      Out-CoS(DP=1): 0  1  2  3  4  5  6  7
      Out-CoS(DP=2): 0  1  1  3  4  2  6  7
      Out-CoS(DP=3): 0  1  2  3  4  5  6  7

      Enabled on the following interfaces:
      Eth 2/2
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Apply a QoS traffic class-to-CoS mutation map to an egress interface configuration example

```
device# configure terminal
device(config)# interface ethernet 2/2
device(conf-if-eth-2/2)# qos traffic-class-cos tcCoSMap
device(conf-if-eth-2/2)# end
device# show qos maps traffic-class-cos tcCoSMap
device# copy running-config startup-config
```

# Configuring congestion control

For information on congestion control, refer to Congestion control on page 51.

## *Configuring WRED*

WRED is configurable on the ingress side to control when to perform a tail drop or Random Early Drop (RED). Follow these steps to configure WRED.

1. Enter configuration mode.

```
device# configure terminal
```

2. Create a WRED profile identified as profile 1, set the thresholds, and set the drop probability.

```
device(config)# qos red-profile 1 min-threshold 30 max-threshold 60 drop-probability 44
```

3. Access the interface to which you will configure the WRED profile.

```
device(config)# interface Ethernet 0/1
```

4. Set the thresholds, and set the drop probability for WRED profile 1.

```
device(config-if-eth-0/1)# qos random-detect traffic-class 0 red-profile-id 1
```

5. Return to privileged EXEC mode.

```
device(config-if-eth-0/1)# end
```

6. Verify the WRED configuration.

```
device# show qos red profiles 1

Red Profile 1
        Minimum Threshold: 30
        Maximum Threshold: 60
        Drop Probability: 44


Applied on the following interfaces:
Eth 0/1 Traffic-class: 2
```

7. Save the configuration.

```
device# copy running-config startup-config
```

### WRED configuration example

```
device# configure terminal
device(config)# qos red-profile 1 min-threshold 30 max-threshold 60 drop-probability 44
device(config)# interface Ethernet 0/1
device(config-if-eth-0/1)# interface Ethernet 0/1
device(config-if-eth-0/1)# qos random-detect traffic-class 0 red-profile-id 1
device(config-if-eth-0/1)# end
device# show qos red profiles 1
device# copy running-config startup-config
```

### Displaying WRED statistics for an interface

The following example shows the displaying of WRED statistics for the interface.

```
device# show qos red statistics interface Eth 0/1
Statistics for interface: Eth 0/1
     Traffic-class: 2, ProfileId: 20
     Packets Dropped: Red: 0, Yellow: 0, Green: 0, Queue Drops: 0
     Bytes Dropped: Red: 0, Yellow: 0, Green: 0, Queue Drops: 0

Traffic-class: 3, ProfileId: 10
     Packets Dropped: Red: 0, Yellow: 0, Green: 0, Queue Drops: 0
     Bytes Dropped: Red: 0, Yellow: 0, Green: 0, Queue Drops: 0
```

## *Configuring link level flow control*

Link level flow control (LLFC) allows a congested receiver to communicate a PAUSE frame to a transmitter to stop data transmission until the congestion is cleared.

Before configuring LLFC on an interface or a port channel, Extreme recommends that you stop the traffic on the interface.

LLFC can be configured only at the interface level.

Perform the following steps to configure LLFC.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 0/18
```

3. Enable link level flow control in the receive direction for the port.

```
device(conf-eth-0/18)# qos flowcontrol tx off rx on
```

4. Return to privileged EXEC mode.

```
device(conf-eth-0/18)# end
```

5. Verify the configuration.

```
device# show qos flowcontrol interface ethernet 0/18
Interface Ethernet 0/18
  Mode 802.3x
     TX      RX         TX Output Paused
  Admin   Admin     Frames  512 BitTimes
  ------------------------------------
     Off     On
```

6. Save the configuration.

```
device# copy running-config startup-config
```

## LLFC configuration example

```
device# configure terminal
device(config)# interface ethernet 0/18
device(conf-eth-0/18)# qos flowcontrol tx off rx on
device(conf-eth-0/18)# end
device# show qos flowcontrol interface ethernet 0/18
device# copy running-config startup-config
```

### Displaying flow control information and clearing its statistics

You can display flow control information for all interfaces, a specific Ethernet interface, or a port channel. This information includes the flow control mode, generation (Tx) and reception (Rx) status, and Tx and Rx PAUSE frame counts.

The following example shows how to display the flow control information for all interfaces.

```
device# show qos flowcontrol interface all
Interface Ethernet 0/1
  Mode Off
Interface Ethernet 0/2
  Mode Off
Interface Ethernet 0/3
  Mode Off
Interface Ethernet 0/4
  Mode Off
...
  Mode 802.3x
     TX    RX         TX Output Paused    RX
  Admin Admin     Frames  512 BitTimes     Frames
  ------------------------------------------------
     Off   On          0                        0
```

The following example displays the flow control information on a specific interface.

```
device# show qos flowcontrol interface ethernet 0/18
Interface Ethernet 0/18
  Mode 802.3x
     TX    RX         TX Output Paused    RX
  Admin Admin     Frames  512 BitTimes     Frames
  ------------------------------------------------
     Off   On          0                        0
```

You can also clear the flow control statistics for all interfaces, a specific Ethernet interface, or a port channel. The following example clears the statistics for all interfaces.

```
device# clear qos flowcontrol interface all
```

The following example clears the flow control statistics on a specific interface.

```
device# clear qos flowcontrol interface ethernet 0/18
```

## *Configuring TM deleted or discarded packet, or VOQ discarded packet monitoring*

By default, the monitoring of the deleted or discarded packets is disabled. You can enable the monitoring of all TM deleted or discarded packets on the SLX device, or all VOQ discarded packets.

Perform the following steps to enable the monitoring of these packets.

1. In privileged EXEC mode, access global configuration mode.

   ```
   device# configure terminal
   ```

2. Access sys-mon TM configuration mode.

   ```
   device(config)# system-monitor TM
   ```

3. Configure the threshold interval to enable the monitoring of all TM deleted packets on the device.

   ```
   device(sys-mon tm)# delete-packets threshold 10
   ```

   This step configures the threshold of 10 TM deleted packets. A threshold of 0 disables the monitoring of the packets.

4. Optionally, configure the logging interval to monitor all TM deleted packets.

```
device(sys-mon tm)# delete-packets logging-interval 100
```

   This step configures the logging interval of 100 minutes. By default, the interval is 60 minutes.

5. Configure the threshold interval to enable the monitoring of all TM discarded packets.

```
device(sys-mon tm)# discard-packets threshold 15
```

   This step configures the threshold of 15 TM device discarded packets. A threshold of 0 disables the monitoring of the packets.

6. Optionally, configure the logging interval to monitor all TM discarded packets.

```
device(sys-mon tm)# discard-packets logging-interval 100
```

   This step configures the logging interval of 100 minutes. By default, the interval is 60 minutes.

7. Configure the threshold interval to enable the monitoring of all VOQ discarded packets.

```
device(sys-mon tm)# discard-voq-packets threshold 10
```

   This step configures the threshold of 10 VOQ discarded packets. A threshold of 0 disables the monitoring of the packets.

8. Optionally, configure the logging interval to monitor the VOQ discarded packets.

```
device(sys-mon tm)# discard-voq-packets logging-interval 100
```

   This step configures the logging interval of 100 minutes. By default, the interval is 60 minutes.

The following example shows the configuration in the previous steps.

```
device# configure terminal
device(config)# system-monitor TM
device(sys-mon tm)# delete-packets logging-interval 100
device(sys-mon tm)# delete-packets threshold 10
device(sys-mon tm)# discard-packets threshold 15
device(sys-mon tm)# discard-packets logging-interval 100
device(sys-mon tm)# discard-voq-packets threshold 10
device(sys-mon tm)# discard-voq-packets logging-interval 100
```

## Displaying the TM packet monitoring configurations

Use the **show system monitor tm** command to display the configuration for the deleted and discarded TM-device packets or the discarded VOQ discarded packets.

The following example displays the monitoring configuration for the TM-device deleted packets.

```
device# show system monitor tm delete-packet
Delete packet count monitoring configuration:
Logging-interval :  60 minutes
Threshold        :  10 packets
```

The following example displays the monitoring configuration for the TM-device discarded packets.

```
device# show system monitor tm discard-packet
Discard packet count monitoring configuration:
Logging-interval :  60 minutes
Threshold        :  10 packets
```

The following example displays the monitoring configuration for the VOQ discarded packets.

```
device# show system monitor tm discard-voq-packet
Discard VOQ packet count monitoring configuration:
Logging-interval :  60 minutes
Threshold        :  10 packets
```

### *Displaying the egress queue state information for an interface*

You can display the summary of the runtime egress queue state information applied to a Layer 2 interface. This information is retrieved from the dataplane.

To display this information, use the **show qos tx-queue interface** command, as shown in the following example.

```
device# show qos tx-queue interface ethernet 0/1
Interface Ethernet 0/1
          In-use        Max             TX        Dropped               TX           Dropped
  TC       Bytes      Bytes        Packets        Packets            Bytes             Bytes
  ------------------------------------------------------------------------------------------
    0          0     748288              0              0                0                 0
    1          0     748288    35739153669              0    1133120185038                 0
    2          0     748288              0              0                0                 0
    3          0     748288              0              0                0                 0
    4          0     748288              0              0                0                 0
    5          0     748288              0              0                0                 0
    6          0     748288              0              0                0                 0
    7          0     748288       30715725              2       2765239372               164
```

# Configuring scheduling

Perform the following tasks to configure scheduling.

### *Configuring strict priority egress scheduling*

Follow these steps to configure strict priority scheduling.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Select the policy map.

```
device(config)# policy-map policy_1
```

3. Select the classification.

```
device(config-policymap)# class default
```

4. Specify the scheduling attributes.

```
device(config-policymap-class)# scheduler strict-priority 3 dwrr 10 10 10 10 60 TC535000 TC6 36000
TC7 37000
```

5. Return to privileged EXEC mode.

```
device(config-policymap-class)# end
```

6. Verify the configuration.

```
device# show running-config | include strict-priority
  scheduler strict-priority 3 dwrr 10 10 10 10 60 TC5 40000 TC6 41000 TC7 42000
```

7. Enter global configuration mode.

```
device# configure terminal
```

8. Enter interface configuration mode.

```
device(config)# interface ethernet 0/1
```

9. Bind the policy to the port.

```
device(conf-if-e-0/1)# service-policy out policy_1
```

10. Return to privileged EXEC mode.

```
device(conf-if-e-0/1)# end
```

11. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Strict priority scheduling configuration example

```
device# configure terminal
device(config)# policy-map policy_1
device(config-policymap)# class default
device(config-policymap-class)# scheduler strict-priority 3 dwrr 10 10 10 10 60 TC535000 TC6 36000 TC7
37000
device(config-policymap-class)# end
device# show running-config | include strict-priority
device(config)# interface ethernet 0/1
device(conf-if-e-0/1)# service-policy out policy_1
device(conf-if-e-0/1)# end
device# copy running-config startup-config
```

## *Configuring the QoS multicast queue to a strict priority on an interface*

To configure multicast QoS follow these steps.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device# interface ethernet 0/1
```

3. Set the QoS scheduler to a strict priority.

```
device(conf-if-eth-0/1)# qos rx-queue multicast best-effort-rate 3000
```

   The multicast best effort data rate is in kilobits per second (kbps) and has range from 704 through 600000000.

4. Return to privileged exec mode.

```
device(conf-if-eth-0/1)# end
```

5. Verify the configuration.

```
device# show qos interface 0/1
```

6. Save the `running-config` file to the `startup-config` file

```
device# copy running-config startup-config
```

### QoS Multicast queue to a strict priority on an interface configuration example.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(conf-if-eth-0/1)# qos rx-queue multicast best-effort-rate 3000
device(conf-if-eth-0/1)# end
device# show qos interface 0/1
device# copy running-config startup-config
```

# Configuring flow-based QoS

Follow these tasks to configure ingress flow-based QoS.

## *Configuring a class map using an ACL*

To configure a classification or class map by using an ACL, follow these steps.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create an IP access list to define the traffic.
   a) Create and name a standard IP access list and enter IP ACL configuration mode.

      ```
      device(config)# ip access-list standard ip_acl
      ```

   b) Allow traffic from a specific IP address.

      ```
      device(conf-ipacl-std)# permit host 10.10.10.0
      ```

   c) Exit IP ACL configuration mode to global configuration mode.

      ```
      device(conf-ipacl-std)# exit
      ```

   For details on creating access lists, refer to the *Extreme SLX-OS Security Configuration Guide* for the device.
3. Verify the IP ACL.

   ```
   device(config)# do show running-config | include ip_acl
   ip access-list standard ip_acl
   ```

4. Create and name a class map.

   ```
   device(config)# class-map class_1
   ```

5. Provide match criteria for the class.

   ```
   device(config-classmap)# match access-group ip_acl
   ```

6. Return to privileged EXEC mode.

   ```
   device(config-classmap)# end
   ```

7. Verify the class configuration.

   ```
   device# show running-config | include class
   ...
   class-map cee
   class-map class_1
   class-map default
   ```

8. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Class map using an ACL configuration example

```
device# configure terminal
device(config)# ip access-list standard IP_acl
device(conf-ipacl-std)# permit host 10.10.10.0
device(conf-ipacl-std)# exit
device(config)# do show running-config | include ip_acl
device(config)# class-map class_1
device(config-classmap)# match access-group ip_acl
device(config-classmap)# end
device# show running-config | include class
device# copy running-config startup-config
```

## *Configuring a policy map*

Follow these steps to create a policy map.

A rate limit policy map is configured and then applied to the type of QoS flow defined by the class map.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create and name a policy map.

```
device(config)# policy-map policyMap1
```

3. Return to privileged EXEC mode.

```
device(config-policymap)# end
```

4. Verify the configuration

```
device# show policy-map

Number of policy maps : 2

Policy-Map policy
  Bound To:None

Policy-Map policyMap1
  Bound To:None
```

5. Display policy map details.
6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Policy map configuration example

```
device# configure terminal
device(config)# policy-map policyMap1
device(config-policymap)# end
device# show policy-map
device# copy running-config startup-config
```

## Binding the policy map at the system level

Follow these steps to apply policing parameters to an interface.

1. Enter global configuration mode.

   .

   ```
   device# configure terminal
   ```

2. Bind the policy map to inbound traffic.

   ```
   device(config)# qos service-policy in policyMap1
   ```

   You cannot use a policy map that is bound to class maps, default, or CEE maps.

3. Return to privileged EXEC mode.

   ```
   device(config-service-policy-in/policyMap1)# end
   ```

4. Verify the configuration.

   ```
   device# show policy-map detail policyMap1

   Policy-Map policyMap1
       Class class_1
         Police cir 40000 cbs 5000 eir 40000 ebs 3000

     Bound To: none
   ```

5. Save the `running-config` file to the `startup-config` file.

   ```
   device# copy running-config startup-config
   ```

### Binding a policy map globally configuration example

```
device# configure terminal
device(config)# qos service-policy in policyMap1
device(config-service-policy-in/policyMap1)# end
device# show policy-map detail policyMap1
device# copy running-config startup-config
```

## Binding the policy map to an interface

Follow these step to configure the default remapping priorities.

Consider the following rules when binding a policy map to an interface:

- You can bind the same policy map to multiple interfaces but only one policy per interface per direction is allowed.
- You cannot bind policy maps to an interface if the policy map has no class map associations.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

   ```
   device(config)# interface ethernet 0/4
   ```

3. Bind a policy map to ingress traffic on the interface.

```
device(config-if-eth-0/4)# service-policy in policyMap1
```

4. Return to privileged EXEC mode.

```
device(config-if-eth-0/4)# end
```

5. Verify the configuration.

```
device# show policy-map interface ethernet 0/4

Ingress Direction :
 Policy-Map policyMap1
    Class class_1
      matches 0 packets
      Police cir 40000 cbs 5000 eir 40000 ebs 3000
        Stats:
          Operational cir:39856 cbs:5000 eir:39856 ebs:3000
          Conform Byte:0 Exceed Byte:0 Violate Byte:0
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Binding the policy map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/4
device(conf-if-eth-0/4)# service-policy out policyMap1
device(conf-if-eth-0/4)# service-policy in policyMap1
device(conf-if-eth-0/4)# end
device# show policy-map interface ethernet 0/4
device# copy running-config startup-config
```

## *Configuring QoS mutation map actions*

Follow these steps to configure a QoS mutation map.

A policy map and a class map have been configured.

Different kinds of mutations can be used depending on the command. For complete information, refer to relevant Command Reference guide. The available commands are **cos-mutation**, **cos-traffic-class**, **dscp-cos**, **dscp-mutation**, and **dscp-traffic-class**.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Select the policy map.

```
device(config)# policy-map policyMap1
```

3. Select the class.

```
device(config-policymap)# class default
```

4. Specify the mutation map.

```
device(config-policyclass)# map dscp-cos all-zero-map
```

In this example a DSCP-to-CoS mutation is configured.

5. Return to privileged EXEC mode.

```
device(config-policyclass)# end
```

6. Verify the configuration.

```
device# show run policy-map
policy-map policyMap1
 class default
  map dscp-cos all-zero-map
 !
```

7. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### QoS mutation map configuration example

```
device# configure terminal
device(config)# policy-map policyMap1
device(config-policymap)# class default
device(config-policyclass)# map dscp-cos all-zero-map
device(config-policyclass)# end
device# show run policy-map
device# copy running-config startup-config
```

## *Applying QoS mutation maps to an interface*

Follow these steps to specify the mutation map to be used on a port.

A mutation map has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Select the policy map.

```
device(config)# policy-map policyMap1
```

3. Enter interface configuration mode.

```
device(config-policymap)# interface ethernet 0/1
```

4. Apply a DSCP-to-DSCP mutation map to the interface.

```
device(conf-if-eth-0/1)# qos dscp-mutation dscpMutMap
```

5. Return to privileged EXEC mode.

```
device(conf-if-eth-0/1)# end
```

6. Verify the configuration.

```
device# show qos map dscp-mutation dscpMutMap

Dscp-to-Dscp Mutation map 'dscpMutMap' (dscp= d1d2)
d1 :  d2 0  1  2  3  4  5  6  7  8  9
---------------------------------------
 0 :    11 11 11 03 11 11 11 11 11 11
 1 :    11 11 11 11 11 11 11 11 11 11
 2 :    11 11 11 23 24 25 26 27 28 29
 3 :    30 31 32 33 34 35 36 37 38 39
 4 :    40 41 42 43 44 45 46 47 48 49
 5 :    50 51 52 53 54 11 11 11 11 11
 6 :    11 11 11 11

   Enabled on the following interfaces:
    Eth 0/1
```

7. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### Applying a QoS mutation map to an interface configuration example

```
device# configure terminal
device(config)# policy-map policyMap1
device(config-policymap)# interface ethernet 0/1
device(conf-if-eth-0/1)# qos dscp-mutation dscpMutMap
device(conf-if-eth-0/1)# end
device# show qos map dscp-mutation dscpMutMap
device# copy running-config startup-config
```

## *Configuring the QoS policing rate*

To configure QoS for rate policing on an interface, you apply a policy map top the interface.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a policy map and enter policy map configuration mode.

```
device(config)# policy-map policy_1
```

3. Under policy map configuration mode, attach the classification map to the policy map.

```
device(config-policymap)# class default
```

4. Set the QoS action.

```
device(config-policymap-class)# police cir 40000
```

5. Return to privileged EXEC mode.

```
device(config-policymap-class)# end
```

6. Verify the configuration.

```
device#  show policy-map detail policy_1

Policy-Map P1
    Class default
      Police cir 40000

  Bound To:None
```

7. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### QoS policing rate configuration example

```
device# configure terminal
device(config)# policy-map policy_1
device(config-policymap)# class default
device(config-policymap-class)# police cir 40000
device(config-policymap-class)# end
device# show policy-map detail policy_1
device# copy running-config startup-config
```

## *Applying the QoS policing rate to an interface*

Follow these steps to apply the policing rate to an interface.

A policy map has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Access interface configuration mode.

```
device(config-policymap-class)# interface ethernet 0/4
```

3. Bind the ingress policy map policy map to the interface.

```
device(conf-if-eth-0/4)# service-policy in policy_1
```

4. Return to privileged EXEC mode.

```
device(conf-if-eth-0/4)# end
```

5. Verify the configuration.

```
device# show policy-map

Number of policy maps : 2
...
Policy-Map policy_1
  Bound To: Et 0/4(in)
```

6. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

### QoS policing rate on an interface configuration example

```
device# configure terminal
device(config-policymap-class)# interface ethernet 0/4
device(conf-if-eth-0/4)# service-policy in policy_1
device(conf-if-eth-0/4)# end
device# show policy-map
device# copy running-config startup-config
```

# Configuring virtual output queueing

Virtual output queueing (VOQ) is a technique where instead of one input traffic queue, multiple queues are maintained.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Configure an interface ingress queue, enter interface configuration mode.

   ```
   device(config)# interface ethernet 0/2
   ```

3. Use the traffic class to set parameters for unicast packet handling on the interface.

   ```
   device(conf-if-eth-0/2)# qos rx-queue unicast traffic-class 3 min-queue-size 128 max-queue-size 1024
   ```

   This command sets the values for the traffic class value, with a range of 0 through 7; minimum queue size, with a range of 0 through 1024 KB per-second (KBps); and the maximum queue size, with a range of 0 through 2048 MB per-second (MBps).

4. Use the traffic class to set parameters for multicast packet handling on the interface.
   a) Configure multicast data best effort rate

      ```
      device(conf-if-eth-0/2)# qos rx-queue multicast best-effort-rate 3000
      ```

      The range of values is from 0 through 600000000 kilobits per-second (kbps).

   b) Configure multicast data guarantee rate

      ```
      device(conf-if-eth-0/2)# qos rx-queue multicast guarantee-rate 30000
      ```

      The range of values is from 0 through 600000000 kbps.

   c) Set parameter values, by traffic class, for multicast packet handling on the interface.

      ```
      device(conf-if-eth-0/2)# qos rx-queue multicast traffic-class 3 min-queue-size 512 max-queue-
      size 1024
      ```

5. Return to privileged exec mode

   ```
   device(conf-if-eth-0/2)# end
   ```

6. Verify the configuration.

   ```
   device# show qos rx-queue interface all

   device# show running-config | include queue
   qos rx-queue queue-size 512
   ```

7. View the buffer pool statistics..

   ```
   device# show buffmgr stats slot 1
   ```

8. Save the `running-config` file to the `startup-config` file.

```
device# copy running-config startup-config
```

## *Virtual output queueing configuration example*

```
device# configure terminal
device(config)# interface ethernet 0/2
device(conf-if-eth-0/2)# qos rx-queue unicast traffic-class 3 min-queue-size 128 max-queue-size 1024
device(conf-if-eth-0/2)# qos rx-queue multicast best-effort-rate 3000
device(conf-if-eth-0/2)# qos rx-queue multicast guarantee-rate 30000
device(conf-if-eth-0/2)# qos rx-queue multicast traffic-class 3 min-queue-size 512 max-queue-size 1024
device(conf-if-eth-0/2)# end
device# show qos rx-queue interface all
device# copy running-config startup-config
```