Extreme SLX-OS QoS and Traffic Management
Configuration Guide, 20.3.4

Supporting ExtremeRouting and ExtremeSwitching
SLX 9740, SLX 9640, SLX 9540, SLX 9250
SLX 9150, Extreme 8720, and Extreme 8520

Table of Contents

Preface

Read the following topics to learn about:

- The meanings of text formats used in this document.
- Where you can find additional information and help.
- How to reach us with questions and comments.

Text Conventions

Unless otherwise noted, information in this document applies to all supported environments for the products in question. Exceptions, like command keywords associated with a specific software version, are identified in the text.

When a feature, function, or operation pertains to a specific hardware product, the product name is used. When features, functions, and operations are the same across an entire product family, such as ExtremeSwitching switches or SLX routers, the product is referred to as *the switch* or *the router*.

Table 1: Notes and warnings

Icon	Notice type	Alerts you to...
	Tip	Helpful tips and notices for using the product
	Note	Useful information or instructions
	Important	Important features or instructions
	Caution	Risk of personal injury, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 2: Text

Convention	Description
`screen displays`	This typeface indicates command syntax, or represents information as it is displayed on the screen.
The words *enter* and *type*	When you see the word *enter* in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says *type*.
Key names	Key names are written in boldface, for example **Ctrl** or **Esc**. If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press **Ctrl**+**Alt**+**Del**
Words in italicized type	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles.
NEW!	New information. In a PDF, this is searchable text.

Table 3: Command syntax

Convention	Description
bold text	Bold text indicates command names, keywords, and command options.
italic text	Italic text indicates variable content.
[ ]	Syntax components displayed within square brackets are optional. 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, such as passwords, are enclosed in angle brackets.
…	Repeat the previous element, for example, `member[member...]`.
\	In command examples, the backslash indicates a "soft" line break. When a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

Find Extreme Networks product information at the following locations:

> Current Product Documentation
>
> Release Notes
>
> Hardware and software compatibility for Extreme Networks products
>
> Extreme Optics Compatibility
>
> Other resources such as white papers, data sheets, and case studies

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

Help and Support

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: (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, or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any actions 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

Subscribe to Product Announcements

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

1. Go to The Hub.
2. In the list of categories, expand the **Product Announcements** list.
3. Select a product for which you would like to receive notifications.
4. Select **Subscribe**.
5. To select additional products, return to the **Product Announcements** list and repeat steps 3 and 4.

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

Send Feedback

The Information Development team at Extreme Networks has made every effort to ensure that this document is accurate, complete, and easy to use. We strive to improve our documentation to help you in your work, so we want to hear from you. We welcome all feedback, but we especially want to know about:

- Content errors, or confusing or conflicting information.

- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, do either of the following:

- Access the feedback form at https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@extremenetworks.com.

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

What's New in this Document

The following table describes information added to this guide for the SLX-OS 20.3.4 software release.

Table 4: Summary of changes

Feature	Description	Described in
Support for IPv4 Egress Rate Limiting.	Added support for IPv4 egress rate limiting on SLX 9740 devices.	Configuring IPv4 Egress Rate Limiting on page 46

For complete information on this SLX-OS software release, refer to the *Extreme SLX-OS Release Notes* for this version.

Supported Hardware

For instances in which a topic or part of a topic applies to some devices but not to others, the topic specifically identifies the devices.

SLX-OS 20.3.4 supports the following hardware platforms.

- Devices based on the Broadcom XGS® chipset family:
  - Extreme 8720
  - Extreme 8520
  - ExtremeSwitching SLX 9250
  - ExtremeSwitching SLX 9150
- Devices based on the Broadcom DNX® chipset family:
  - ExtremeRouting SLX 9740
  - ExtremeRouting SLX 9640

- ○ ExtremeSwitching SLX 9540

> **Note**
>
> All configurations and software features that are applicable to SLX 9150 and SLX 9250 devices are also applicable for the Extreme 8520 and Extreme 8720 devices respectively.
>
> The "Measured Boot with Remote Attestation" feature is only applicable to the Extreme 8520 and Extreme 8720 devices. It is not supported on the SLX 9150 and SLX 9250 devices.

> **Note**
>
> Although many software and hardware configurations are tested and supported for this release, documenting all possible configurations and scenarios is beyond this document's scope.

For information about other releases, see the documentation for those releases.

Traffic Policing

Traffic Policing Overview

Traffic policing is the process of monitoring network traffic for compliance with a traffic policy and then enforcing that policy. Traffic policing involves such tools as rate limiting and shaping, CIR, EIR, color markers, service policies, class and policy maps, and storm control.

Rate limiting and shaping

Rate limiting controls the amount of bandwidth that is consumed by an individual flow or an aggregate of flows. For inbound and outbound traffic, rate limiting drops packets that exceed committed rates.

For more information, see the following topics.

- VLAN-based rate limiting on page 33
- Bridged-domain based rate limiting on page 36
- Receive ACL Rate Limiting on page 39
- Egress ACL Rate Limiting on page 41
- Rate Limiting Scalability on page 51

Rate shaping controls traffic bursts applicable to egress traffic by buffering and queuing excess packets that are above the committed rate.

CIR and EIR

The Committed Information Rate (CIR) is the amount of available bandwidth that is committed to the user. Available bandwidth should not fall below this committed rate.

The Excess Information Rate (EIR) is an accommodation that you configure for traffic that exceeds the CIR.

For more information, see:

- Committed Information Rate and Committed Burst Size on page 14
- Excess Information Rate and Excess Burst Size on page 15

Color markers

The single-rate, three-color marker (SrTCM) and the two-rate, three-color marker (TrTCM) indicate traffic compliance with bandwidth requirements. SrTCM is based on RFC 2697. TrTCM is based on RFC 4115.

For more information, see:

- Single rate three color marker on page 20
- Two-rate, three-color marker on page 21

Service policies

A service policy consists of a policy map that specifies traffic policing rules and QoS parameters that match associated class maps. One service policy can be applied per interface, per direction.

For more information, see:

- Service Policy Configuration Rules
- Policy Maps on page 14
- Class Maps on page 17

Storm control

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

For more information, see Storm Control for Broadcast, Unknown Unicast, and Multicast Traffic on page 68.

Service Policy Configuration Rules

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

Follow these rules when configuring 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.
- You can bind a service policy only to physical ports and port-channel interfaces (LAGs). You cannot bind it to virtual interfaces.
- You cannot bind a service policy 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 on an interface.

Policy Maps

A policy map is a unique set of class maps, policing parameters, and QoS parameters that you can apply to a certain class of traffic.

When you configure a policy map, you specify traffic policing parameters in one location that can be applied to multiple ports. Follow these rules when configuring traffic policing:

- A policy map name must be unique among all maps of this type.
- A policy map name must begin with an alphabetic character (a-z or A-Z). An underscore, a hyphen, and the 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 1,024 policy maps.
- You can specify only one policy map per service policy.
- You can use ACL-based class maps and default class maps in one policy map.
- ACL-based, storm-control, and port-based rate limiting can coexist on an 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, you can specify one default class map per policy map.
- Broadcast, unknown unicast, and multicast (BUM) policies are counted separately.
- You cannot delete a policy map that is referenced in a service policy that is applied on an interface.
- For each port attached to the policy map, you can specify the shaping rate for smoothing egress traffic from an interface.

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).

The CIR is the maximum number of bits that a port can 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 bandwidth, expressed in bits per second (bps), and it cannot exceed the port's bandwidth.

CIR-defined traffic that does not use its available CIR accumulates credit up to the amount defined by the CBS. This credit is the number of bytes that can be used to accommodate temporary bursts in traffic that exceeds the CIR.

Excess Information Rate and Excess Burst Size

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

EIR and EBS accommodate traffic that exceeds the conditions of the committed information rate (CIR) and the committed burst size (CBS). EIR is the maximum sustained rate at which inbound and outbound traffic can exceed the CIR. Traffic that does not use all of its allotted EIR accumulates credit (bandwidth) up to the amount defined by the EBS. This accumulated bandwidth can be used to accommodate temporary bursts of traffic that exceeds the EIR. Inbound or outbound traffic that exceeds the amount of accumulated bandwidth is dropped.

Traffic Policing Behaviors

Traffic policing is the process of monitoring network traffic for compliance with a traffic policy and then enforcing that policy.

- Policing actions are applicable only to data traffic.
- When a Layer 2 control protocol is not enabled on an interface, 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 committed burst size (CBS) in the hardware. For example, 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 updated.
- When the CBS and excess burst size (EBS) values are not configured, the values are derived from the committed information rate (CIR) and excess information rate (EIR), respectively. The burst size calculation is as follows:
  ```
  Burst size (CBS or EBS) = (1.2 × information rate (CIR or EIR)) ÷ 8
  ```
- The underlying architecture of SLX 9740 supports the burst size only upto 256MegaBytes, Any higher Burst size configuration will result in Operational Burst size of 256MegaBytes.
- You are responsible for configuring 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 policing attribute, you cannot delete the CIR parameter. However, you can delete the CIR attribute by using the **no police** command in policy-map-class sub-mode.
- Packet drops caused by any action other than the ACL are included in the policing counter.
- Layer 3 control packets are policed at the egress side.
- When you attempt to bind the policer with a configured CIR or EIR value that is less than minimum supported CIR of the platform, the operational CIR or EIR value becomes zero and you receive a syslog notification on the console.
- The minimum CIR/EIR in SLX 9150, SLX 9250 and SLX 9740 platforms is 1000 bits/sec.
- The minimum cir/eir in SLX 9540 is 22000 bits/sec. On SLX 9640 minimum cir/eir is 26000 bits/sec.

- Unknown-unicast storm control does not rate limit the VLL or P2P traffic in the default TCAM profile.
- When an ingress service policy is bound on a port channel that has multiple member ports, then all port-channel member ports that belong to the same chip-core are governed by the same policy. The configured rate is per chip-core. For example:
  - A service policy with a CIR or EIR value of 10 Gpbs is applied on port channel 10. Port channel 10 has member ports 1/1 and 1/3 that belong to chip 1-core 0. So the total rate-limit value for the port channel is 10 Gpbs.
  - A service policy with a CIR or EIR value of 10 GPBS is applied on port channel 10. Port channel ten has member ports 1/6 and 1/13 that belong to chip 1-core 0, and chip 1-core 1. So the total rate-limit value of the port channel is 2*10, or 20 Gpbs.
  - A service policy with a CIR or EIR value of 10 GPBS is applied on port channel 10. Port channel ten has member ports 1/10 and 3/10 that belong to chip 1-core x, and chip 3-core x. So the total rate-limit value of the port channel is 2*10, or 20 Gpbs.

Traffic management egress buffer thresholds

For egress buffer management in the SLX 9150 device, all ports and all priorities share a buffer pool of approximately 30MB. The buffers are dynamically allocated. If there is only one queue that is congested, it can take up to 66.67% of the shared pool buffer to absorb a large burst. You can view this information as max-bytes through the **show tx-queue internet ethernet** command. However, as the number of congested queues increases, the shared pool continues to divide. As a result, the number of buffers that a queue can take up continues to become less. You can view the real-time status of the max-bytes that a queue is using by using the **show tx-queue internet ethernet** command.

For all other devices, during the traffic management (TM) initialization process, TM egress buffer thresholds are configured and set to the values listed in the following tables.The following table shows the egress thresholds for unicast and multicast traffic that are configured during the TM initialization process at the device level.

Table 5: TM egress thresholds on the device

Threshold	Packet descriptor (PD) flow control (FC)	PD Drop	DB (Data buffers) FC	DB Drop
Unicast	6100	6000	6100	6000
Multicast		26000		6000

The following table shows the egress thresholds for unicast and multicast traffic that are configured during the TM initialization process at port level on the device.

Table 6: Local port TM egress thresholds on the device

Threshold	Packet descriptor (PD) flow control (FC)	PD Drop	DB (Data buffers) FC	DB Drop
Unicast priority low	1024	4000	84	6000
Unicast priority high	1024	4000	84	6000
Unicast port	167	6000	167	6000
Multicast priority low		722		7220
Multicast priority high		722		7220
Multicast port		135		1350

> **Note**
> The burst size on special CPU ports (202 and 203) is set to 600.

Class Maps

Class maps are used in a policy map to apply policing and QoS policies to a particular class of traffic. You can use matching criteria to classify the traffic.

The port-based rate limit, applied with the default class map, is applicable to all traffic and can be used for ingress and egress service policies. Port-based policing is only implemented to match any criteria.

You can use an IP standard or extended ACL to classify traffic for ingress-only service policies. You can use an ACL for match criteria.

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 (a-z or A-Z). An underscore, a hyphen, and the numeric values 0-9 can be used in the body of the name but not as the first character.
- You cannot delete a class map that is referenced in a policy map.
- You cannot delete a class map when the policy map is bound to an active service policy.
- You can configure a maximum of 32,000 class maps.
- You can combine the default and match access-group class maps in one policy map.
- You can specify only one default class map 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 7: Map parameters for rate limiting in SLX 9540

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) policing feature controls the amount of bandwidth consumed by a flow or an aggregate of flows.

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

Consider the following when you configure the feature:
- You can configure up to 1024 policy maps.
- You can configure one default class map per policy map.
- For ExtremeRouting SLX 9640 and ExtremeSwitching SLX 9540, you can configure up to 32656 policers (ingress - egress).
- For ExtremeSwitching SLX 9150, you can configure up to 8000 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.
- For port-based traffic policing, only FWD and DROP counters are supported.
- SLX 9540 and SLX 9640
  - FWD and DROP counters must use a counter profile other than default.
  - **Conformed**: Shows FWD packets including green and yellow color packets.
  - (SLX 9640 and SLX 9540 only) **Violated**: Shows DROP packets including red color packets.
  - (SLX 9640 and SLX 9540 only) **Exceed**: Is always ZERO.

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.
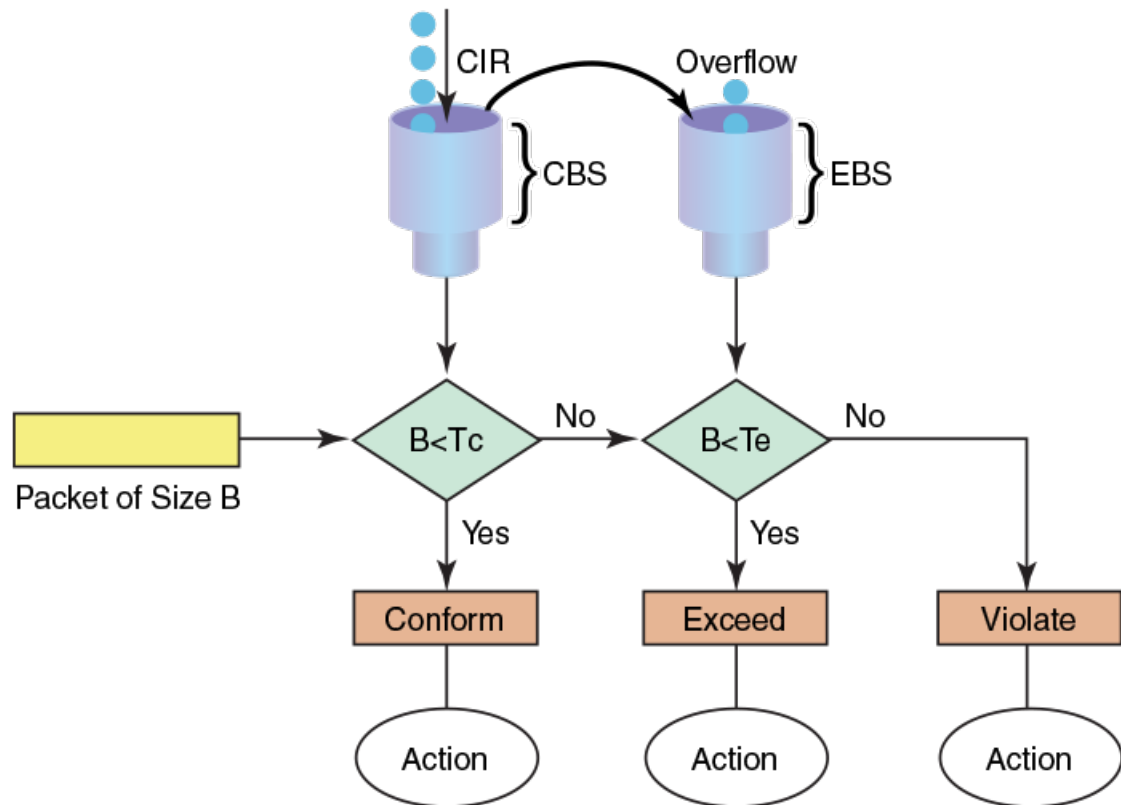
Figure 1: Single rate three color marker

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.
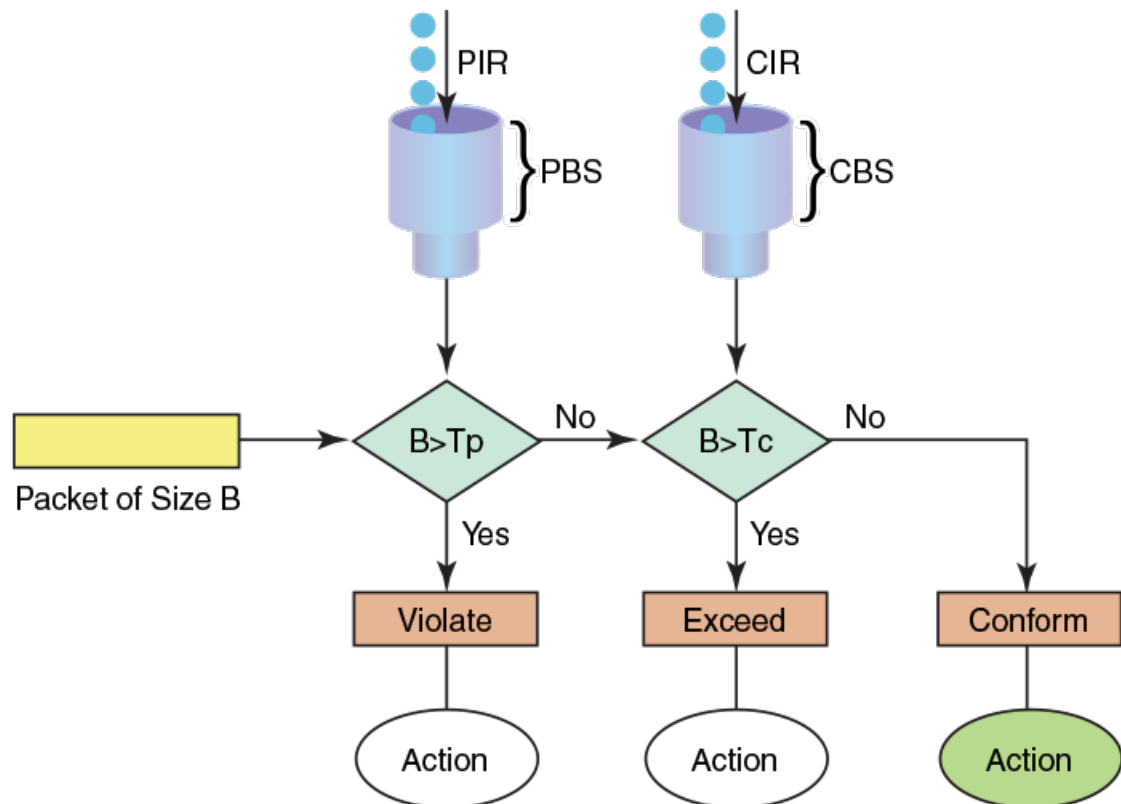
Figure 2: Two-rate, three-color marker

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

The ACL-based policing feature controls the amount of bandwidth consumed by an individual flow or aggregate of inbound flows by limiting the traffic rate on a port according to criteria defined by the `match access-group` class map.

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

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

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

   Layer 2 ACL-based rate limiting can occur on VPLS endpoints when the TCAM profile is set to Default and MAC ACLs support VPLS-based filtering. You should configure the Layer 2 filter parameters to match the outer VLAN, VLAN-tag format and to match the inner VLAN based on the traffic received on a logical interface (LIF) for which rate limiting is required. For more information on filtering by the VLAN-tag type, see the *Extreme SLX-OS Security Configuration Guide*.

> **Note**
> Layer 2 ACL-based rate limiting on VPLS endpoints is supported only on devices based on the DNX chipset family. For more information, see Supported Hardware.

Consider the following when you configure `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
- Ternary Content Addressable Memory (TCAM) entries for use with rate limiting and ingress policers are dependent on the hardware TCAM profile that is used.
- The ACL-based rate limiting feature can serve as a hardware solution to prevent DoS attacks, including:
  - PING attacks
  - TCP Reset attacks
  - TCP SYN attacks
  - UDP attacks
- Layer 2 MAC and Layer 3 IPv4 ACL-based rate limiting are supported.
- ACL-based rate limiting applies only to ingress traffic.
- There is one policer per ACL, which applies to all the rules for that ACL.
- Control protocols are rate-limited if they match the configured ACL clause.

Precedence for ACL and Rate Limiting Features

ACL features have the following precedence:

1. OpenFlow has precedence over rACL.
2. rACL has precedence over policy-based routing.

3. Policy-based routing has precedence over ACL.
4. ACL has precedence over VLAN rate limiting.

Rate limiting features have the following precedence:

1. ACL rate limiting has precedence over BUM storm control.
2. BUM storm control has precedence over VLAN rate limiting and bridge domain rate limiting.
3. VLAN rate limiting and bridge domain rate limiting have precedence over port rate limiting.

Rate limiting on an interface or port-channel has precedence over system rate limiting.

All ACL and ACL rate limiting features reside in one of two TCAM databases.

Table 8: TCAM databases and features

Database	Feature
TCAM User	Layer 3 ACL
	Layer 2 ACL
	Layer 3 ACL rate limiting
	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, such as first come, first served. For inter-database features, when there is a hit in both databases, the device first looks at the following actions:

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

Considerations 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 bridge domain, 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.
- Packet 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, and ignoring FCS and CRC).
- A policy map can be applied to a physical port and to LAG interfaces.
- Multiple class maps with user-defined ACLs can be added in a policy map. However, after 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. Violated includes dropped or a RED color packet.
- You cannot delete a policy map that 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.
- When ACL rate limiting is applied to a LAG logical port, the maximum rate on this port is the number of towers in this LAG * CIR. For example, if Ethernet 1/1, 1/2 , 2/1, and 3/1 are LAG member ports, then the maximum rate is 3 * CIR.
- 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:
  ○ Layer 3 ACL rate limiting = 2K class maps
  ○ Layer 3 ACL global rate limiting = 512 class maps
  ○ Layer 2 ACL rate limiting = 512 class maps
  ○ Layer 2 ACL global rate limiting = 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 is limited by the TCAM size. For more information, see the scalability numbers in the release notes.
- Layer 2 ACL rate limiting can be configured on a VPLS or VLL endpoint when the default TCAM profile is enabled. Configure the Layer 2 ACL filter parameters to match outer-vlan, vlan-tag-format, and inner-vlan based on the traffic received on a LIF for which rate limiting is required.

  > **Important**
  > Layer 2 ACL-based rate limiting on VPLS endpoints is not supported on SLX 9150.

- To disable the learning of MAC addresses for stream matching of the ACL-based rate limiting entries, change the CIR and EIR values to 0.

  The MAC address entries in the MAC-address table that are already learned are not flushed when you configure the CIR or EIR value as 0. Use the `clear mac-address-table dynamic` command to explicitly clear the entries in the MAC-address table.

Configure Layer 2 ACL Rate Limiting

Before You Begin

- For Layer 2 ACL-based rate limiting on VPLS endpoints, ensure that the default TCAM profile is enabled.

  > **Important**
  > Layer 2 ACL-based rate limiting on VPLS endpoints is not supported on SLX 9150.

- Configure Layer 2 filter parameters to match the outer VLAN, VLAN-tag format, and inner VLAN based on the traffic received on a logical interface for which rate limiting is required. For more information on filtering by the VLAN-tag type, see the *Extreme SLX-OS Security Configuration Guide*.

- For Layer 2 ACL-based rate limiting to filter known unicast traffic only, configure a rule for a MAC extended ACL with the **known-unicast-only** keyword. This configuration is supported only for an ingress ACL on an L2_Ratelimit profile. Note that an implicit deny applies for both unknown and known unicast traffic.

About This Task

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.

Procedure

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
```

12. Access the interface configuration mode for the interface where 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
```

14. For egress rate limiting, enable the L2-Egress-RateLimit TCAM profile and then reboot the device.

```
device(config)# hardware
device(config-hardware)# profile tcam layer2-egress-ratelimit
device(config-hardware)# exit
device# copy running-config startup-config
device# reload system
```

Example

The following example summarizes the commands in this procedure.

```
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
```

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 and Configuring use case 1 - protection against TCP SYN attacks.

Use case 2 - protection against TCP RST attacks and Configuring use case 2 - protection against TCP RST attacks.

Use case 3 - protection against Ping attacks and Configuring use case 3 - protection against ping flood attacks.

Use case 4 - protection against UDP flood attacks and Configuring use case 4 - protection against UDP flood attacks.

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.

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.

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.

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.

Configuring all the use cases for ACL traffic filtering

You can configure all four use cases and apply them to a 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, and associate it to the ACL.
3. Create a policy map using the class map created in step 2, and assign a policer.
4. Associate the policy map to an ingress port.

Refer to Configuring all the use cases for ACL traffic filtering.

Control Plane Policing

Control Plane Policing helps regulate the flow of control packets to a local processor.

A packet that is processed by an ASIC in a device can have different destinations in the device. A packet can exit the switch from its front-end ports (the data path) or it can enter the local processor for further processing (the control path).

Control packets such as SSH, ICMP, Telnet, ARP, and BGP are handled by the local processor. These control packets are matched either as exact match or prefix match with address fields. A decision whether to trap to local processor is configured in the control classifier engine in the ASIC. The control packets can be trapped to the local processor at the highest rate the software module expects to handle. The rest of the packets are dropped.

Each type of control packet has a different level of significance in software modules. Some protocols are critical for operations and maintenance, some are intolerant to latency, and others are intolerant to packet loss. Over-subscription of the control path is a typical problem and it needs regulated policing. Rogue packets from events (malicious and non-malicious) can overwhelm processor resources and bring down critical operations on the processors. Therefore, it is essential to regulate the flow of control plane packets based on control packet type.

Control Plane Policing (CoPP) helps regulate the flow of control packets to the local processor at a predefined rate, up to and including discarding the packets. The control plane handles various types of flows:

* **Control Classifier Trapped Frames**: Well-known protocol packets that are processed by protocol software handlers in the local processor. This scenario uses either a Control Classifier or an ACL engine to trap flows to the CPU.
* **Access List (ACL) Traps**: Exceptions that can be handled as part of regular traffic. This scenario uses either a Control Classifier or an ACL engine to trap flows to the CPU.
* **Hardware-based Trap Conditions**: Exception scenarios built into the hardware pipeline that may not take the classification route.

SLX-OS allows for pattern matching using a variety of packet fields and signatures in flows sent toward the control plane and uses pattern matching engines to trap traffic toward the CPU. CoPP extends the action with components that offer policing, metering, and denial to throttle or drop the pattern-matched control packets. CoPP allows individual flows to be controlled at a granular level according to your needs.

CoPP Discard and Permit for Control Packets

You can use received ACL (RACL) policies to permit or deny unicast and multicast control packets that are destined to the local processor.

You can configure RACL entries with match criteria for packets and an action for discarding packets. RACL policies can permit or deny control packets and also count the number of packets hitting the policy.

The process for creating a RACL that allows or denies control packets is as follows:

1. Create an IP access list (ACL).
2. Bind the ACL to an IP Receive Access Group.

For more information about RACL, see Receive ACL Rate Limiting on page 39.

For more information about ACLs, see the *Extreme SLX-OS Security Configuration Guide*.

For more information about the commands in the following examples, see the *Extreme SLX-OS Command Reference*.

Examples for permitting and denying SSH flows (unicast)

This example creates an IPv4 ACL and binds it to a Receive Access Group. It allows all flows that match the originating subnet and discards the rest of the SSH flows.

```
ip access-list extended <name_ssh_flow> seq <num> permit tcp <ip network addr>
<subnet mask> any eq 22 count

ip receive access-group <name_ssh_flow> sequence <num>
```

This example allows two subnets to send SSH packets to the local CPU, discards the rest of the flows, counts the number of discards, and binds the IPv4 ACL to a Receive Access Group.

```
ip access-list extended ip-ssh
 seq 10 permit tcp 10.10.10.10 0.0.0.255 any eq 22 count
 seq 20 permit tcp 11.11.11.11 0.0.0.255 any eq 22 count
 seq 100 deny tcp any any eq 22 count

ip receive access-group ip-ssh sequence 10
```

This example creates an IPv6 ACL that allows two subnets to sent SSH packets, discards the rest of the flows, counts the number of discards, and binds the ACL to a Receive Access Group.

```
ipv6 access-list extended ipv6-ssh
 seq 10 permit tcp 2000::/32 any eq 22 count
 seq 20 permit tcp 2004::/32 any eq 22 count
 seq 100 deny tcp any any eq 22 count

ipv6 receive access-group ipv6-ssh sequence 10
```

Example for permitting and denying VRRP advertisement packets (multicast)

This example permits VRRP advertisement packets from a specific subnet, discards them from other IP addresses, counts the number of discards, and binds the ACL to a Receive Access Group.

```
ip access-list extended ip-vrrp
 seq 10 permit 112 host 10.1.1.10 host 224.0.0.18 count
 seq 20 permit 112 7.7.7.7 0.0.0.255 host 224.0.0.18 count
 seq 100 deny 112 any host 224.0.0.18 count

ip receive access-group ip-vrrp sequence 20
```

> **Note**
> CoPP for multicast flows does not support IPv6 control packets.

CoPP Rate Limiting

You can use RACL entries to configure rate limiting actions that police the traffic to the control plane.

The process for configuring rate limiting is as follows:

1. Configure a permit IP ACL for matching the flow.
2. Create a class-map binding to that ACL.
3. Create the policy map for Control Plane protection.
4. Bind the policy map to the Control Plane interface as a service policy toward the ingress direction.

> **Note**
> You can bind only one service policy at the control plane interface, so all control plane rate-limiting configurations must be bound in the same policy-map with different class-maps. Each class-map can bind to only one IP access-group. A policy map can bind to multiple class-maps, and each class-map can be configured with a rate limit.

For more information about RACL, see Receive ACL Rate Limiting on page 39.

For more information about ACLs, see the *Extreme SLX-OS Security Configuration Guide*.

For more information about class maps, see Class Maps on page 17.

For more information about the commands in the following examples, see the *Extreme SLX-OS Command Reference*.

Example for allowing flows at different rates

This example creates an ACL to allow one SSH flow at the rate of 100 Kbps and another ACL to allow SSH flows at the rate of 1 Mbps toward the Control Plane. All other flows are not rate limited. Class maps are bound to the ACL and the policy map is created and bound to the Control Plane interface.

```
# Create an IP-ACL to permit SSH flows originating from host 11.11.11.100
ip access-list extended ip-ssh-2
 seq 10 permit tcp host 11.11.11.100 any eq 22 count

# Create an IP-ACL to permit SSH flows originating from network 2.2.2.2/24
ip access-list extended ip-ssh-3
 seq 10 permit tcp 2.2.2.2 0.0.0.255 any eq 22 count

# Create a class-map that binds to IP ACL ip-ssh-2
```

```
class-map class-ssh-2
 match access-group ip-ssh-2

# Create a class-map that binds to IP ACL ip-ssh-3

class-map class-ssh-3
 match access-group ip-ssh-3
 !

# Create a policy-map map-ssh, with class-maps class-ssh-2 rate limited at 100 KBPS
# and with class-ssh-3 rate limited at 1 MBPS.

policy-map map-ssh
 class class-ssh-2
  police cir 100000
 !
 class class-ssh-3
  police cir 1000000
 !
!

# Bind above configured policy-map to control-plane interface.
control-plane
 service-policy in map-ssh
 !
```

CoPP-related Commands

Several commands can help you configure, safeguard, and troubleshoot Control Plane Policing.

Table 9: Configuration commands

Command	How to use for CoPP
`ip icmp-fragment enable`	Drop ICMP fragment packets before they are used by hackers for Denial of Service (DoS) attacks.
`ip option disable`	Discard IP packets with options before hackers send such packets to initiate DoS attacks.

`ip access-list extended`
`class-map`
`match access-group`
`policy-map`
	Configure rate limiting actions. For more information, see CoPP Rate Limiting on page 31.

`ip access-list extended`
`ip receive access-group`
`ipv6 access-list extended`
`ipv6 receive access-group`
	Permit or deny unicast and multicast control packets. For more information, see CoPP Discard and Permit for Control Packets on page 30.

Table 10: Show commands

Command	How to use for CoPP
`show access-list receive`	See the configuration for permit and deny rules for control plan protection.
`show statistics access-list`	See statistics for packets that meet the permit and deny rules configured for control plane protection.

Table 10: Show commands (continued)

Command	How to use for CoPP
`show policy-map control-plane`	See the configuration of the policy map attached to a control plane interface.
`show interface ethernet inc rate`	See whether the control plane is receiving packets at the configured rate.
`show qos cpu info` `show qos cpu cfg`	CPU ports that allow packets into the control plane have limited bandwidth. View the maximum CPU rates and weighted fair queue values for the various VOQ groups.

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 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, which applies to all the rules for that ACL.
- One thousand and twenty four (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 policing 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, a LAG interface, and a system.
- Multiple class maps with user ACLs can be added in a policy map. However, when a matching ACL clause is found, the device does not evaluate subsequent ACL clauses and rate limits those ACLs.
- A configured rate in bps is rounded up to the 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 packets. Violated includes dropped or a RED packet.
- You cannot delete a policy map that is active on the interface.
- You cannot delete a class map from a policy map when the policy map is active on the interface.
- (SLX 9640 and SLX 9540 only) Configure CIR and EIR in multiples of 22 kbps. If the value 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 are marked as green (conforming).
- If you do 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 an interface if a class map is not associated with that policy map.
- (SLX 9640 and SLX 9540 only) VLAN rate limiting does not work for VLANs that are carried inside a pseudowire (PW) tunnel.

Configure VLAN-based Rate Limiting

Procedure

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 interface configuration mode for the interface where you want to apply the policy map.

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

10. Apply the policy map to the interface.

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

11. Access global configuration mode.

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

12. (Optional) Apply the policy map globally.

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

Example

The following example summarizes the commands in this procedures.

```
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 1/1
device(conf-if-eth-1/1)# service-policy in p2
device(conf-if-eth-1/1)# exit
device(config)# qos service-policy in p2
```

Show Commands for VLAN-based Rate Limiting

You can use show commands to display information about VLAN-based rate limiting.

You can run show commands from any level of the CLI.

policy-map

Displays the rate limiting policies that are implemented in the configured policy maps, and displays policer confirm, exceeded, or violate counters. For example:

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

show policy-map interface ethernet 1/1 in

Displays interface-specific policy map information. For example:

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

clear policy-map-counters interface ethernet

Clears the policer counters for the interface. If you do not specify an interface, all rate limit counters are cleared on all interfaces.

show policy-map system map-name pm1

Displays system-specific policy map information. For example:

```
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
```

show running-config policy-map

Displays policy map configuration information. For example:

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

show running-config class-map

Displays class map configuration information. For example:

```
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 VPLS instance or 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 for Bridge Domain-based Rate Limiting

- Bridge domain-based rate limiting coexists and works in parallel with all other ACL and rate-limiting features.
- Bridge domain-based rate limiting applies only to ingress traffic only. Egress traffic is blocked by the SLX-OS device.
- There is one policer per ACL, which applies to all the rules for that ACL.
- One thousand and twenty four (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 policing 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, a LAG interface, and a system.
- Multiple class maps with user ACLs can be added in a policy map. However, when a matching ACL clause is found, the device does not evaluate subsequent ACL clauses and rate limits those ACLs.
- A configured rate in bps is rounded up to the 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 packets. Violated includes dropped or a RED packet.

- You cannot delete a policy map that 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 the Committed Information Rate (CIR) and Excess Information Rate (EIR) in multiples of 22 kbps. If the value 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 are marked as green (conforming).

- If the Committed Burst Size (CBS) and Excess Burst Size (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 do 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 a class map is not associated with that policy map.

Configuring bridge-domain based rate limiting

Before You Begin
Before configuring bridge-domain based limiting, the associated bridge-domain VPLS or VLL instance must be configured.

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

Procedure

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 1/4
```

10. Apply the policy map to the interface.

```
device(conf-if-eth-1/4)# service-policy in QOS-BD
```

11. Access global configuration mode.

```
device(conf-if-eth-1/4)# exit
```

12. Optionally, apply the policy map globally.

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

Example

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 1/4
device(conf-if-eth-1/4)# service-policy in QOS-BD
device(conf-if-eth-1/4)# exit
device(config)# qos service-policy in QOS-BD
```

Show Commands for Bridge Domain-based Rate Limiting

You can use show commands to display information about bridge domain-based rate limiting.

You can run show commands from any level of the CLI.

policy-map

Displays the rate limiting policies that are implemented in the configured policy maps, and displays policer confirm, exceeded, or violate counters. For example:

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

show policy-map interface ethernet 1/4 in

Displays interface-specific policy map information. For example:

```
device# show policy-map interface ethernet 1/4 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
```

clear policy-map-counters interface ethernet

Clears the policer counters for the interface. If you do not specify an interface, all rate limit counters are cleared on all interfaces.

show policy-map system map-name QOS-BD

Displays system-specific policy map information. For example:

```
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
```

show running-config policy-map

Displays policy map configuration information. For example:

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

show running-config class-map

Displays class map configuration information. For example:

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

Receive ACL Rate Limiting

IP Receive access list (RACL) provides hardware-based filtering capability for Layer 3 IPv4 or IPv6 traffic that is destined to the CPU.

RACL can protect the CPU from overloading due to heavy traffic that was sent to an IP interface on the device. Using the RACL, an ACL is applied at the system level to eliminate the need to add an ACL to each interface on the device. For more information about RACL, see the *Extreme SLX-OS Security Configuration Guide*.

Using policy maps, you can apply rate liming to an RACL on IPv4 and IPv6 traffic destined to the CPU control plane. Policy maps can support maximum of 1,000 class maps. The rate-limited RACL does not

have dedicated TCAM space. Instead, it shares the ACL TCAM space. RACL rate limiting for IPv4 and Ipv6 traffic is supported only in the default TCAM profile, which can support 2,048 entries.

Consider the following when configuring RACL:

- An ACL must be defined before it can be used for RACL rate limiting.
- Only one policy map can be applied on the control plane.
- A rate-limited RACL has a higher precedence than a user ACL configured on an interface.
- RACL rate-limiting rules have lower precedence than RACL rules.
- An IPv4 or IPv6 RACL supports all the matching criteria and actions that are supported by an IPv4 or IPv6 user ACL.
- A deny action in a rule is ignored for RACL rate limiting.
- No default drop entry is programmed for RACL limiting.
- The CPU shaper rate also polices traffic that is destined to the CPU. The default shaper rate is 10 Mbps. You can configure the shaper rate to a maximum of 150 Mbps.

> **Note**
> The RACL-RL match fields icmp_type, icmp_code, and ip_ttl are added in SLX 20.2.1. The match fields icmp_code and ip_ttl are valid only when icmp_type match is valid.
>
> The match fields icmp_type, icmp_code, and ip_ttl are supported only on SLX 9150 and SLX 9250.

Configuring RACL rate limiting

Before You Begin

Before configuring rate limiting for a RACL, configure an IPv4 or IPv6 ACL.

Procedure

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

   ```
   device# configure terminal
   ```

2. Create a class map.

   ```
   device(config)# class cp-cmap
   ```

3. Create a match statement for the RACL.

   ```
   device(config-classmap)# match access-group racl-1
   ```

4. Access global configuration mode.

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

5. Create a policy map.

   ```
   device(config)# policy-map cp-pmap
   ```

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

   ```
   device(config-policymap)#class cp-cmap
   ```

7. Define the action on the classified traffic.

   ```
   device(config-policymap-class)# police cir 121212
   ```

   This step specifies the committed information rate of 121,212 bps.

8. Access global configuration mode.

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

9. Access control-plane configuration mode.

```
device(config)# control-plane
```

10. Apply the policy map to the CPU control plane.

```
device(config-control-plane)# service-policy in cp-pmap
```

11. Access privileged EXEC mode.

```
device(config-control-plane)# end
```

12. Verify the configuration.

```
device# show policy-map control-plane
Ingress Direction :
Policy-Map cp-pmap
  Class cp-cmap
   matches 0 packets 0 bytes
   Police cir 121212
    Stats:
       Operational cir:121212 cbs:0 eir:0 ebs:0
       Conform Byte:0 Exceed Byte:0 Violate Byte:0
```

Example

The following example shows the steps in the previous configuration.

```
device# configure terminal
device(config)# class cp-cmap
device(config-classmap)# match access-group racl-1
device(config-classmap)# exit
device(config)# policy-map cp-pmap
device(config-policymap)# class cp-cmap
device(config-policymap-class)# police cir 121212
device(config-policymap-class)# exit
device(config-policymap)# exit
device(config)# control-plane
device(config-control-plane)# service-policy in cp-pmap
device(config-control-plane)# end
device# show policy-map control-plane
```

Egress ACL Rate Limiting

> **Important**
> The egress ACL rate limiting feature is not supported on SLX 9150.

In general, you configure a rate limit profile by creating a QoS policy and associating it with a QoS class map. Under this rate limit profile, you configure the expected rate. Egress ACL rate limiting supports the following scheduling and shaping rules:

- Eight priorities for each rate limit profile and strict priority for traffic in a rate limit profile (VLAN, BD, ACL).
- Traffic with the same priority destined to the same port but belonging to a different rate limit profile is treated as round robin.

- Traffic with different priorities destined to the same port but belonging to a different rate limit profile is treated as LOW (0-3) and HIGH (4-7). The device supports strict priority between HIGH and LOW.
- All priority flows for a rate limit profile are under the same shaper. Each rate limit profile shaping rate is aggregated for all priorities.
- Flows from all class maps, including normal traffic, are rate limited by the configured port rate.

Table 11: Supported network topology for unicast ingress and egress traffic

Ingress packets	Egress packets
Layer 2 port, VLAN or BD	Layer 2 port, VLAN or BD
VPLS	Layer 2 port, VLAN or BD

> **Note**
> A VPLS or VxLAN egress topology is not supported. Multicast traffic is not rate limited by the RL profile but it is rate limited by the port rate limit.

Egress ACL Rate Limiting Considerations

> **Important**
> The egress ACL rate limiting feature is not supported on SLX 9150, SLX 9250, Extreme 8520, and Extreme 8720 devices.

SLX 9540 and SLX 9640

Consider the following when you configure egress ACL rate limiting:

- Egress rate limit supports packets that are received at a physical port but transmitted on a different physical port. If the packets are received and transmitted on the same physical port, you should deploy ingress rate limit.
- If the multiple VLANs are on the same ingress port belonging to the same bridge domain and egress ACL rate limiting is configured to rate limit one of the VLANs, the traffic of all VLANs is rate limited. As a workaround, add a matching source or destination MAC address with the VLAN in the ACL.
- Ingress ACL rate limit and egress ACL rate limit do not work together on the same flow of traffic.
- To redirect traffic that requires egress rate limit to new sets of VOQ, the devices uses PMF and VOQ resources that limits the number of supported policies.
  - The following system scale numbers are based on the limitation of VOQ connector and PMF entries:

Table 12: Scale numbers

Device type	Maximum RL class per tower	Maximum RL class per system
SLX 9540	2,000	2, 000

- The PMF is configured to filter the certain types of Layer 2 or VPLS traffic and reroute them to the VOQs for egress rate limiting.

Table 13: PMF filter keys

Filter key	Description
Port	Egress port of the Layer 2 packets.
VLAN	VLAN-based rate limit: The VLAN in which the port participates.
Bridge Domain	Bridge domain-based rate limit: The bridge domain in which the port's logical interface participates.
VLAN Tag	ACL-based rate limit: The configuration of either the single tag or the double tags of the traffics on the egress port is required with the following match: • For a VLAN, the single tag must match the VLAN to which the port participates. • For a bridge domain, the associated logical interface must exist first.
Ethernet Type	ACL-based rate limit: Ethernet type of Layer 2 packets. It is optional, but assists in restricting packets for egress rate limiting.
Source MAC	ACL-based rate limit: Source MAC address of Layer 2 packets. It is optional, but assists in restricting packets for egress rate limiting.
Destination MAC	ACL-based rate limit: Destination MAC address of Layer 2 packets. It is optional, but assists in restricting packets for egress rate limiting.

- For egress rate limiting, enable the L2-Egress-RateLimit TCAM profile and then reboot the device after making this configuration change. A maximum of 2,048 egress rate limiting is supported system wide.

SLX 9740

- SLX 9740 supports policing on the egress side and Egress TCAM helps to identify the flow. Egress RL supports CBS, EIR and EBS configurations on SLX 9740.

- The Maximum value for the CBS is 255MB.

- Egress TCAM has support for only 1024 entries for egress ACL lookup which is also shared with Security ACL. This limits the maximum scale number supported on Flow based Egress Rate-limiting to 1024.

- Egress Rate limiting cannot be used along with BD configuration. Rate limiting at egress works for known unicast and BUM traffic. But due to hardware issue, BUM traffic gets corrupted. If BUM traffic hits the Egress RL, the VLAN ID of the out going frame will be replaced with Bridge Domain ID.

- Egress metering does not consider *inter-frame-gap* (IFG)/*inter-packet-gap* (IPG) in SLX 9740. This is a hardware limitation. Hence, a deviation is seen between the operational CIR and the on-wire rate. This deviation is more for smaller sized packets. For packets with size more than 1024 bytes, the deviation is less than 3%.

- Supported functionality includes the following:
  - Per port egress rate limit
  - Per flow egress rate limit for Layer 2 VLAN, bridge domain, and ACL
  - Per flow egress strict priority for Layer 2 VLAN, bridge domain, and ACL

- ◦ VPLS ingress direction
- ◦ Per flow egress rate limit for Layer 3 IPv4 and ACL
- Unsupported functionality includes the following:
  - ◦ WFQ
  - ◦ Per-priority shaping
  - ◦ VPLS egress direction
  - ◦ VXLAN (in and out direction)

Configure Egress ACL Rate Limiting

Before You Begin

Before you begin, perform the following tasks:

- For VLAN-based or ACL-based rate limiting, create a VLAN and bind it to a port.
- For bridge domain-based rate limiting, configure the logical interfaces, create the pseudowire for VPLS, and create the bridge domain.

> **Note**
>
> For egress rate limiting, each bridge domain can have only one logical interface from one egress port.

- Create an ACL for the VLAN and logical interfaces.

About This Task

> **Important**
>
> The egress ACL rate limiting feature is not supported on SLX 9150.

Procedure

1. Create a VLAN, BD, or ACL class map and bind the associated VLAN, bridge domain, or ACL to it. The following example creates a VLAN class map and binds VLAN 1000 to it.

   ```
   device(config)# class-map eVLAN100
    device(config-classmap)# match vlan 100
   device(config-classmap)# exit
   ```

   > **Note**
   >
   > To configure BFD, you must configure the timing and interval parameters on each interface. When two adjacent interfaces with BFD are configured, they negotiate the conditions for determining if the connection between them is still active.

2. Create a policy map and bind the class map to it.

   ```
   device(config)# policy-map epol
    device(config-policymap)# class eVLAN100
   ```

   A policy map is used to apply the policer and QOS attributes to a particular interface. Each policy map can have different class maps. Each class map in the policy map can be associated with separate policing and QOS parameters.

3. Configure the scheduling scheme or policing rate.

   ```
    device(config-policymap-class)# police cir 800000000
   device(config-policymap)# exit
   ```

4. Bind the policy map to a port.

```
device(config)# interface ethernet 3/1
device(conf-if-eth-3/1)# service-policy out epol
device(conf-if-eth-3/1)# end
```

A policy map is attached to the interface for the ingress or egress direction with the **service-policy** command. For egress rate limit, only an egress port is supported.

For each egress port, only one policy map is allowed. However, each policy map can have multiple class maps. To configure a port with different shaping criteria for multiple traffic streams, configure a policy map with multiple classes and configure a different criteria for each class.

Example

This example summarizes the steps in the procedure.

```
class-map eVLAN100
 match vlan 100
!
class-map eVLAN200
 match vlan 200
!
class-map eBDpol
 match bridge-domain 100

policy-map epol
   class eVLAN100
   police cir 800000000
   !
    class eBDpol
    police cir 1000000000
    !
    class eVLAN200
    police cir 500000000
  !
 interface Ethernet 3/1
  service-policy out epol
 switchport
 switchport mode trunk-no-default-native
 no shutdown
 logical-interface ethernet 3/1.100
  vlan 100
 !
 logical-interface ethernet 3/1.200
  vlan 200
 !
!
```

Configuring IPv4 Egress Rate Limiting

Support for IPv4 egress rate limiting is available for SLX 9740 devices on the ethernet, port-channels and VE interfaces. Support for ingress rate limiting is also available for VE interfaces. Filtering can be done on Source IP, Destination IP, IPv4 TOS, IPv4 Protocol, L4 Source Port, and L4 Destination Port.

Before You Begin

Before you begin, perform the following tasks:

- Create a Policy map with the appropriate Class Map and Police configurations.

> **Note**
> IPv4 egress rate limiting is only supported on SLX 9740 devices.

Procedure

1. Create an IPv4 Extended ACL.

```
SLX (config)# ip access-list extended ipv4acl
    seq 100 permit ip 10.10.10.10 255.255.255.0 10.10.10.20 255.255.255.0
```

2. Create a class map.

```
SLX (config)# class-map class-map_1
SLX (config-classmap)# match access-group ipv4acl
```

3. Create a policy map.

```
SLX (config)# policy-map policy-map_1
SLX (config-policymap)#
```

4. Apply the class map to the policy map.

```
SLX (config-policymap)# class class-map_1
SLX (config-policymap-class)#
```

5. Create a Police entry for this policy map.

```
SLX (config-policymap-class)# police cir 300000 eir 300000000
SLX (config-policymap-class)#
```

6. Apply the policy map.

- Apply to the interface.

```
SLX (config)# int eth 0/1
SLX (config-if-eth-0/1)# service-policy out policy-map_1
```

- Apply to the port-channel.

```
SLX (config)# interface port-channel 10
SLX (config-port-channel-10)# service-policy out policy-map_1
```

- Apply to the VE interface in the ingress direction

```
SLX (config)# interface ve 100
SLX (config-ve-100)# service-policy in policy-map_1
```

- Apply to the VE interface in the egress direction

```
SLX (config)# interface ve 100
SLX (config-ve-100)# service-policy out policy-map_1
```

Example

The following example summarizes the configuration of IPv4 rate limiting on an ethernet interface for egress traffic.

```
SLX(config)# ip access-list extended ipv4acl
SLX(conf-ipacl-ext)# seq 100 permit ip 10.10.10.10 255.255.255.0 10.10.10.20 255.255.255.0
SLX(conf-ipacl-ext)# exit
SLX(config)# class class-map_1
SLX(config-classmap)# match access-group ipv4acl
SLX(config-classmap)# exit
SLX(config)# policy-map policy-map_1
SLX(config-policymap)# class class-map_1
SLX(config-policymap-class)# police cir 300000 eir 300000000
SLX(config-policymap-class)# exit
SLX(config-policymap)# exit
SLX(config)# int eth 0/1
SLX(conf-if-eth-0/1)# service-policy out policy-map_1
SLX(conf-if-eth-0/1)# exit
SLX(config)#
```

The following example summarizes the configuration of IPv4 rate limiting on a port-channel for egress traffic. For the configuration of the *policy-map* refer to the previous section.

```
SLX (config)# interface port-channel 10
SLX (config-port-channel-10)# service-policy out policy-map_1
```

This example summarizes the configuration of IPv4 rate limiting on a VE interface for ingress traffic.

```
SLX (config)# interface ve 100
SLX (config-ve-100)# service-policy in policy-map_1
```

This example summarizes the configuration of IPv4 rate limiting on a VE interface for egress traffic.

```
SLX (config)# interface ve 100
SLX (config-ve-100)# service-policy out policy-map_1
```

Configure Statistics Support for Egress ACL Rate Limit

In SLX9540 and SLX 9640, the egress ACL rate limit feature supports statistics when the `COUNTER-PROFILE-5` counter profile is enabled. The MPLS tunnel statistic also uses this counter profile.

About This Task

> **Important**
> - This procedure requires a system reload to activate the profile configuration. Perform this procedure during a maintenance window so that the new counter can be activated without interrupting normal network services.
> - The egress ACL rate limiting feature is not supported on SLX 9150.

Procedure

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

   ```
   device# configure terminal
   ```

2. Access hardware configuration mode.

    ```
    device(config)# hardware
    ```

3. Configure the counter profile.

    ```
    device(config-hardware)# profile counters counter-profile-5
    ```
    The following message is displayed.
    ```
    Warning: To activate the new profile config, please run
    'copy running-config startup-config' followed by 'reload system'.
    ```
    You run these commands in steps 5 and 6.

4. Access privileged EXEC mode.

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

5. Save the running configuration to the startup configuration file.

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

6. Activate the profile configuration.

    ```
    device# reload system
    ```

7. After the system reloads, verify the configuration.

    ```
    device# show hardware profile counters counter-profile-5
     switch type: BR-SLX9540-4


                          current COUNTERS profile:    COUNTERS-PROFILE-5
                                  InLIF - HitCount:     16384
                                   InL4 - FwdDrop:      8192
                                OutLIF - HitCount:      16384
                        Egress ACL - VOQ Counters:      8192

    _____
    ```

Example

This example summarizes the steps in the procedure.
```
device# configure terminal
device(config)# hardware
device(config-hardware)# profile counters counter-profile-5
device(config-hardware)# exit
device# copy running-config startup-config
device# reload system
device# show hardware profile counters counter-profile-5
```

Egress ACL Rate Limiting Show Commands

You can use show commands to display information about egress ACL rate limiting.

➡ **Important**
The egress ACL rate limiting feature is not supported on SLX 9150.

You can run show commands from any level of the CLI.

show policy-map

Displays data traffic information. For example:

```
device# show policy-map int e 4/1
Egress Direction :
 Policy-Map eACLpol
    Class eACL100
      matches 0 packets 0 bytes
```

```
      Police cir 1000000000
        Stats:
          Operational cir:1000000000 cbs:0 eir:0 ebs:0
          Conform Byte:161527005348 Exceed Byte:0 Violate Byte:1904690363136
```

In this example, the Conform Byte field displays the data traffic that is queued and forwarded. The Violate Byte field displays the data traffic that is dropped due to the rate limit.

`show running-config class-map`

Displays class map information and match criteria. For example:

```
device# show running-config class-map
class-map eVLAN100
 match vlan 100
 !
class-map eVLAN200
 match vlan 200
 !
class-map eBDpol
 match bridge-domain 100
```

`show running-config policy-map`

When used with **`show policy-map`**, displays rate-limiting policies in the policy maps and policer counters. For example:

```
device# show policy-map
Number of policy maps : 1
Policy-map epol
  Bound To: Eth 3/1(out)

device# show running-config policy-map
policy-map epol
   class eVLAN100
   police cir 800000000
   !
    class eBDpol
    police cir 1000000000
    !
    class eVLAN200
    police cir 500000000
  !
```

TTL 0/1 Rate Limiting

IPv4 and IPv6 frames with a TTL of 0 or 1 are primarily data frames. These frames are rate limited so that they do not congest the queues and prevent other control protocol frames from reaching the CPU.

SLX 9150 and SLX 9250 rate limiting

Frames with a Time to Live (TTL) value of 0 or 1 (TTL 0/1) are diverted to a separate queue (queue number 0, with a rate limit of 1,000 pps), with further rate limiting of 40 Kbps and TTL-1 frames to 2,000 Kbps provided by the `ctrl-classifier` policers. Because the TTL 0/1 trap queue is separate from other control protocol queues, the other control protocols are not blocked.

SLX 9540 and SLX 9640 rate limiting

The TTL 0/1 rate limit of 40 Kbps is provided by the `ctrl-classifier` policers. The IPv4 and IPv6 TTL-1 and IPv6 TTL-0 are rate limited to 40 Kbps in all TCAM profiles. IPv4 TTL-0 is rate limited to 1

Mbps in the Default, App-Telemetry, and L2-rate Limit TCAM profiles and to 40 Kbps in the remaining TCAM profiles.

Because a low TTL 0/1 rate limit is required to prevent flapping in scale protocol tests and scenarios, the rate limit is hard-coded in the driver software. You cannot change it.

Subnet Trap Rate Limiting

When the destination IP address of an ingress Layer 3-routed frame is not present in the forwarding routing table, the frame is trapped to the CPU (subnet trap frame) to generate an ICMP message. An `ICMP destination host unreachable` message is returned to the sender, informing the source host that the destination address is unreachable. If not rate-limited, the subnet trap frames can prevent other important control frames from reaching the CPU.

Rate limits vary among SLX devices because the devices have different hardware architectures.

SLX 9150 and SLX 9250 subnet trap rate limiting

By default, IPv4 and IPv6 subnet trap frames are diverted to a separate queue (queue number 9). You can use the CLI to further limit the rate of IPv6 subnet trap frames by configuring a Committed Information Rate (CIR) and a Committed Burst Size (CBS). Valid CIR values range from 22 Kbps through 1,200 Kbps.

For more information, see Committed Information Rate and Committed Burst Size on page 14.

SLX 9540 and SLX 9640 subnet trap rate limiting

By default, IPv4 and IPv6 subnet trap frames are rate-limited to 10,000 Kbps. You can use the CLI to further limit IPv4 and IPv6 subnet trap frames by configuring a CIR and a CBS. Valid CIR values range from 400 Kbps through 10,000 Kbps.

Subnet trap rate limiting commands

To configure the CIR and the CBS, use the **ip subnet-rate-limit** command with the **cir** and **cbr** options. For example:

```
device# configure terminal
device(config)# control-plane
device(config-control-plane)# ip subnet-rate-limit cir 134 cbr 34
```

> **Note**
> Applying a subnet trap rate limit can increase the time it takes for conversational Neighbor Discovery to converge.

To display the configured CIR and CBS, use the **show running-config control-plane ip subnet-rate-limit** command.

To display the number of packets and bytes per second for IPv4 and IPv6 subnet traps, use the **show ip subnet-rate-limit stats** command.

Rate Limiting Scalability

- 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: 1,024
- Number of class maps in a policy map: 4,096
- Number of policers that are reserved for port channels: 1,215
- Number of supported policers in a system: 32,712 (SLX 9640 and SLX 9540)
- Number of supported policers in a system: 8,000 (SLX 9150 and SLX 9150)
- The number of supported port, BUM, ACL, VLAN, and bridge domain-based rate limiting entries in a chip depends on the configured TCAM profile and the number of resources available for the subtype.

Configuring traffic policing

Follow these tasks to configure traffic policing.

- Configuring a class map using an ACL on page 51
- Configuring a policy map on page 52
- Configuring port-based traffic policing on page 53
- Configuring ACL-based rate limiting on page 54

Configuring a class map using an ACL

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

Procedure

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.

Procedure

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.

Before You Begin

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

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode

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

3. Attach an input or output policy map.

```
device(conf-if-eth-2/2)# service-policy out policy_2
```

For rate limiting ingress traffic just replace the **out** keyword with **in** and use an ingress policy map.

4. Return to privileged EXEC mode.

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

5. Verify the configuration.

```
device# show policy-map interface ethernet 2/2 out

Egress Direction :
 Policy-Map policy_2
    Class default
      Police cir 4000000 cbs 50000 eir 800000 ebs 400000
        Stats:
          Operational cir:4006912 cbs:50000 eir:0 ebs:400000
          Conform Byte:0 Exceed Byte:0 Violate Byte:0
```

> **Note**
> Egress policer is not supported on the SLX 9150 device. The **police** command affects only the CIR.

Port-based traffic policing configuration example

```
device# configure terminal
device(config)# interface ethernet 2/2
device(conf-if-eth-2/2)# service-policy out policy_2
device(conf-if-eth-2/2)# end
device# show policy-map interface ethernet 2/2 out
```

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.

Procedure

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.

Before You Begin

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

Procedure

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 1/2
```

12. Bind the policy map to the port.

```
device(conf-if-eth-1/2)# service-policy in policyAclFilter
2015/04/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4
access list
acl1 configured on interface Ethernet 1/2 at Ingress by FbQos_9_11.
```

13. Return to privileged EXEC mode.

```
device(conf-if-eth-1/2)# end
```

14. 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 1/2(in)
```

15. Save the configuration.

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

ACL-based protection against TCP SYN 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 1/2
device(conf-if-eth-1/2)# service-policy in policyAclFilter
device(conf-if-eth-1/2)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

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.

Procedure

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.

Before You Begin

A TCP RST matching ACL has been configured.

Procedure

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 1/2
```

12. Bind the policy map to the port.

```
device(conf-if-eth-1/2)# service-policy in policyAclFilter
2015/04/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4
access list
acl1 configured on interface Ethernet 1/2 at Ingress by FbQos_9_11.
```

13. Return to privileged exec mode.

```
device(conf-if-eth-1/2)# end
```

14. 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 1/2(in)
```

15. 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 1/2
device(conf-if-eth-1/2)# service-policy in policyAclFilter
```

```
device(conf-if-eth-1/2)# 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.

Procedure

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.

Before You Begin

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

Procedure

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. Enter interface configuration mode.

```
device(config)# interface ethernet 1/2
```

12. Bind the policy map to the port.

```
device(conf-if-eth-1/2)# service-policy in policyAclFilter
2015/04/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4
access list
acl1 configured on interface Ethernet 1/2 at Ingress by FbQos_9_11.
```

13. Return to privileged EXEC mode.

```
device(conf-if-eth-1/2)# end
```

14. 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 1/2(in)
```

15. 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 1/2
device(conf-if-eth-1/2)# service-policy in policyAclFilter
device(conf-if-eth-1/2)# 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.

Procedure

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

Before You Begin
You have configured an extended Layer 3 UDP ACL.

Procedure

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 1/2
```

12. Bind the policy map to the port.

```
device(conf-if-eth-1/2)# service-policy in policyAclFilter
2015/04/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device, IPv4
access list
acl1 configured on interface Ethernet 1/2 at Ingress by FbQos_9_11.
```

13. Return to privileged EXEC mode.

```
device(conf-if-eth-1/2)# end
```

14. 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 1/2(in)
```

15. 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 1/2
device(conf-if-eth-1/2)# service-policy in policyAclFilter
device(conf-if-eth-1/2)# 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.

Procedure

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 1/2
```

20. Bind the policy map to the port.

```
device(conf-if-eth-1/2)# service-policy in policyAclFilter
2015/04/02-14:13:31, [SSMD-1405], 2511, SW/device | Active | DCE, INFO, device,
IPv4 access list acl1 configured on interface Ethernet 1/2 at Ingress by FbQos_9_11.
```

21. Return to privileged EXEC mode.

```
device(conf-if-eth-1/2)# 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: Et 1/2(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 1/2
device(conf-if-eth-1/2)# service-policy in policyAclFilter
device(conf-if-eth-1/2)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

Storm Control for 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.

Use BUM storm control to limit the amount of BUM ingress traffic globally or on a specified physical interface. All ingress traffic in excess of the configured rate is discarded.

Consider the following when you configure storm control:
* BUM storm control applies only to ingress traffic.
* The device supports both global and interface-level BUM storm control.
* BUM storm control and input service-policy features can coexist on an interface.
* It can only be configured on physical interfaces in SLX 9540, SLX 9640, SLX 9150 and SLX 9250.
* For LAG ports, BUM RL needs to be attached to each LAG member port on SLX 9150, SLX 9250, SLX 9540 and SLX 9640.
* On SLX 9740, BUM RL is applied directly on LAG. If there are multiple LAG members on one line card, the CIR on that Line card is limited to configured CIR. The CIR will not be applied per member port, but on the LAG. Hardware creates and maintains policer on each core separately, hence there will be Rate-limiting applied per core, and in case where Rate-limiting is done on Port Channel and there are Port-Channel members across cores / Units, we will see separate Rate-limiting applied. It is advisable to have the port Channel members to be part of the same core, to have uniform Rate-limiting.
* A Single-rate Two Color Marking (SrTCM) scheme is used.

- Metering is performed on the packet size as received on the wire, including IPG (inter-packet gap) and preamble, but ignoring CRC (cyclic redundancy check).
- BUM rate limiting is ineffective if the BUM traffic is also classified by an ACL.
- The configured rate in bits per second (bps) is rounded up to next achievable rate.
- On SLX 9150 and 9250 only hit counters are supported, hence violated counters will not be updated.
- In SLX 9540, 9640 and 9740 only FWD and DROP counters are supported.
- ∘ **Conformed**: Shows FWD packets including green and yellow color packets.
  - ∘ (SLX 9640 and SLX 9540 only) **Violated**: Shows DROP packets including red color packets.
  - ∘ (SLX 9640 and SLX 9540 only) **Exceed**: Is always ZERO.
- In SLX 9540, 9640, FWD and DROP counters must use a counter profile other than default.
- (SLX 9640 and SLX 9540 only) When storm control is configured at the interface level, you can specify whether to shut down an interface if the maximum defined rate is exceeded within a 10-second sampling period. When a port is shut down, you receive a log message. From global configuration mode, shutdown is not supported.

Configuring Storm Control on an Ethernet Interface

You can control a broadcast, unknown unicast, and multicast (BUM) traffic storm by limiting the inbound traffic on an interface.

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

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

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

3. Specify a traffic limit for broadcast traffic on the interface.

```
device(conf-if-eth-2/2)# storm-control ingress broadcast limit-bps 400000
```

This example controls the inbound broadcast traffic, limiting the rate to 400000 bits per second (bps).

4. Specify a traffic limit for unknown-unicast traffic on the interface.

```
device(conf-if-eth-2/2)# storm-control ingress unknown-unicast limit-bps 50000000
```

This example controls the inbound unknown-unicast traffic, limiting the rate to 50000000 bps.

5. Specify a traffic limit for multicast traffic on the interface.

```
device(conf-if-eth-2/2)# storm-control ingress multicast limit-percent 3
```

This example controls the inbound multicast traffic, limiting the rate to 3% of traffic.

6. Return to privileged EXEC mode.

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

7. Verify the storm control configuration.

```
device# show running-config interface ethernet 2/2 | 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
```

Configuring storm control globally on the device

About This Task

Perform the following steps to configure BUM storm control globally on the device.

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

2. Issue the storm control ingress command to set a traffic limit for broadcast traffic.

```
device(config)# 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).

3. Issue the storm control ingress command to set a traffic limit for multicast traffic.

```
device(config)# storm-control ingress multicast limit-bps 400000
```

   In this example you set a control on the inbound multicast 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.

```
device(config)# 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. Return to privileged EXEC mode.

```
device(config)# end
```

6. Verify the storm control configuration.

```
device# show running-config storm-control
storm-control ingress broadcast limit-bps 400000
storm-control ingress multicast limit-bps 400000
storm-control ingress unknown-unicast limit-bps 50000000
```

7. Save the configuration.

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

BUM storm control configuration example

```
device# configure terminal
device(config)# storm-control ingress broadcast limit-bps 400000
device(config)# storm-control ingress multicast limit-bps 400000
device(config)# storm-control ingress unknown-unicast limit-bps 50000000
device(config)# end
```

```
device# show running-config 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 Unicast and Multicast Traffic

QoS considerations vary for each type of traffic.

> **Note**
> QoS unicast and multicast traffic is supported only on devices based on the DNX chipset family. For more information, see Supported Hardware.

In the Traffic Manager, QoS for unicast traffic follows the process shown in the following figure.
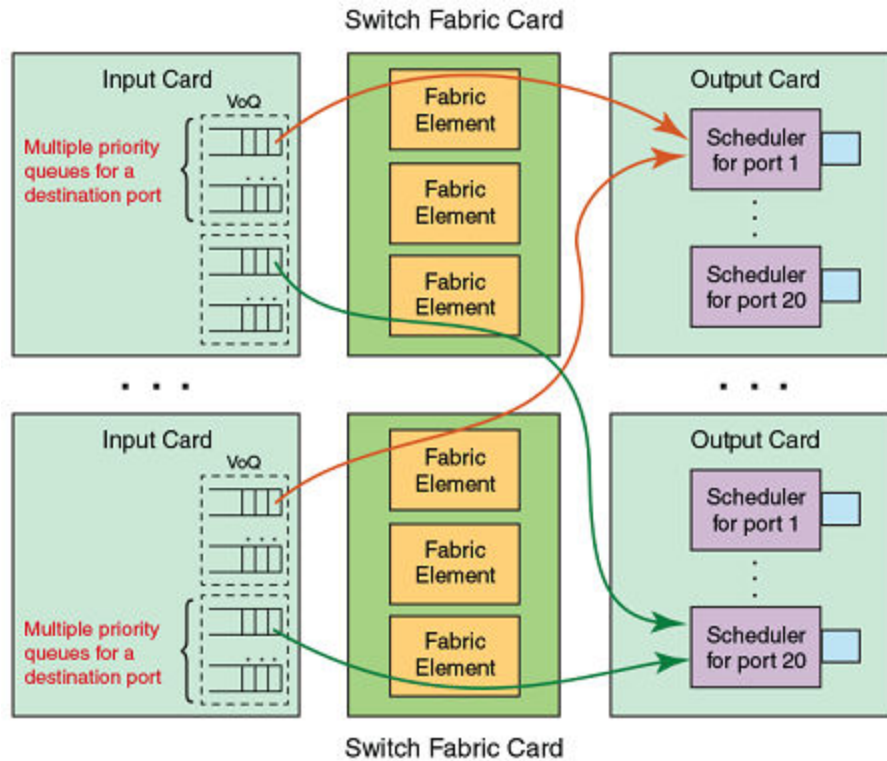
Figure 3: QoS for unicast traffic

- The input is buffered through Virtual Output Queues (VOQ) with output port-driven scheduling.
- Each ingress Traffic Manager maintains a set of eight distinct priority queues for every output port on the system. Incoming packets are enqueued to a VOQ corresponding to the destination output port and classified with an internal priority.
- Packets are dequeued by an output port on the egress card when the output port is ready to send a packet.
- Switch fabric messaging is used to maintain a tight coupling between ingress and egress cards.
- Egress Traffic Manager dequeues packets from appropriate VOQs by sending VOQ transmission credits to the respective ingress traffic managers.

Consider the following when you manage multicast traffic:

- There are four fabric multicast queues (FMQs) for multicast traffic.
- The Traffic Manager maps incoming packets to these queues based on the traffic class or drop precedence received from the packet processor (PP).
- The ingress Traffic Manager pushes multicast traffic to the fabric by strict priority (SP) or by a mix of SP and weighted priority.
- There is a CLI to configure ingress multicast shaping. By default, it is opened to the maximum rate of the tower.
- The egress Traffic Manager maps these multicast packets to two egress queues (EGQ).
- EGQs share memory from a 3MB pool.

QoS on the SLX 9150 and SLX 9250 Devices

QoS on the devices consists of the following:

- 32MB buffer
- 128K 256-byte cells
- 10 UC and 10 MC queues, however only 8 of each are supported

The devices support the following:

- Port shaping
- Priority shaping
- Strict priority (SP)
- Weighted round robin (WRR)
- 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

Cell packing is not supported

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 14: Default DSCP mappings

DSCP IP precedence	User priority
0–7	0
8–15	1
16–23	2
24–31	3
32–39	4

Table 14: Default DSCP mappings (continued)

DSCP IP precedence	User priority
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.

Weighted random early detection

Weighted random early detection (WRED) is a traffic control feature that uses IP precedence to determine how it treats or drops traffic.

On the device, queues are provided to buffer traffic levels that exceed the bandwidth of individual ports. For each output port, a set of eight priority queues is allocated. When traffic exceeds the bandwidth of a port, packets are dropped randomly as long as the congestion persists. Under these conditions, traffic of greater priority can be dropped instead of traffic with a lesser priority.

Instead of being subject to random selection, you can configure a device to monitor traffic congestion and drop packets according to a WRED algorithm. This algorithm enables the system to detect the onset of congestion and take corrective action. In practice, WRED causes a device to start dropping packets as traffic in the device starts to back up. WRED provides various control points that can be configured to change a system's reaction to congestion. The following variables are used when calculating whether to drop or forward packets:

- Queue-Size—The user-configurable queue size.
- Current-Q-Size—The current size of the queue as calculated on the device.
- Min-Q-Threshold—The queue threshold in percentage below which all packets are accepted. This variable is user configurable.

- Max-Q-Threshold—The queue threshold in percentage above which all packets are dropped. This variable is user configurable.
- Drop-Probability—The maximum drop probability when the Queue-Size is at Max-Q-Threshold. This variable is user configurable.

How WRED works

The following WRED operation graph describes the interaction of the previously described variables in the operation of WRED. When a packet arrives at a device, if Current-Q-Size is below the configured Min-Q-Threshold, the packet is accepted. If the Current-Q-Size is above the Max-Q-Threshold, the packet is dropped. If the Current-Q-Size falls between the Min-Q-Threshold and Max-Q-Threshold, the packet is dropped according to the calculated probability Pdrop described in the following "Calculating packets that are dropped " section.
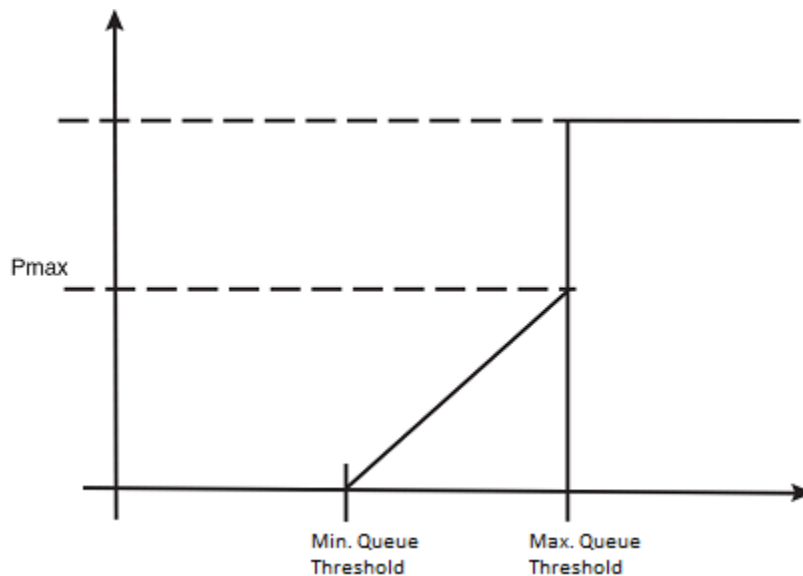
Figure 4: WRED operation graph

Calculating avg-q-size

The algorithm first calculates the `avg-q-size` through the following equation.

```
avg-q-size = [(1 - Wq) × Statistical Average-Q-Size)] + (Wq × Current-Q-Size)
```

The user-configured W_q value is instrumental to the calculation and can be:

- equal to the statistical average queue size (W_q == 0), or
- equal to the current queue size (W_q == 1) or
- be between 0 and 1 (0 < W_q < 1).

Lower Wq values cause the `avg-q-size` to lean towards the statistical average queue size, reducing WRED's sensitivity to the current state of the queue and thus reducing WRED's effectiveness. On the other hand, higher W_q values cause the `avg-q-size` to lean towards the instantaneous queue size, which exposes WRED to any change in the instantaneous queue size and thus may cause WRED to overreact in cases of bursts. Thus, the value of W_q should be carefully chosen according to the application at hand.

Calculating packets that are dropped

The `Pdrop` value, as calculated in the following equation, is the probability that a packet will be dropped in a congested device.

```
Pdrop = Drop-Probability * (Current-Q-Size/Queue-Size - Min-Q-Threshold) / (Max-Q-
Threshold - Min-Q-Threshold)
```

Applying WRED

You configure Min-Q-Threshold, Max-Q-Threshold and Drop-Probability for a WRED profile, and apply the WRED profile to a device per-traffic class and per-drop-precedence.

Link Level Flow Control

Link level flow control (LLFC) is a way to alleviate system congestion by pausing data transmission.

When a receiving device is congested, it communicates with the transmitting device by sending a PAUSE frame that instructs the device to stop data transmission for a specific time. This feature is available for each port in all front ports and applies to all traffic on the link. Extreme supports the generation (Tx) and reception (Rx) of PAUSE frames for each physical interface or port channel.

By default, LLFC reception is enabled.

Refer to Configuring link level flow control for the steps to configure LLFC.

> **Note**
> Pause and PFC LLFC are supported only on devices based on the XGS chipset family. For more information, see Supported Hardware.

- As a best practice, stop the traffic on the interface before you configure LLFC on an interface or port channel.
- A flow control configuration on an interface applies to all traffic received over the interface.
- Flow control thresholds are tuned through default maximum queue sizes of ingress BD consumption.
- A service policy cannot be applied to an interface if a flow control configuration exists. The reverse is also true.

Transient Buffer Congestion Detection

Transient buffer congestion can cause the dropping of packets and its cause is difficult to determine without a way to detect and log its occurrences. The SLX-OS device allows you to configure the monitoring of discarded packets for all VOQs that map to physical ports or multicast VOQs. You can also monitor the discarded and deleted packets on the device.

> **Note**
> - VOQs that are used by CPU traffic are not monitored.
> - Transient buffer congestion detection is supported only on devices based on the DNX chipset family. For more information, see Supported Hardware.

When traffic flows from the ingress packet processor to the ingress TM, it is buffered in the destination VOQ which marks the destination port of the packet. If this VOQ does not receive enough credits from the egress TM, the VOQ buffer overflows and the packets are discarded or deleted (aged out) at the ingress TM.

By default, monitoring of these packets is not enabled. To enable logging, you must configure a threshold limit for the discarded or deleted packet count. The discarded packet count of a VOQ is logged based on the configured threshold value. You can also set the time interval that a RASlog message is generated if the discard count threshold limit is reached.

> **Note**
> To restrict logging other than the logging interval, the number of logs recorded within the logging interval is also set to a maximum of 50 logs for each line card. Within the expiry of logging time interval, only 50 logs are generated for 50 VOQs reporting the maximum discarded packets beyond the threshold limit.

When the discarded packet count exceeds the threshold value, an INFO severity RASlog is generated with information of the interface, ingress tower, ingress core and configured threshold limit, similar to the following example.

```
device# show logging raslog reverse count 100 | inc SYSD-1003
2017/01/05-11:03:59, [SYSD-1003], 793, M2 | Active | DCE, WARNING, SLX,  TM threshold
2017/01/05-11:00:34, Discarded packets 930587727, interface 3/2 prio 0 on device 3.1.0
```

The RASlog is only generated if the logging-time interval has expired. If the discarded packet count threshold is constantly exceeded, the RASlog is only generated after the configured logging-time interval expires.

The time interval to monitor VOQ discarded packet counts is four minutes. Within every four minutes, the discarded packet count for all VOQs of all towers are monitored. If any VOQ discarded packet count has exceeded the threshold, a breach instance is recorded and a RASlog is generated. If the current recorded breach instance is before the logging time expiry, the RASLOG is not generated.

Although the configured RASlog interval is between 10-2880 minutes and the VOQ statistic monitoring time is four minutes, the logging time set is adjusted to the nearest multiple of 4. For instance, if you configure the monitoring time to 10 minutes, the log for the VOQ is generated every 12 minutes.

The SLX-OS device also allows the monitoring of the total discarded packet count and deleted packet count on the device. These are device counters and are monitored every five seconds. You can configure different threshold limits for the discarded and deleted packet counts. If a RASlog message for the TM device statistics is generated within the logging interval, the message includes the time stamp, the number of discarded packets, and ingress slot, tower and core, similar to the following example.

```
device# show logging raslog reverse count 10
2017/01/05-10:56:58, [SYSD-1005], 788, M2 | Active | DCE, WARNING, SLX,  TM threshold,
Tail discarded packets 20734462 on device 3.1.1.
```

Explicit Congestion Notification

Explicit Congestion Notification (ECN) allows routers and switches to signal traffic end points of traffic congestion at their ends without dropping any frames. On receipt of such a signal, the end points can adjust their transmission rate. This ensures that there is no change in existing latency or jitter that would have caused by dropped packets and their subsequent re-transmission.

Without Explicit Congestion Notification, any network congestion is signalled by packet loss. Protocols, such as TCP (that respond to congestion), will reduce the transmission rate by shrinking the connection's congestion window. To recover from congestions, these protocols have to re-transmit the dropped packets. This dropping and subsequent re-transmission of packets, results in loss in throughput and increase in latency and jitter.

Traffic congestion is signalled by the router or switch to the end points by modifying the two least significant bits in the IP header's *Traffic Class* field. The following code points are used:

* 00 - Non ECN capable transport (Non-ECT)
* 01 - ECN capable transport (ECT-0)
* 10 - ECN capable transport (ECT-1)
* 11 - Congestion Encountered (CE)

To signal to an endpoint that congestion is seen, the switch/router sends a CE message to the concerned end point. The end point that receives the CE message echoes this message to its other endpoint. They, in coordination with each other, adjust their transmission rates, to mitigate the congestion.

Explicit Congestion Notification only applies to unicast frames.

Explicit Congestion Notification (ECN) is implemented as an extension of the existing Random Early Detection (RED) function.

Hardware Buffer Configuration for Lossless Traffic

Buffers for lossless and lossy traffic are managed differently. Lossy traffic buffers are managed by the egress ports and lossless traffic buffers by the ingress ports. Lossy traffic require a large size buffer to absorb bursts of traffic and which would enable the switch/router to manage these infrequent bursts without resorting to dropping frames.

For lossless traffic, the switch/router can indicate to the sender to stop sending more traffic when congestion is encountered. However, during the time interval between the switch/router sending a congestion signal and the sender receiving and acting on that signal, some traffic would already have been sent. Any frame that is in transit, after the *back-pressure* signal is sent and before the sender stops transmission, must be absorbed without any drops. Allocation of a portion of the available buffer exclusively to absorb the above traffic will ensure lossless traffic.

To enable lossless support, 25% (twenty five percent) of the packet buffers can be allocated to the lossless pool at device initialization. Once allocated, this buffer cannot be use for lossy traffic. Also, once allocated, the buffer size cannot be shifted between the lossless and lossy buffer pools. Depending on your requirement, you can elect to enable or disable support for this buffer allocation.

Two QoS hardware profiles are provided to enable/disable support for buffer allocation. Select the *lossy* hardware profile to disable this feature. If you execute any PFC/Pause related commands, they will throw error messages. Select the *lossless* profile to enable this feature. PFC/Pause configurations can now be applied to the selected interface.

Use the `profile qos` command, applied to the QoS Hardware context, to configure the hardware buffer.

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 15: 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.
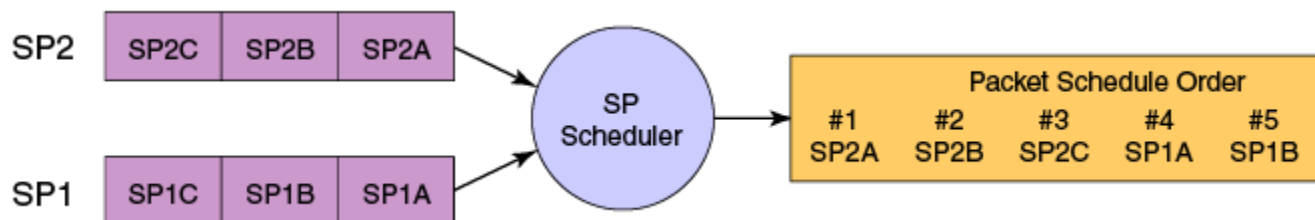
Figure 5: Strict priority schedule — two queues

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.
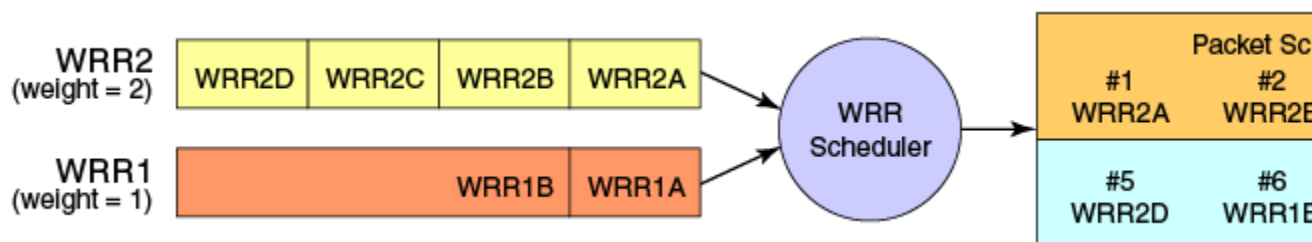
Figure 6: WRR schedule — two queues

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 16: 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.

QoS Ingress Data Buffer Management

> **Note**
> This topic does not apply to SLX 9150 devices.

Buffer management consists of the following.

- Packets arrived at ingress are stored in a data buffer (DB).
- A DB can be an on-chip buffer (OCB, 128 MB) or external DRAM (up to 8 GB).
- Each OCB size is 256 B and external DRAM buffer size is 2 KB.
- The Virtual output queue (VOQ) holds packet descriptors, which are lists of buffer descriptors (BD).
- The entire packet buffer can be configured for a given VOQ to 1.5 GB for D cards and 2 GB for M cards.

> **Important**
> Specifying a value over 1.5 GB for a D card generates an error:
> QSizeLimit LC Type -D ifIdx: 0x80000000, slot_id: 1, maxQueueSize: 1536.

- The DB and BD pools are managed per-license, as follows:
  - Avoid starving high-priority traffic by allocating too many resources to high-rate, low-priority traffic.
  - Each VOQ has its own minimum guaranteed BD and DB (10% of the queue size). The default VOQ size is 1 MB.
  - The non-guaranteed BD and DB are allocated from a shared pool.

For more information, see Configure Virtual Output Queuing on page 125.

> **Note**
> The buffer management feature is supported only on devices based on the DNX chipset family. For more information, see Supported Hardware.

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 and drop precedence can be described as following:

- Collect priority and drop precedence information from various portions of the packet header:
  - If a packet's EtherType matches 8100, decoding the PCP value derives a priority value and drop precedence.
  - For MPLS packets on supported SLX devices, decoding the EXP bits derive priority value and drop precedence.

- For IPv4 or IPv6 packets, decoding the DSCP bits derive priority value and drop precedence.
- For untagged Layer 2 packet, the port's default value derives traffic class and drop precedence.
- The derived values for PCP, EXP 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 and drop precedence values are 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. For MPLS packets on supported SLX devices, by decoding the EXP value from MPLS header.
4. Physical port default value.

Egress QoS Mutation

The QoS operation on egress traffic involves marking packets as they leave the chip on the egress port. As the packets are prepared to exit the device you can set the PCP, DSCP, and CoS values in the packet headers.

> **Note**
> Egress QoS mutation does not apply to SLX 9150 devices.

802.1P priority mapping (traffic class-to-PCP mutation)

The internal traffic class can be mapped to the outgoing PCP value when the packet egresses the switch. You can create a priority mapping table using a CoS mutation map. This CoS mutation map can then be applied to an egress interface to effect the priority re-mapping. This feature only maps the internal traffic class to outgoing priority.

DSCP remarking

This feature allows the user to remark the DSCP value of the egressing IP packet using the ingress DSCP value. The configuration must be bound to an egress interface.

DSCP-to-CoS mutation mapping

The ingress DSCP value can be mapped to outgoing 802.1P values by configuring a DSCP-to-CoS mutation map on the egress interface.

QoS Mutation Maps

Several types of QoS mutation maps can be applied to Ethernet ports.

Specifying the mutation map used on a port can lead to contradictions if there are other user-defined classes used in the same policy map that have a set CoS action configured. In this case the defined CoS takes priority over the mutation map.

> **Note**
> Egress QoS mutation does not apply to SLX 9150 devices.

> **Note**
> DSCP Mutation map is not supported on SLX 9150 & 9250 devices.

The available mutations are :

- **cos-cos**
- **cos-traffic-class**
- **dscp-dscp**
- **dscp-cos**
- **dscp-mutation**
- **dscp-traffic-class**
- **traffic-class-cos**

Configuring QoS for control traffic

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

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Set the TM CPU port shaper on slot 0 to 4000 Kbps with a burst size of 1KB.

   ```
   device(config)# qos cpu slot 0 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 0 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 0 port shaper rate 4000 burst 1
device(config)# exit
device# show run qos cpu
device# copy running-config startup-config
```

Increase the Egress Throughput on a TM Port

You can increase the egress throughput rate on a Traffic Management (TM) port on 10G and 100G line cards.

About This Task

On a 10G card, you can increase the rate from 300 Mbps to 13 Gbps. On a 100G card, you can increase the rate from 300 Mbps to 130 Gbps. The maximum speed supported by an interface is 130% of the default interface speed.

If you configured a breakout connector on an Ethernet interface, you can also increase the egress throughput rate on a breakout port.

> **Note**
> This feature is supported only on devices based on the DNX chipset family. For more information, see Supported Hardware.

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

   ```
   device(config)# interface Ethernet 1/2
   ```

3. Increase the egress throughput rate on the port.

   ```
   device(conf-if-eth-1/2)# qos port-speed-up 300000
   ```

   In this example, the egress throughput rate on the port is increased by 300 Mbps.

4. Return to privileged EXEC mode.

   ```
   device(conf-if-eth-1/2)# end
   ```

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

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

Example

The following example summarizes the commands in this procedure.

```
device# configure terminal
device(config)# interface ethernet 1/2
device(conf-if-eth-1/2)# qos port-speed-up 300000
device(conf-if-eth-1/2)# end
device# copy running-config startup-config
```

Configure a CoS-to-traffic Class Mutation Map

Configure a Cos-to-traffic class map to use the ingress 802.1p priority values to classify traffic to a specific traffic class (priority queue) and drop precedence.

About This Task

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.

Procedure

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
   ```

   The drop-precedence parameter is mandatory.

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
   ```

Example

The following example summarizes the commands in this procedure.

```
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
```

Applying a CoS-to-traffic class mutation map to an interface

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

Before You Begin
You have configured a QoS CoS-to-traffic class map.

About This Task

The internal traffic class can be mapped to the outgoing PCP value when the packet egresses the switch. A user can create a priority mapping table using a CoS mutation map. This CoS mutation map can then be applied to an egress interface to effect the priority remapping. This feature only maps the incoming priority to outgoing priority.

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode..

```
device(config)# interface ethernet 1/5
```

3. Apply the CoS-to-traffic class map to an ingress interface and return to privileged EXEC mode.

```
device(conf-if-eth-1/5)# qos cos-traffic-class tc_1
```

4. Return to privileged EXEC mode.

```
device(conf-if-eth-1/5)# end
```

5. Verify the configuration.

```
device# show qos maps cos-traffic-class tc_1

Cos-traffic-class map 'tc_1'
    In-Cos    : 0  1  2  3  4  5  6  7
    -------  -----------------------
Traffic-class: 5  5  5  5  5  5  5  5

    Enabled on the following interfaces: Eth 1/5
```

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

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

Applying a QoS CoS-to-traffic class mutation map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 1/5
device(conf-if-eth-1/5)# qos cos-traffic-class tc_1
device(conf-if-eth-1/5)# end
device# show qos maps cos-traffic-class tc_1
device# copy running-config startup-config
```

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.

Procedure

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.

About This Task

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

Before You Begin
A QoS DSCP-to-DSCP mutation map has been configured.

Procedure

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.

Configure a DSCP-to-CoS Mutation Map

Use the DSCP value of ingress packets to remap the egress 802.1p CoS priority values.

Procedure

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 step 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.

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

   In this example, the DSCP value 23 is mapped to CoS priority 4.

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

   In this example, DSCP value 43 is mapped to CoS priority 5.

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:
    Eth 1/3
   ```

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

   ```
   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.

Before You Begin

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

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

   ```
   device(config)# interface ethernet 1/5
   ```

3. Enable the DSCP mutation map on the interface.

   ```
   device(conf-if-eth-1/5)# qos dscp-cos dscpCosMap
   ```

4. Return to privileged EXEC mode.

   ```
   device(conf-if-eth-1/5)# 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: Eth 1/5
   ```

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 1/5
device(conf-if-eth-1/5)# qos dscp-cos dscpCosMap
device(conf-if-eth-1/5)# end
device# show qos maps dscp-cos
device# copy running-config startup-config
```

DSCP-to-Traffic Class Mappings

You can assign (map) a traffic class to a set of DSCP (Differentiated Services Code Point) values. This map is then used to assign the traffic class to a data packet based on the DSCP field in the packet header.

The following topics describe how 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.

Procedure

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.

Before You Begin
A QoS DSCP-to-traffic class map has been configured

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

```
device(config)# interface ethernet 1/2
```

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

```
device(conf-if-eth-1/2)# qos dscp-traffic-class dscpTcMap
```

4. Return to privileged EXEC mode.

```
device(conf-if-eth-1/2)# 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: Eth 1/2
```

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 1/2
device(conf-if-eth-1/2)# qos dscp-traffic-class dscpTcMap
device(conf-if-eth-1/2)# end
device# show qos maps dscp-traffic-class
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.

Procedure

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.

Before You Begin

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

About This Task

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.

Procedure

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.

Procedure

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.

Before You Begin

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

Procedure

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.

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.

Procedure

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. Verify the WRED configuration.

```
device(config)# do show qos red profiles 1

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

4. Return to privileged exec mode.

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

5. 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)# do show qos red profiles 1
```

```
device(config)# exit
device# copy running-config startup-config
```

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 You Begin

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

About This Task

LLFC can be configured only at the interface level.

Perform the following steps to configure LLFC.

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

   ```
   device(config)# interface ethernet 3/18
   ```

3. Enable LLFC in the transmission and reception directions for the port.

   ```
   device(conf-eth-3/18)# qos flowcontrol tx on rx on
   ```

4. Return to privileged EXEC mode.

   ```
   device(conf-eth-3/18)# end
   ```

5. Verify the configuration.

   ```
   device# show qos flowcontrol interface ethernet 3/18
   Interface Ethernet 3/18
     Mode 802.3x
        TX     RX          TX Output Paused    RX
     Admin Admin     Frames   512 BitTimes    Frames
     ----------------------------------------------
        On    On
   ```

6. Save the configuration.

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

LLFC configuration example

```
device# configure terminal
device(config)# interface ethernet 3/18
device(conf-eth-3/18)# qos flowcontrol tx on rx on
device(conf-eth-3/18)# end
```

```
device# show qos flowcontrol interface ethernet 3/18
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
```

Enable Priority Flow Control

Priority flow control (PFC) extends the basic IEEE 802.3x based flow control to multiple classes (8 classes). It enables applications that require flow control to coexist on the same link with applications that can manage without flow control.

About This Task

PFC defines each one of the eight different types of flows that can be subject to flow control. In the case of an Layer 2 network, PFC uses the priority bits within the VLAN tag (IEEE 802.1p) to differentiate up to eight types of flows that can be subject to flow control, each one independently.

PFC can be configured only at the interface level.

> **Note**
> PFC is supported only on devices based on the XGS chipset family. For more information, see Supported Hardware.

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

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

3. Enable PFC the CoS for the port.

   ```
   device(conf-eth-0/1)# qos flowcontrol pfc 2 tx on rx on
   ```

   In this example, PFC is on CoS 2.

4. Return to privileged EXEC mode.

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

5. Verify the configuration.

   ```
   show qos flowcontrol interface ethernet 0/1
   Interface Ethernet 0/1
   Mode PFC
   TX frames
   RX 0 frames
           TX   TX    RX    RX Output Paused
   CoS Admin Oper Admin Oper  512 BitTimes
   ------------------------------------
     0   Off  Off   Off   Off
     1   Off  Off   Off   Off
     2    On   On    On    On
     3   Off  Off   Off   Off
     4   Off  Off   Off   Off
     5   Off  Off   Off   Off
     6   Off  Off   Off   Off
     7   Off  Off   Off   Off
   ```

6. Save the configuration.

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

Example

This example summarizes the commands in this procedure.

```
device# configure terminal
device(config)# interface ethernet 0/1
device(conf-eth-0/1)# qos flowcontrol pfc 2 tx on rx on
device(conf-eth-0/1)# end
device# show qos flowcontrol interface ethernet 0/1
device# copy running-config startup-config
```

Configuring Hardware Buffer for Lossless Traffic

About This Task

Two QoS hardware profiles are provided to enable/disable support for buffer allocation. Select the lossy hardware profile to disable this feature. If you execute any PFC/Pause related commands, they will

throw error messages. Select the lossless profile to enable this feature. PFC/Pause configurations can now be applied to the selected interface.

Procedure

1. Enter configuration mode.

```
device# configure terminal
```

2. Enter the hardware context.

```
device# hardware
```

3. Use the following command to use the lossless profile.

```
device(hardware)# profile qos ?

Possible completions:
  lossless   set qos hardware lossless profile
  lossy      set qos hardware lossy profile

device(hardware)# profile qos lossless
device(hardware)#
```

4. Return to privileged exec mode.

```
device(hardware)# exit
```

5. Save the configuration.

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

6. To view the PFC/Pause statistics, use the `show qos flowcontrol stats` command.

```
device# show qos flowcontrol stats interface eth 0/1

Interface Ethernet 0/1
                                TX              RX
    --------------------------------------------
    PAUSE Frames:         565856          441122
    PFC Pri0   Frames:    565856          441122
    PFC Pri1   Frames:    565856          441122
    PFC Pri2   Frames:    565856          441122
    PFC Pri3   Frames:    565856          441122
    PFC Pri4   Frames:    565856          441122
    PFC Pri5   Frames:    565856          441122
    PFC Pri6   Frames:    565856          441122
    PFC Pri7   Frames:    565856          441122
```

Monitor TM Deleted or Discarded Packets

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.

About This Task

> **Note**
> This feature is supported only on devices based on the DNX chipset family. For more information, see Supported Hardware.

Procedure

1. Access global configuration mode.

   ```
   device# configure terminal
   ```

2. Access system-monitor 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 example configures the threshold of 10 TM deleted packets. A threshold of 0 disables the monitoring of the packets.

4. (Optional) 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 example configures the threshold of 15 TM device-discarded packets. A threshold of 0 disables the monitoring of the packets.

6. (Optional) Configure the logging interval to monitor all TM discarded packets.

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

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

8. (Optional) Configure the logging interval to monitor the VOQ discarded packets.

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

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

Example

The following example summarizes the commands in this procedure.

```
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
```

Example

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
```

Displaying the RED Profile and dropped packets information for an interface

Use the **do show red profiles** command to view the RED profile's configuration.

```
SLX(conf-if-eth-0/1)# do show qos red profiles 1

Red Profile 1
        Minimum Threshold: 10
        Maximum Threshold: 50
        Drop Probability: 100
        ECN: Off
```

```
    Activated on the following interfaces:
        Eth 0/1  traffic-class: 0 drop-precedence: 03
```

Similarly, use the **show qos red statistics** command to view the per port RED dropped
packets.

```
SLX# show qos red statistics interface eth 0/1
Statistics for interface: Eth 0/1
        Port Statistics:
                Packets Dropped: 147, Queue Full Drops: 222, ECN Marked: 234
```

Configuring Explicit Congestion Notification

About This Task

Explicit Congestion Notification (ECN) enables the switch/router to signal to endpoints about
congestion seen in traffic. This then enables the endpoints to take appropriate actions to mitigate traffic
congestion. The advantage of this method is that it reduces the packet drops and jitter encountered
when traffic congestion occurs.

Procedure

1. Enter configuration mode.

    ```
    device# configure terminal
    ```

2. Create a RED profile identified as profile 1, set the thresholds, set the drop probability, and enable the
    Explicit Congestion Notification feature.

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

3. Verify the ECN configuration.

    ```
    device(config)# do show qos red profiles 1

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

        Activated on the following interfaces:
            Eth 0/1  traffic-class: 0 drop-precedence: 03
    ```

4. Return to privileged exec mode.

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

5. Save the configuration.

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

6. To view the per port ECN statistics, use the `show qos red statistics` command.

    ```
    device# show qos red statistics interface eth 0/1

    Statistics for interface: Eth 0/1
    ```

```
        Port Statistics:
              Packets Dropped: 147, Queue Full Drops: 222, ECN Marked: 234
```

Example

This example creates a RED profile and applies it to an interface.

```
device# qos red-profile 1 min-threshold 10 max-threshold 50 drop-probability 100 ecn on
device# int eth 0/1
device(conf-if-eth0/1)# qos random-detect traffic-class 0 red-profile-id 1
```

Example

This example shows how to view the per port ECN statistics using the `show qos red statistics` command.

```
device# show qos red statistics interface eth 0/1

Statistics for interface: Eth 0/1
        Port Statistics:
              Packets Dropped: 147, Queue Full Drops: 222, ECN Marked: 234
```

Configuring Enhanced Transmission Selection

About This Task

Enhanced Transmission Selection (ETS) is a transmission selection algorithm (TSA) that is specified by IEEE 802.1Qaz draft standard. It defines priority-based processing and bandwidth allocation on different traffic classes with different traffic types (such as LAN, SAN, IPC).

ETS enables the switch/router to prioritize traffic based on CoS to Priority mapping. ETS enables you to configure the ETS/CEE (Converged Enhanced Ethernet) parameters. These parameters include mapping of incoming priorities (traffic with different CoS) to traffic-classes, assigning weights and corresponding bandwidth to those priorities, and scheduling of all occupied traffic-class traffic egress wire.

These settings are applied as a policy. This policy, the CEE Map, is then applied on the target interfaces. CEE maps can be applied to Layer 2 interfaces (physical switch interfaces), Layer 2 Port-channels, MCT Client Interfaces, and ICL Links. CEE maps cannot be applied to Physical or Logical Layer 3 interfaces and Tunnel interfaces.

Before You Begin

For Enhanced Transmission Selection to work, QoS hardware profile must be configured for *Lossless Profile*. Do the following to set the hardware profile to the *lossless* profile. To enable the *Lossless Profile*, we need to reboot the setup.

```
device# configure terminal
device(config)# hardware
device(config-hardware)# profile qos lossless
```

Procedure

1. Enter configuration mode.

```
device# configure terminal
```

2. Create a CEE map. There can only be one CEE map and it is always named as *default*.

```
device(config)# cee-map default
```

The CEE map is created if not already created. Only one CEE map can be created and it is always named as *default*.

3. Create the *priority-group-table* entries. First define the Strict Priority Group entries. Packets in the Strict Priority Group are accorded higher priority than normal traffic. A value in the range 15.0-15.7 indicates strict priority traffic. Within this range, packets marked with 15.0 has a higher priority than packets marked with 15.1, packets marked with 15.1 has higher priority than those marked with 15.2, and so on. Within the Strict Priority Group traffic, packets with the highest priority is assigned all the egress bandwidth. The next priority traffic is allocated the remaining bandwidth, and so on, till the bandwidth is exhausted. In a congestion scenario, once all the bandwidth has been allocated, there is a possibility that the Strict Priority Group traffic can also be dropped. There are eight (8) priorities in the range 15.0-15.7.

```
device(config-cee-map-default)# priority-group-table 15.0 pfc off
device(config-cee-map-default)# priority-group-table 15.1 pfc off
device(config-cee-map-default)# priority-group-table 15.2 pfc off
device(config-cee-map-default)# priority-group-table 15.3 pfc off
...
device(config-cee-map-default)# priority-group-table 15.7 pfc off
```

PFC cannot be configured for Strict Priority Group traffic. Trying to configure PFC will result in an error.

4. Next create the Low Priority (DWRR) entries. Eight (8) priorities are available in the range 0-7. When creating these entries, ensure that you allocate bandwidth ratios for these flows. Bandwidth is allocated using the *weight* parameter. Enter numerical values for *weight*. For example, set 10 for 10%, 20 for 20%. Ensure that the sum of all DWRR weights must equal 100%. If the sum does not equal 100%, then an error is thrown.

```
device(config-cee-map-default)# priority-group-table 1 weight 40 pfc on
device(config-cee-map-default)# priority-group-table 2 weight 60 pfc off
device(config-cee-map-default)#
```

Only create entries for those DWRR priorities that you plan to use. Do not create entries for those priorities which you do not intend to use and assign a weight of 0 to them. For example, this is an invalid configuration.

```
device(config-cee-map-default)# priority-group-table 3 weight 0 pfc off
```

5. Map the Priority with the entries in the priority-group-table. Use the **priority-table** command to create this mapping.

```
device(config-cee-map-default)# priority-table 2 2 2 1 2 2 2 15.0
```

Each entry in the row represents a CoS. In this mapping CoS 0-2, 4-6 are mapped with priority group 2. Priority group 2 is configured with 60% bandwidth allocation and PFC off. CoS 3 is mapped to Priority group 1. Priority group 1 is configured with 40% bandwidth allocation with PFC on. CoS 7 is assigned 15.0 indicating Strict Priority group.

In scenarios where Hybrid scheduling is used, the packets with Strict Priority Group traffic classes are served first. The remaining bandwidth, if any, is then allocated to the active DWRR queues according to the ratio of the weights allocated to them.

> **Note**
> CoS 7 is reserved for Strict Priority Group and is always assigned a value of 15.0.

6. Exit out of the CEE Map context.

```
device(config-cee-map-default)# exit
device(config)#
```

7. Assign the CEE Map to an interface.

```
device(config)# interface Ethernet 0/1
device(conf-if-eth-0/1)# cee default
device(conf-if-eth-0/1)#
```

The CEE map is applied to the selected interface. Once a CEE map is applied to the interface, the device starts advertising its capabilities to peers using LLDP TLVs. For more information on the LLDP TLVs that are used, see LLDP TLVs for Enhanced Transmission Selection on page 111.

8. To verify the configuration, use the **show cee maps default** command.

```
device(config)# do show cee maps default

CEE Map 'default'
Precedence: 1

Priority Group Table
1: Weight 40, PFC Enabled, BW% 40
2: Weight 60, PFC Disabled, BW% 60
15.0: PFC Disabled
15.1: PFC Disabled
15.2: PFC Disabled
15.3: PFC Disabled
15.4: PFC Disabled
15.5: PFC Disabled
15.6: PFC Disabled
15.7: PFC Disabled
Priority Table
CoS: 0 1 2 3 4 5 6 7
--------------------------------------
PGID: 2 2 2 1 2 2 2 15.0
Enabled on the following interfaces:
Ethernet 0/1
```

9. You can also use the **show qos interface** to view the CEE information.

```
SLX# show qos interface ethernet  0/1
[Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]
```

```
Interface Ethernet 0/1
  Provisioning Mode: cee
  CEE Map: default
  Default TC: 0

  TC-to-CoS Map: default
             In-TC: 0   1   2   3   4   5   6   7
       ------------------------------------
       Out-CoS(DP=0): 0   1   2   3   4   5   6   7
       Out-CoS(DP=1): 0   1   2   3   4   5   6   7
       Out-CoS(DP=2): 0   1   2   3   4   5   6   7
       Out-CoS(DP=3): 0   1   2   3   4   5   6   7

  TC-to-DSCP Map: default
             In-TC:  0    1    2    3    4    5    6    7
       ----------------------------------------------
       Out-DSCP(DP=0): 00   08   16   24   32   40   48   56

       Out-DSCP(DP=1): 00   08   16   24   32   40   48   56

       Out-DSCP(DP=2): 00   08   16   24   32   40   48   56

       Out-DSCP(DP=3): 00   08   16   24   32   40   48   56

  DSCP Mutation Map: default (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 :     60 61 62 63

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

  DSCP-to-CoS Map: default (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 02 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

  Per Traffic-Class Tail Drop Threshold (bytes)
             TC:      0      1      2      3      4      5      6      7
       ----------------------------------------------------------------
       Threshold:     0      0      0      0      0      0      0      0

  Flow control mode PFC
       CoS3 TX on, RX on
```

```
Traffic Class Scheduler configured for 1 Strict Priority queues
    TrafficClass:   0   1   2   3   4   5   6   7
    -------------------------------------------
    DWRRWeight:   0   0   0  40   0   0  60 ---
```

LLDP TLVs for Enhanced Transmission Selection

The following LLDP TLVs are used to share the capabilities of the local device with its peers. LLDP is used for this communication. When a configured CEE map is applied to an interface, the capabilities of this device is shared with its peers to enable proper configuration of the ETS parameters. The following TLVs are available for use.

- Priority Based Flow Control (PFC) Configuration TLV
- ETS Recommendation TLV
- Link Aggregation TLV

Priority Flow Control Configuration TLV

This LLDP TLV is used to advertise Priority Flow Control (PFC) configuration.

TLV type = 127	TLV information string length = 6	802.1 OUI 00-80-C2	802.1 subtype = 11	Willing	MBC	Re-served	PFC cap	PFC Enable
7 bits	9 bits	3 octets	1 octet	1 bit	1 bit	2 bits	4 bits	1 octet

TLV header ← ; → TLV information string = 6 octets
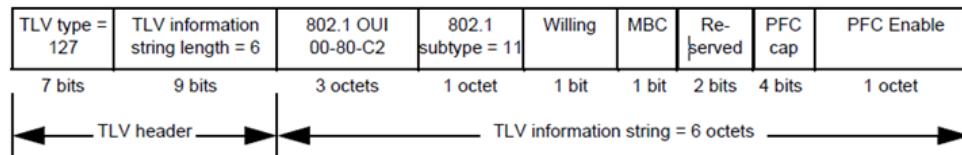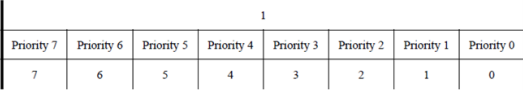
Figure 7: Priority Based Flow Control TLV Format

The various configuration parameters are:

Field	Description
TLV Type	Size: 7 bits. Describes the Type of this TLV. This value is always 127 as this is a custom TLV.
TLV Information String Length	Size: 9 bits. This field indicates the size of the TLV information being passed. For this TLV this value is six (6).
802.1 OUI	Size: 3 octets. The Organizationally Unique Identifier (OUI) assigned to the organization. This value is always *00-80-C2* for this TLV.
802.1 Subtype	Size: 1 octet. This is an internally defined subtype. Its value is always eleven (11). This is assigned and managed by the organization.
Willing	Size: 1 bit. When set to one (1), it indicates that the station is willing to accept configuration from the remote peer. Set to zero (0) when the station is not willing to change its configuration.
MBC	Size: 1 bit. When set to zero (0), it indicates that the station is capable of bypassing MACsec processing when MACsec is disabled. When set to one (1), the station indicates that it is not capable of bypassing MACsec processing when MACsec is disabled. MACSec and MBC are not supported on SLX-OS, and this bit is disregarded.
Reserved	Size: 2 bits. Reserved for future use and is not used.

Field	Description
PFC Cap	Size: 4 bits. A 4 bit unsigned integer value. Indicates the device's limitation of how many traffic classes May simultaneously support PFC. SLX-OS supports PFC on all eight (8) traffic classes.
PFC Enable	Size: 1 octet. The PFC Enable bit vector of 8 bits, one per priority. **Figure 8: PFC Enable bit vector** Refer to the above image. A value of one (1) indicates that PFC is enabled for the priority. A value of zero (0) indicates that PFC is disabled for the priority.

Enhanced Transmission Selection (ETS) Recommendation TLV

This LLDP TLV is used to advertise the recommended Enhanced Transmission Selection configuration. A willing system might receive a recommendation TLV which uses more traffic classes than the receiving system might support. In such cases, the receiving system may assign the received recommendations to match its capabilities.
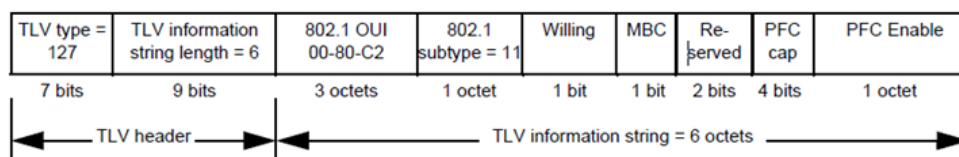
Figure 9: ETS Recommendation TLV

The various configuration parameters are:

Field	Description
TLV Type	Size: 7 bits. Describes the Type of this TLV. This value is always 127 as this is a custom TLV.
TLV Information String Length	Size: 9 bits. This field indicates the size of the TLV information being passed. For this TLV this value is twenty five (25).
802.1 OUI	Size: 3 octets. The Organizationally Unique Identifier (OUI) assigned to the organization. This value is always *00-80-C2* for this TLV.
802.1 Subtype	Size: 1 octet. This is an internally defined subtype. Its value is always ten (10). This is assigned and managed by the organization.
Reserved	Size: 8 bits. Reserved for future use and is not used.

Field	Description
Priority Assignment Table	Size: 4 octets. Indicates the mapping of priority with the corresponding traffic class. **Figure 10: Priority Assignment Table** This table consists of one 4 bit entry per Priority. Values of 0-7 can be assigned as Priority value and is mapped to the corresponding traffic class. All other values are considered *Reserved* and ignored.
Traffic Class Bandwidth Table	Size: 8 octets. This table consists of one eight (8) bit entry per traffic class. This table always contains eight (8) entries where each entry indicates the current allocated bandwidth for the traffic class. Ensure that the sum of allocated bandwidths is always 100. **Figure 11: Traffic Class Bandwidth Table**
TSA Assignment Table	Size: 8 octets. This table one eight (8) bit entry per traffic class. Each entry indicates the Transmission Selection Algorithm to be used for the traffic class. This value is always two (0) for each octet and indicates Enhanced Transmission Selection.

Link Aggregation TLV

This LLDP TLV is used to advertise aggregation capabilities of the link to the remote device. It also provides additional information such as, whether the link is currently in an aggregation (as specified in IEEE 802.1AX), and if it is in an aggregation, advertises the port identifier of the aggregation.
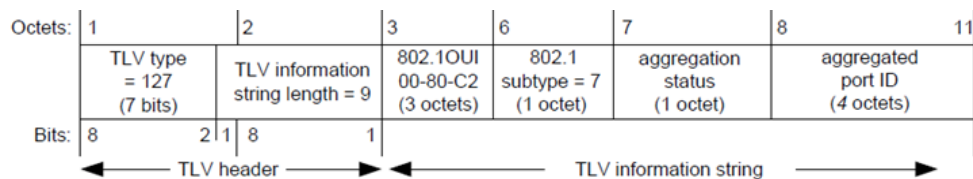
Figure 12: Link Aggregation TLV

The various parameters are:

Field	Description
TLV Type	Size: 7 bits. Describes the Type of this TLV. This value is always 127 as this is a custom TLV.
TLV Information String Length	Size: 9 bits. This field indicates the size of the TLV information being passed. For this TLV this value is nine (9).
802.1 OUI	Size: 3 octets. The Organizationally Unique Identifier (OUI) assigned to the organization. This value is always *00-80-C2* for this TLV.

Field	Description
802.1 Subtype	Size: 1 octet. This is an internally defined subtype. Its value is always seven (7). This is assigned and managed by the organization.
Aggregation Status	Size: 1 octet. Contains a bitmap of the link aggregation capabilities and the current aggregation status of the link. Only the first two bits are used. The first bit (bit 0) advertises the aggregation capability for the link. A value of zero (0) indicates that the link is not capable of being aggregated . A value of one (1) indicates that the link is capable of being aggregated. The second bit (bit 1) advertises the aggregation status for the link. A value of zero (0) indicates that the link is not currently in aggregation. A value of one (1) indicates that the link is currently in aggregation. Bits 2-7 are reserved.
Aggregated Port ID	Size: 4 octets. This field contains the IEEE 802.3 aggregated port identifier. This field is only meaningful if the link is currently a member of an aggregation.

Configuring scheduling

Perform the following tasks to configure scheduling.

Configuring strict priority egress scheduling

Follow these steps to configure strict priority scheduling.

Procedure

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
```

Configure a Strict Priority for the Multicast Queue

About This Task

> **Note**
> This feature is supported only on devices based on the DNX chipset family. For more
> information, see Supported Hardware.

Procedure

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
```

Example

This example summarizes the commands in this procedure.

```
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
```

Flow-based QoS

Flow-based Quality of Service (QoS) provides priority or a certain level of performance to data flows.

The high-level process for configuring flow-based QoS is as follows.

1. Configure a class map that classifies traffic according to the traffic properties required for your flow-based QoS needs. For more information, see Configuring a class map using an ACL on page 51.

2. Configure a policy map and associate it to the class map. For more information, see Configuring a policy map on page 118.

> **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. For more information, see Configuring QoS mutation map actions on page 119.

4. Bind the policy map to an interface. For more information, see:
   - Bind the Policy Map at the System Level on page 121
   - Bind the Policy Map to an Interface on page 122

5. Configure the OoS policing rate. For more information, see Configuring the QoS policing rate on page 123.

Related Topics
Match Access-Group Class Map Policing on page 23

The ACL-based policing feature controls the amount of bandwidth consumed by an individual flow or aggregate of inbound flows by limiting the traffic rate on a port according to criteria defined by the `match access-group` class map.

Configuring a class map using an ACL

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

Procedure

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.

About This Task
A rate limit policy map is configured and then applied to the type of QoS flow defined by the class map.

Procedure

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.

```
device# show policy-map detail policy
Policy-Map policy
    Class cmap1
      Police cir 43454
 Bound To: ET  1/33(in), Te 5/33(out)
```

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
```

Configuring QoS mutation map actions

Follow these steps to configure a QoS mutation map.

Before You Begin

A policy map and a class map have been configured.

About This Task

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**.

Procedure

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
```

Apply QoS Mutation Maps to an Interface

Follow these steps to specify the mutation map to be used on a port.

Before You Begin

A mutation map has been configured.

Procedure

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 1/1
   ```

4. Apply a DSCP-to-DSCP mutation map to the interface.

   ```
   device(conf-if-eth-1/1)# qos dscp-mutation dscpMutMap
   ```

   Different kinds of mutations can be used.

5. Return to privileged EXEC mode.

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

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

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

Example

The following example summarizes the commands in this procedure.

```
device# configure terminal
device(config)# policy-map policyMap1
device(config-policymap)# interface ethernet 1/1
device(conf-if-eth-1/1)# qos dscp-mutation dscpMutMap
device(conf-if-eth-1/1)# end
device# show qos map dscp-mutation dscpMutMap
device# copy running-config startup-config
```

Bind the Policy Map at the System Level

Follow these steps to apply policing parameters to an interface.

Procedure

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 conform-tc 6 exceed-tc 2 conform-
dscp 61 exceed-dscp 63

   Bound To: none
```

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

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

Example

This example summarizes the commands in this procedure.

```
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
```

Bind the Policy Map to an Interface

Follow these step to configure the default remapping priorities.

About This Task

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.

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

   ```
   device(config)# interface ethernet 1/40
   ```

3. Bind a policy map to egress traffic on the interface.

   ```
   device(config-if-eth-1/40)# service-policy out policyMap1
   ```

4. Bind a policy map to ingress traffic on the interface.

   ```
   device(config-if-eth-1/40)# service-policy in policyMap1
   ```

5. Return to privileged EXEC mode.

   ```
   device(config-if-eth-1/40)# end
   ```

6. Verify the configuration.

   ```
   device# show policy-map interface ethernet 1/40

   Ingress Direction :
    Policy-Map policyMap1
       Class class_1
         matches 0 packets
         Police cir 40000 cbs 5000 eir 40000 ebs 3000 conform-tc 6 exceed-tc 2 conform-
   dscp 61 exceed-dscp 63
           Stats:
             Operational cir:39856 cbs:5000 eir:39856 ebs:3000
             Conform Byte:0 Exceed Byte:0 Violate Byte:0

   Egress Direction :
    Policy-Map policyMap1
       Class class_1
         matches 0 packets
         Police cir 40000 cbs 5000 eir 40000 ebs 3000 conform-tc 6 exceed-tc 2 conform-
   dscp 61 exceed-dscp 63
           Stats:
             Operational cir:39856 cbs:5000 eir:39856 ebs:3000
             Conform Byte:0 Exceed Byte:0 Violate Byte:0
   ```

   > **Note**
   > Egress policer is not supported on the SLX 9150 device. The **police** command affects only the CIR.

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

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

Example

This example summarizes the commands in this procedure.

```
device# configure terminal
device(config)# interface ethernet 1/40
device(conf-if-eth-1/40)# service-policy out policyMap1
device(conf-if-eth-1/40)# service-policy in policyMap1
device(conf-if-eth-1/40)# end
device# show policy-map interface ethernet 1/40
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.

Procedure

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.

Before You Begin

A policy map has been configured.

Procedure

1. Enter global configuration mode.

    ```
    device# configure terminal
    ```

2. Access interface configuration mode.

    ```
    device(config-policymap-class)# interface ethernet 1/49
    ```

3. Bind the ingress policy map policy map to the interface.

    ```
    device(conf-if-eth-1/49)# service-policy in policy_1
    ```

4. Return to privileged EXEC mode.

    ```
    device(conf-if-eth-1/49)# end
    ```

5. Verify the configuration.

    ```
    device# show policy-map

    Number of policy maps : 2
    ...
    Policy-Map policy_1
      Bound To: Et 1/49(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 1/49
device(conf-if-eth-1/49)# service-policy in policy_1
device(conf-if-eth-1/49)# end
device# show policy-map
device# copy running-config startup-config
```

Configure Virtual Output Queuing

Virtual output queueing (VOQ) is a technique where multiple queues are maintained.

About This Task

> **Note**
> This feature is supported only on devices based on the DNX chipset family. For more information, see Supported Hardware.

Procedure

1. Access global configuration mode.

   ```
   device# configure terminal
   ```

2. Access 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 example 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 0
   ```

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

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

Example

This example summarizes the commands in this procedure.

```
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
```

Configure an MPLS QoS DSCP-to-EXP Mutation Map

You can configure an MPLS QoS DSCP-to-EXP mutation map.

About This Task

This configuration is not supported on SLX 9150 devices.

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create and name an MPLS QoS DSCP-to-EXP mutation map.

   ```
   device(config)# qos-mpls map dscp-exp dscpExpMap
   ```

3. Define the DSCP-to-EXP values.

   ```
   device(dscp-exp-dscpExpMap)# dscp 0 to exp 7
   device(dscp-exp-dscpExpMap)# dscp 3 to exp 4
   device(dscp-exp-dscpExpMap)# dscp 61 to exp 5
   device(dscp-exp-dscpExpMap)# end
   ```

4. Verify the configuration.

   ```
   device# show qos-mpls maps dscp-exp dscpExpMap

   dscp-exp map 'dscpExpMap'(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 02 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 slots:
   ```

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

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

Example

This example summarizes the commands in this procedure.

```
device# configure terminal
```

```
device(config)# qos-mpls map dscp-exp dscpExpMap
device(dscp-exp-dscpExpMap)# dscp 0 to exp 7
device(dscp-exp-dscpExpMap)# dscp 3 to exp 4
device(dscp-exp-dscpExpMap)# dscp 61 to exp 5
device(dscp-exp-dscpExpMap)# end
device# show qos maps dscp-exp dscpExpMap
device# copy running-config startup-config
```

Applying an MPLS QoS DSCP-to-EXP mutation map globally

Follow these steps to apply an MPLS QoS DSCP-to-EXP mutation map globally.

Before You Begin
An MPLS QoS DSCP-to-EXP map is configured.

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Apply an MPLS QoS DSCP-to-EXP map globally.

   ```
   device(config)# qos-mpls map-apply dscp-exp dscpExpMap all
   ```

3. Return to privileged exec mode

   ```
   device(config)# end
   ```

4. Verify the configuration.

   ```
   device# show qos maps dscp-exp

   dscp-exp map 'dscpExpMap'(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 02 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 slots:
   ```

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

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

Apply an MPLS QoS DSCP-to-EXP mutation map globally configuration example

```
device# configure terminal
device(config)# qos-mpls map-apply dscp-exp dscpExpMap all
device(config)# end
device# show qos-mpls maps dscp-exp
device# copy running-config startup-config
```

Configure an MPLS QoS EXP-to-DSCP Mutation Map

You can configure an MPLS QoS EXP-to-DSCP mutation map.

About This Task

This configuration is not supported on SLX 9150 devices.

Procedure

1. Enter global configuration mode.

    ```
    device# configure terminal
    ```

2. Create and name an MPLS QoS EXP-to-DSCP mutation map.

    ```
    device(config)# qos-mpls map exp-dscp expDscpMap
    ```

3. Define the EXP-to-DSCP values.

    ```
    device(exp-dscp-expDscpMap)# exp 0 to dscp 7
    device(exp-dscp-expDscpMap)# exp 1 to dscp 23
    device(exp-dscp-expDscpMap)# exp 3 to dscp 31
    device(exp-dscp-expDscpMap)# exp 4 to dscp 62
    ```

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

4. Return to privileged exec mode

    ```
    device(exp-dscp-expDscpMap)# end
    ```

5. Verify the configuration.

    ```
    device# show qos maps exp-dscp expDscpMap

    exp-dscp map 'expDscpMap'
        Exp   : 0  1  2  3  4  5  6  7
        ------- -----------------------
        DSCP  : 0  2  4  3  6  4  5  7

        Enabled on the following slots:
    ```

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

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

Example

This example summarizes the commands in this procedure.

```
device# configure terminal
device(exp-dscp-expDscpMap)# exp 0 to priority 7 drop-precedence 0
device(exp-dscp-expDscpMap)# exp 0 to dscp 7
device(exp-dscp-expDscpMap)# exp 1 to dscp 23
device(exp-dscp-expDscpMap)# exp 3 to dscp 31
device(exp-dscp-expDscpMap)# exp 4 to dscp 62
device(exp-dscp-expDscpMap)# end
device# show qos maps exp-dscp expDscpMap
device# copy running-config startup-config
```

Applying an MPLS QoS EXP-to-DSCP mutation map globally

Follow these steps to apply an MPLS QoS EXP-to-DSCP mutation map globally.

Before You Begin

An MPLS QoS EXP-to-DSCP mutation map is configured.

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

2. Apply an MPLS QoS EXP-to-DSCP mutation map globally.

```
device(config)# qos-mpls map-apply exp-dscp expDscpMap all
```

3. Return to privileged exec mode

```
device(exp-dscp-expDscpMap)# end
```

4. Verify the configuration.

```
device# show qos-mpls maps exp-dscp

exp-dscp map 'expDscpMap'
    Exp   : 0  1  2  3  4  5  6  7
    ------- -----------------------
    DSCP  : 0  2  4  3  6  4  5  7

    Enabled on the following slots:
       ALL
```

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

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

Apply an MPLS QoS EXP-to-DSCP mutation map globally configuration example

```
device# configure terminal
device(config)# qos-mpls map-apply exp-dscp expDscpMap all
device(config)# end
device# show qos-mpls maps exp-dscp
device# copy running-config startup-config
```

Configure an MPLS QoS EXP-to-Traffic Class Mutation Map

Follow these steps to configure an MPLS QoS EXP-to-traffic class mutation map. This configuration is not supported on SLX 9150 devices.

Procedure

1. Enter global configuration mode.
```
device# configure terminal
```

2. Create and name an MPLS QoS EXP-to-traffic class mutation map.
```
device(config)# qos-mpls map exp-traffic-class expTcMap
```

3. Define the EXP-to-traffic class values.
```
device(exp-traffic-class-expTcMap)# exp 0 to traffic-class 7 drop-precedence 0
device(exp-traffic-class-expTcMap)# exp 1 to traffic-class 7 drop-precedence 1
device(exp-traffic-class-expTcMap)# exp 4 to traffic-class 7 drop-precedence 0
device(exp-traffic-class-expTcMap)# exp 5 to traffic-class 7 drop-precedence 1
device(exp-traffic-class-expTcMap)# exp 6 to traffic-class 7 drop-precedence 2
device(exp-traffic-class-expTcMap)# exp 7 to traffic-class 7 drop-precedence 3
```

You can enter eight mappings corresponding to EXP values of 0 to 7.

4. Return to privileged EXEC mode.

```
device(exp-traffic-class-expTcMap)# end
```

5. Verify the configuration.

```
device# show qos-mpls maps exp-traffic-class
exp-traffic-class map 'expTcMap'
    Exp  :    0  1  2  3  4  5  6  7
    ------- ----------------------
Traffic-class: 5  5  4  6  5  5  5  5
Drop-Preced  : 0  1  1  1  0  2  2  1

    Enabled on the following slots:
```

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

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

Example

This example summarizes the commands in this procedure.

```
device# configure terminal
device(config)# qos-mpls map exp-traffic-class expTcMap
device(exp-traffic-class-expTcMap)# exp 0 to traffic-class 7 drop-precedence 0
device(exp-traffic-class-expTcMap)# exp 1 to traffic-class 7 drop-precedence 1
device(exp-traffic-class-expTcMap)# exp 4 to traffic-class 7 drop-precedence 0
device(exp-traffic-class-expTcMap)# exp 5 to traffic-class 7 drop-precedence 1
device(exp-traffic-class-expTcMap)# exp 6 to traffic-class 7 drop-precedence 2
device(exp-traffic-class-expTcMap)# exp 7 to traffic-class 7 drop-precedence 3
device(exp-traffic-class-expTcMap)# end
device# show qos-mpls maps exp-traffic-class
device# copy running-config startup-config
```

Applying an MPLS QoS EXP-to-traffic class mutation map globally

Follow these steps to apply an MPLS QoS EXP-to-traffic class mutation map globally.

Before You Begin
A MPLS QoS EXP-to-traffic class map is configured.

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

2. Apply the MPLS QoS EXP-to-traffic class map globally.

```
device(config)# qos-mpls map exp-traffic-class expTcMap ALL
```

3. Return to privileged exec mode

```
device(exp-traffic-class-expTcMap)# end
```

4. Verify the configuration.

```
device#  show qos-mpls maps exp-traffic-class

exp-traffic-class map 'expTcMap'
```

```
    Exp   :    0  1  2  3  4  5  6  7
   ------- -----------------------
traffic-class  : 5  5  4  6  5  5  5  5
drop-precedence: 0  1  1  1  0  2  2  1

    Enabled on the following slots:
```

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

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

Apply a MPLS QoS EXP-to-traffic class map globally configuration example

```
device# configure terminal
device(config)# qos-mpls map exp-traffic-class expTcMap ALL
device(exp-traffic-class-expTcMap)# end
device# show qos-mpls maps exp-traffic-class
device# copy running-config startup-config
```

Configure an MPLS QoS Traffic Class-to-EXP Mutation Map

You can configure an MPLS QoS traffic class-to-EXP mutation map.

About This Task
This configuration is not supported on SLX 9150 devices.

Procedure

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create and name an MPLS QoS traffic class-to-EXP mutation map.

   ```
   device(config)# qos-mpls map traffic-class-exp tcExpMap
   ```

3. Define the traffic class-to-EXP values.

   ```
   device(traffic-class-exp-tcExpMap)# traffic-class 0 drop-precedence 0 to exp 7
   device(traffic-class-exp-tcExpMap)# traffic-class 1 drop-precedence 0 to exp 6
   device(traffic-class-exp-tcExpMap)# traffic-class 4 drop-precedence 0 to exp 3
   device(traffic-class-exp-tcExpMap)# traffic-class 5 drop-precedence 0 to exp 2
   device(traffic-class-exp-tcExpMap)# traffic-class 6 drop-precedence 0 to exp 2
   device(traffic-class-exp-tcExpMap)# traffic-class 7 drop-precedence 0 to exp 1
   device(traffic-class-exp-tcExpMap)# end
   ```

4. Return to privileged exec mode.

   ```
   device(traffic-class-exp-tcExpMap)# end
   ```

5. Verify the configuration.

   ```
   device#  show qos-mpls maps traffic-class-exp

   traffic-class-cos map    'TcExpMap' (Drop-Precedence = dp)
    dp: traffic-class  :  0  1  2  3  4  5  6  7
   ------------------- -----------------------
    0: exp             :  7  6  2  3  3  2  2  1
    1:                 :  0  1  2  3  4  5  6  7
    2:                 :  0  1  2  3  4  5  6  7
    3:                 :  0  1  2  3  4  5  6  7

   Enabled on the following slots :
   ```

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

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

Example

This example summarizes the commands in this procedure.

```
device# configure terminal
device(config)# qos-mpls map traffic-class-exp tcExpMap
device(traffic-class-exp-tcExpMap)# traffic-class 0 drop-precedence 0 to exp 7
device(traffic-class-exp-tcExpMap)# traffic-class 1 drop-precedence 0 to exp 6
device(traffic-class-exp-tcExpMap)# traffic-class 4 drop-precedence 0 to exp 3
device(traffic-class-exp-tcExpMap)# traffic-class 5 drop-precedence 0 to exp 2
device(traffic-class-exp-tcExpMap)# traffic-class 6 drop-precedence 0 to exp 2
device(traffic-class-exp-tcExpMap)# traffic-class 7 drop-precedence 0 to exp 1
device(traffic-class-exp-tcExpMap)# end
device# show qos-mpls maps traffic-class-exp
device# copy running-config startup-config
```

Applying an MPLS QoS traffic class-to-EXP mutation map globally

Follow these steps to apply an MPLS QoS traffic class-to-EXP mutation map globally.

Before You Begin
An MPLS QoS traffic class-to-EXP map is configured.

Procedure

1. Enter global configuration mode.

```
device# configure terminal
```

2. Apply an MPLS QoS traffic class-to-EXP mutation map globally.

```
device(config)# qos-mpls map-apply traffic-class-exp TcExpMap all
```

3. Return to privileged exec mode

```
device(config)# end
```

4. Verify the configuration.

```
device# show qos-mpls maps traffic-class-exp

traffic-class-cos map    'TcExpMap' (Drop-Precedence = dp)
 dp: traffic-class  :  0  1  2  3  4  5  6  7
 ------------------ ----------------------
 0: exp             :  7  6  2  3  3  2  2  1
 1:                 :  0  1  2  3  4  5  6  7
 2:                 :  0  1  2  3  4  5  6  7
 3:                 :  0  1  2  3  4  5  6  7

Enabled on the following slots:
```

5. Save the configuration.

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

Apply a MPLS QoS traffic class-to-EXP mutation map globally configuration example

```
device# configure terminal
device(config)# qos-mpls map-apply traffic-class-exp TcExpMap all
device(config)# end
device# show qos-mpls maps traffic-class-exp
device# copy running-config startup-config
```

Traffic Management Counters and Statistics

Counters and Statistics Overview

Note

This section does not apply to the SLX 9150 device.

SLX-OS uses algorithmic and sequential sampling simultaneously. This combination ensures that the entire counter engine database is periodically delivered into the software. You can set the sequential sampling timer to a lower rate as the overflow is prevented by algorithmic sampling.

Statistics collection mechanisms

In this implementation, statistics collection uses counter engines as a collection mechanisms.

With counter engines:
- There are 16 on-chip counter engines with 16k packet and octet counters each.
- Two mini counter engines B0/B1 are dedicated to the egress queue and support 4k 64-bit entries.
- Each counter engine can be individually assigned to a statistics source.
- A statistic flow is mapped to a set of counters within one of the counting engines.
- A packet is mapped to a counter pair within a counter set according to the packet's disposition (drop/forward status) and color.

Traffic Management Counter Types

Two counter types apply to traffic management.

Device counter

Device-level counters track packet counts per chip. Ingress counters keep track of the following:
- Number of ingress packets
- IQM enqueued and dequeued packets and bytes
- Total discard packets

- Total deleted packet
- Packets destined to invalid queues (those dropped by the traffic management)

Egress counters keep track of the following:

- Discarded unicast packets counter
- Discarded multicast packets
- EGQ packets

The polling time for traffic management device statistics is 10 seconds.

VOQ counter

This counter counts the packets that are enqueued or discarded for a VOQ.

For SLX devices, 16,000 VOQs statistics are available, out of which 15,000 are used to track VOQ traffic for each egress port. Entries for the TM VOQ counter engine are 1:1 mapped to the VOQs. Each VOQ ID directly results in a counter engine statistics ID and points to the counter set of the engine associated with the VOQ.

The polling time for traffic management VOQ statistics is 4 minutes.

Traffic Management Counters

These commands display traffic management (TM) counter statistics.

TM global statistics command

```
show tm statistics device
```

TM device level statistics commands

Use these commands to display traffic management statistics for interface modules.

```
device# show tm statistics device interface ethernet 4/1

TM Counters:
==================================================

Ingress Counters:
----------------------------------------------
Total Ingress Pkt Count              1164304848
CPU Packet Count                     0
Enque Packet Count                   953260002
DeQue Packet Count                   953260002
Total Discard Pkt Count              211045069
Oldest Discard Pkt Count             0
Resolved to be dropped               437190

Egress Counters:
----------------------------------------------
Unicast Pkt Count                    586664425
FQP Pkt Count                        586664425
Discard UC Pkt Count                 0
Discard MC Pkt Count                 0
```

```
MC Packet Count                         0
EHP Discard Count                       0

device# show tm statistics device interface ethernet 3/25 details

TM Counters:
=====================================================

Ingress Counters:
-------------------------------------------------
Total Ingress Pkt Count                 863033385
CPU Packet Count                        13
NIF Packet Count                        863033385
OAMP Packet Count                       0
OLP Packet Count                        0
Recycle Packet Count                    0
MMU IDR Packet Count                    0
Enque Packet Count                      850127673
DeQue Packet Count                      850127673
Total Discard Pkt Count                 12905725
Oldest Discard Pkt Count                0
Resolved to be dropped                  0
FDT Packet Count                        10837941
CRC Error Count                         0

Egress Counters:
-------------------------------------------------
Reassembly error Discard Count          21538682
    Packet UC Discard Count             10769341
    Packet MC Discard Count             0
    SOP UC Discard Count                0
    SOP MC Discard Count                10769341
Filter Discard Count                    10769341
    MC Pruning High Priority            0
    MC Pruning Low Priority             0
    LAG Pruning Discard Count           0
    PMF Discard Count                   0
    VLAN Member Discard Count           0
Discard UC Pkt Count                    0
Discard MC Pkt Count                    334638
UC Pkt Count                            839289732
MC Packet Count                         10434703
FQP Pkt Count                           849724435
Editor Discard Pkt Count                0
Editor Pkt Count                        849724435
Total Egress Pkt Count                  849724435
```

TM VOQ commands

This command displays a summary count of traffic management VOQ discards.

```
device# show tm voq-stat ingress-device ethernet 2/1 discards

---------------- Ports 1/1 - 1/36 --------------------
Dest Port  |  Prio  |  Queue  |   Discards
-----------------------------------------------------
3/1        |   0    |   320   |    2473804
2/4        |   0    |   224   |    1867789
4/2        |   2    |   434   |    1023452
4/8        |   4    |   487   |     920349
1/2        |   1    |   120   |     858723
1/3        |   1    |   128   |      75328
```

```
2/5         |   0   |   260   |      22234
2/6         |   0   |   268   |       5248
```

> **Note**
> The entries are sorted by highest number of discards with eight entries displayed by default.

This command displays a summary count of traffic management VOQ discards with priority 0.

```
device# show tm voq-stat ingress-device ethernet 2/1 discards priority 0

---------------- Ports 1/1 - 1/36 --------------------
Dest Port  |  Prio  |  Queue  |    Discards
---------------------------------------------------------
3/1         |   0   |   320   |    2473804
2/4         |   0   |   224   |    1867789
2/5         |   0   |   260   |      22234
2/6         |   0   |   268   |       5248
```

> **Note**
> The entries are sorted by highest number of discards with eight entries displayed by default.

Use this command to display TM VOQ statistics for an egress interface.

```
device# show tm voq-stat ingress-device ethernet 2/1 egress-port ethernet 2/7 priority 2

VOQ-Counters:
=================================================

Priority 2
-------------------------------------------------
EnQue Pkt Count                             67404602
EnQue Bytes Count                           1768413221
Total Discard Pkt Count                     0
Total Discard Bytes Count                   0
Current Queue Depth                         0
Maximum Queue Depth since Last read         160
```

This command displays a summary of TM VOQ **max-queue-depth** parameter statistics.

```
device# show tm voq-stat ingress-device 2/1 max-queue-depth

------------------ Ports 1/1 - 1/36 --------------------
Dest Port  |  Prio  |  Queue  |   Max Depth  |  Max Util
---------------------------------------------------------
3/1         |   0   |   320   |    1013804   |    96%
2/4         |   0   |   224   |     902789   |    86%
4/2         |   2   |   434   |     543440   |    51%
4/8         |   4   |   487   |     220349   |    21%
1/2         |   1   |   120   |     138723   |    13%
1/3         |   1   |   128   |      97328   |     9%
2/5         |   0   |   260   |      34234   |     3%
2/6         |   0   |   268   |      11723   |     1%
```

This command displays a summary of TM VOQ maximum buffer utilization.

```
device# show tm voq-stat ingress-device 2/1 max-buffer-util

--------- Ports 1/1 - 1/36 ---------
  Max Buffer Size |  Max Buffer Util
----------------------------------
      6007013804 |       96%
```

TM device-level statistics for XGS-based platforms

Only the following command displays TM statistics for XGS-based platforms. For a list of such devices, see Supported Hardware.

```
device# show qos tx-queue interface ethernet 0/1
```