BROCADE

CONFIGURATION GUIDE

Brocade SLX-OS QoS and Traffic Management Configuration Guide, 17s.1.00

Supporting the Brocade SLX 9140 and 9240 Switches

53-1004981-01
29 March 2017

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Notes, cautions, and warnings

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

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

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

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

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

Text formatting conventions

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

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

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

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	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. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.

Click the **Support** tab and select **Document Library** to access product documentation on MyBrocade or www.brocade.com. You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document feedback

Quality is our first concern at Brocade, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com
- By sending your feedback to documentation@brocade.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.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online or by telephone. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone
Preferred method of contact for non-urgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll-free number: +1-408-333-6061

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/solution provider, contact your OEM/solution provider for all of your product support needs.

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/solution provider.

About This Document

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Brocade Communications Systems, Inc. for SLX-OS Release 17s.1.00, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release:

- Brocade SLX 9140 switch
- Brocade SLX 9240 switch

NOTE

Some of the commands in this document use a slot/port designation. Because the Brocade SLX 9140 switch and the Brocade SLX 9240 switch do not contain line cards, the slot designation must always be "0" (for example, 0/1 for port 1).

To obtain information about other Brocade OS versions, refer to the documentation specific to that version.

Traffic Policing

Rate limiting and traffic policing overview

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

Rate limiting is applicable to inbound traffic, where packets are dropped that are above the configured committed rates.

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

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

Service policies — policy maps, class maps, and policers

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

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

One service policy can be applied per interface per direction.

Policy maps

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

When using the traffic policing policies available from previous versions, the policy parameters are provided explicitly for each port during port configuration. In this version, the policies must be defined using a policy map.

The number of policy maps that you can configure for a system depends on the hardware profile. Refer to the Release Notes for your device.

You can configure a policer by modifying the remarking values in the default policer remarking profile and then applying the profile to the policer. See Configuring a policer with the default policer remarking profile on page 39 for more information.

See the Policy maps on page 12 section for more information.

Class maps

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

Actions supported in the egress direction are shaper and scheduler (using a default class map). A policy map with any other action (whether it uses a default class map or an ACL-based class map) can be applied only in the ingress direction.

The default and **match access-group** class maps cannot be combined in a single policy map.

See the Class maps on page 15 section for more information.

Service policy configuration rules

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

- All policy map and class map names used in a service policy must be unique among all maps of that type.
- You can bind a service policy to multiple interfaces.
- A service policy can only be bound to physical ports and port-channel interfaces (LAGs). It cannot be bound to virtual interfaces.
- A service policy cannot be bound to interface if a class map is not associated with the policy map referenced in the service policy.
- If a service policy is bound to interface, and the policy class map lacks mandatory policer attributes (such as the CIR settings), then 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.
- Broadcast, unknown unicast and multicast (BUM) storm control and service policy port based rate limiting are mutually exclusive features. Only one can be enabled at a time on a given interface.

Policy maps

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

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

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

When using the traffic policing policies available from previous versions, the policy parameters are provided explicitly for each port during port configuration. In this version, the policies must be defined using a policy map. One policy map can be specified per service policy. You can configure up to 2000 policy maps per system.

Policy map configuration rules

Follow these rules when configuring traffic policing:

- A policer map (policy map or class map) name must be unique among all maps of that type.
- A policer name must begin with a-z or A-Z . An underscore, hyphen, and numeric values 0-9 can be used in the body of the name but not as the first character.
- You can configure a policer by modifying the remarking values in the default policer remarking profile and then applying the profile to the policer. See Configuring a policer with the default policer remarking profile on page 39 for more information.
- You can configure a maximum of 2000 policy maps.

- ACL-based class maps and default class maps can be used in a single policy map.
- Flow-based QoS is not supported in the egress direction. You can apply QoS to the flow using the policy map on the interface only in the ingress direction.
- For an ingress or egress service policy, one default class map can be specified per policy map.
- For an ingress-only service policy, 50 class maps, including the default class map, are supported per policy map.
- Broadcast, unknown unicast, and multicast (BUM) policies are counted separately.
- You cannot delete a policy map if is referenced in an active service policy (applied on an interface).

QoS shaping rate

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

Committed information rate and committed burst size

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

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

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

The traffic rate limited by the CIR bucket can have its priority, traffic class, and DSCP values changed.

Excess information rate and excess burst size

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

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

The traffic rate limited by the EIR bucket can have its priority, traffic class, and DSCP values changed.

Traffic policing behavior

Consider these behaviors when configuring traffic policing.

- Policer actions are applicable only to data traffic.

- When a Layer 2 control protocol is not enabled on an interface, those packets are dropped at ingress and are subject to ingress policing.
- If the user-configured CBS value is less than 2 times the jumbo frame size, then 2 times the jumbo frame size will be programmed as the CBS on hardware. For example, if you configure CBS as 12,000 bytes, and the jumbo frame size of the interface is 9000 bytes, then when a policy map is applied on this interface, the CBS value that is programmed on the hardware is 2 times the jumbo frame size, which is 18,000 bytes
- If the CBS and EBS values are not configured, then these values are derived from th CIR and EIR respectively. The burst size calculation is: Burst size (CBS or EBS) = (1.2 × information rate (CIR or EIR)) ÷ 8
- If you do not configure EIR and EBS, then single-rate, two-color scheme is applied (packets are marked as either green or red).
- You have the responsibility to configure rate limit threshold values on an interface based on interface speed. No validation is performed for user configured values against the interface speed.
- Because CIR is a mandatory policer attribute, you cannot delete the CIR parameter. If you want to delete the CIR attribute, then you should execute the **no police** command in policy-map-class sub-mode, which deletes all policer attributes.
- Packet drops caused by any action other than the ACL are included in the policer counter.

Traffic management egress buffer thresholds

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 1 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 2 Local port TM egress thresholds on the device

Threshold	PD FC	PD Drop	DB 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

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

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

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

Class map configuration rules

Follow these rules when configuring class maps:

- A class map name must be unique among all maps of this type.
- A class-map name must begin with an alphabetic character from a to z or from A to Z .
- Underscores, hyphens, and numeric characters 0 to 9 can be used in the body of the name but not as the first character.
- A class map cannot be deleted if it is referenced in a policy map.
- A class map cannot be deleted from a policy map when the policy map is bound to an active service policy.
- When you create a user-defined class map, you can associate it with a MAC, IPv4, or IPv6 ACL, using the **match access-group** command.
- The number of class maps that you can configure depends on the hardware profile you are using. Refer to the Release Notes for more information.
- The default and user-defined class maps created with the **match access-group** command can be combined into a single policy map.
- Only one default class map can be specified per policy map.

Class map policer configuration parameters

Use these values when setting the CIR, CBS, EIR, and EBS parameters.

TABLE 3 Map parameters for rate limiting

Parameter	Values	Range	Increments of
cir - Committed information rate	bits per second	18000 through 300000000000	Starts at 18000 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	18000 through 300000000000	Starts at 18000 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 parameters:

- The cir value must be specified, all other parameters are optional.
- 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.
- If you specify the values of both cir and cbs, in police the configured value takes priority over the default values.
- Should the cir value be updated, the configured cbs value is retained, the default value is not restored.
- If you want to revert to the default cbs value, you must first remove the configured value of cbs.

Traffic policer configuration rules for class maps

The following are rules for 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 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 to 9 are permitted, except as the first character of the name.
- The configurable CIR and EIR range starts from 18000 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 × MTU value, then 2 × MTU is programmed as the CBS in the hardware. For example, if you configure CBS at 4000 bytes and the MTU on an interface is 3000 bytes, when a service policy map is applied on this interface, the CBS programmed in the hardware is 2 × MTU = 6000 bytes.
- 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 2048 service policy maps. Broadcast, unknown unicast, and unknown multicast policies are counted separately.
- You can configure a policer by modifying the remarking values in the default policer remarking profile and then applying the profile to the policer. See Configuring a policer with the default policer remarking profile on page 39 for more information.

Default class map traffic policing

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

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

Default class map traffic policing considerations and limitations

- You can configure up to 2048 policy maps.
- You can configure one or more class maps per policy map.
- Policers are supported for ingress only.
- The number of policers you can configure is dependent on the hardware profile you are using. Refer to the Release Notes for more information.
- Traffic filtered by an ACL is not subject to default service policy policer.
- Match default class map service policy is supported on ingress interfaces. It is also supported on egress interfaces only if the actions are shaper and/or scheduler.
- A class-based service policy and storm-control policy (BUM) are mutually exclusive applications. Only one can be enabled at a time on a given interface.
- 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.

Single-rate, two-color marker

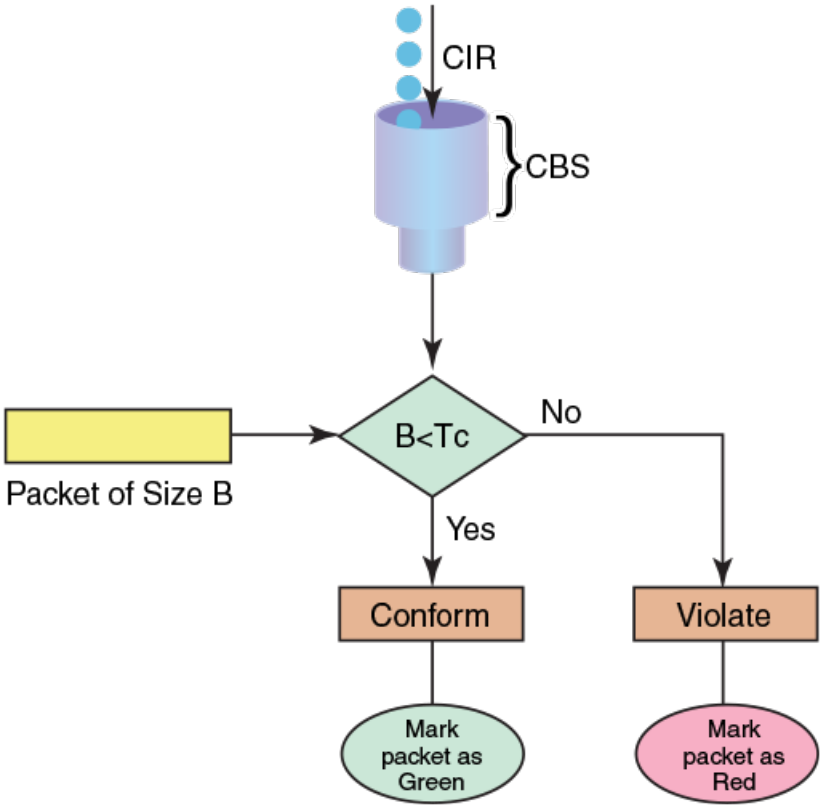
Single-rate, two-color marker meters an IP packet stream and marks its packets either green or red.

Single-rate traffic contract uses the CIR and CBS parameters. Marking is based on CIR and the associated CBS. Packets are marked as follows:

- Green - if it does not exceed the CBS
- Red - otherwise

This marker method 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, two-color marker



Implementation

The method of marking for traffic policing is implemented by tracking the current burst size using token-buckets and discarding packets that exceed the CIR. An incoming burst is classified as either conforming (green) or violating (red).

Every arriving packet is compared to CBS to determine conformance. The drawback of single-rate traffic contracts is that the service provider must be cautious when assigning CIR bandwidth to ensure the less bandwidth is offered than can be serviced at any moment. The reason for this is that not all customers send traffic simultaneously, so network links might effectively become underutilized even at weak spots.

Two-rate, three-color marker

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

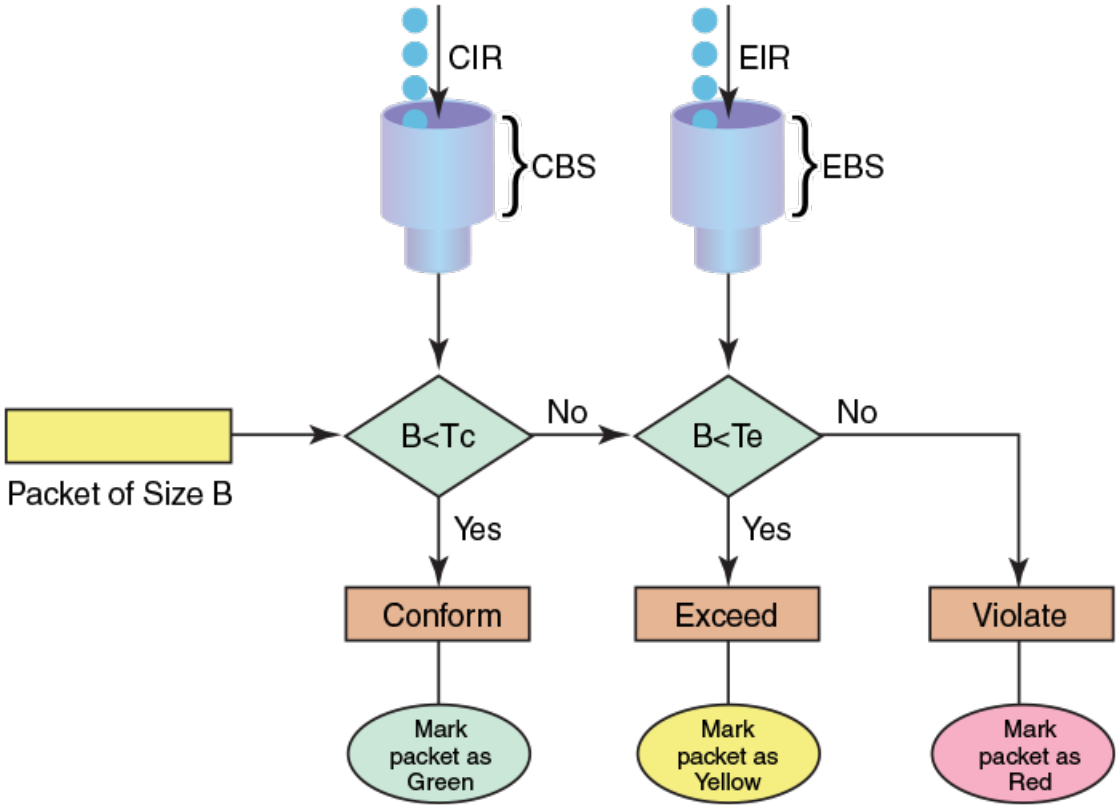
There are four main parameters in a dual-rate traffic contract. CIR, CBS, EIR, and EBS.

Marking is based on CIR. Packets are marked:

- Green - if it does not exceed the CIR.
- Yellow - if it exceeds the CIR, but conforms to the excess information rate (EIR).
- Red - if it exceeds the EIR.

This method is useful for ingress policing of a service, where an excess information 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.

Match access-group — class map policing

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

ACL-based rate limiting is built on top of ACL and policer features, it rate limits the Layer 3 traffic that matches the permit conditions specified in an IPv4 access list. The ACL-based policer feature controls the amount of bandwidth consumed by an individual flow or aggregate of inbound flows by limiting the traffic rate on an individual port according to criteria defined by the **match access-group** class map. This ACL-based rate limiting feature can serve as a hardware solution to prevent DoS attacks.

Match access-group - class map policing rules and limitations

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

- You can configure:
  - 2000 policy maps
  - 50 class maps for each policy map

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

- For protection against:
  - PING attacks
  - TCP Reset attacks
  - TCP SYN attacks
  - UDP attack
- Layer 3 IPv4 and IPv6 ACL-based rate limiting and MAC-based rate limiting are supported.
- ACL-based rate limiting is applicable only to ingress traffic.
- There is one policer per ACL, it applies to all the rules for that ACL
- Control protocols are rate-limited if they match the configured ACL clause.
- When a **match access-group** class map rate limit is applied to a LAG logical port, and all LAG ports belong to the same tower, then MAX CIR value is the interface speed × number of physical ports. For example: if 0/1, 0/2 are LAG member ports, then MAX CIR will be 2 × 10Gbps.
- When a **match access-group** class map rate limit is applied to LAG logical port, MAX rate on that port is the number of the tower in that LAG × CIR. For example: if 0/1, 0/2 , 0/4, 0/5 are LAG member ports, then MAX rate is 3 × CIR.

ACL-based rate limiting use cases

The following describes 4 common DoS attacks and how to protect against them using ACL based rate limiting.

See the topics:

Use case 1 - protection against TCP SYN attacks on page 21 and Configuring use case 1 - protection against TCP SYN attacks on page 26.
Use case 2 - protection against TCP RST attacks on page 21 and Configuring use case 2 - protection against TCP RST attacks on page 28.
Use case 3 - protection against ping flood attacks on page 21 and Configuring use case 3 - protection against ping flood attacks on page 30.
Use case 4 - protection against UDP flood attacks on page 21 and Configuring use case 4 - protection against UDP flood attacks on page 33.

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.

See the topic, Configuring use case 1 - protection against TCP SYN attacks on page 26.

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.

See the topic, Configuring use case 2 - protection against TCP RST attacks on page 28.

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.

See the topic, Configuring use case 3 - protection against ping flood attacks on page 30.

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.

See the topic, Configuring use case 4 - protection against UDP flood attacks on page 33.

Configure all the use cases for ACL traffic filtering

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

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

See the topic, Configuring and applying all four use cases for ACL-based traffic filtering on page 35.

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

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

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

BUM storm control allows you to limit the amount of broadcast, unknown unicast, and multicast ingress traffic on a specified interface. All traffic received in excess of the configured rate is discarded. You also have the option to specify whether to shut down an interface if the maximum defined rate is exceeded within a ten second sampling period. When a port is shut down, you receive a log message.

Storm control considerations and limitations

- BUM storm control applies only to ingress traffic.
- BUM can only be configured on physical interfaces.
- BUM storm control and input service policy are mutually exclusive features. Only one can be enabled at a time on a given interface.
- BUM is not supported on LAG ports or LAG member ports.
- A single rate two color marking scheme is used.
- Metering is performed on the packet size as received on the wire (including IPG and preamble), ignoring CRC.
- If BUM traffic is also classified by an ACL, then BUM rate limiting is not effective
- The configured rate in bits per second (bps) is rounded up to next achievable rate.
- Only FWD and DROP counters are supported:
  - Conformed - Shows FWD packets including green and yellow color packets.
  - Violated - Shows DROP packets including red color packets.
  - Exceed - is always ZERO.
- FWD and DROP counters must use a counter profile other than default.

Configuring traffic policing

Follow these tasks to configure traffic policing.

Configuring a class map using an ACL

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

1. Enter global configuration mode.

```
device# configure terminal
```

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

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

   b) Allow traffic from a specific IP address.

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

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

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

   For details on creating access lists, refer to the *Brocade 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# do show running-config class-map class_1
class-map class_1
 match access-group ip_acl
```

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.

See the topic, Policy maps for policing parameter ranges.

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# do show policy-map detail policy_2

   Policy-Map policy_2
       Class default
         Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

     Bound To:None
   ```

7. Save the configuration.

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

Policy map configuration example

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

Configuring port-based traffic policing

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

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

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode

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

3. Attach an input policy map.

   ```
   device(conf-if-eth-0/3)# service-policy in policy_2
   ```

4. Return to privileged exec mode.

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

5. Verify the configuration.

   ```
   device(conf-if-eth-0/3)# do show policy-map interface ethernet 0/3

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

Port-based traffic policing configuration example

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

Configuring IP-based rate limiting

Configuring use case 1 - protection against TCP SYN attacks

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create an extended IP ACL.

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the ACL.

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

Protection against TCP SYN attacks - ACL configuration example

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

Configuring use case 1 - bind the TCP SYN ACL to an interface

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

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a class map.

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

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the class map to ACL association.

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

6. Create a policy map with a policer.

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

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

7. Associate a class map with the policy map.

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

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

8. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 180000 cbs 50000 eir 36000 ebs 400000
```

   CIR and EIR are in increments of 18000 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 default
      Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

  Bound To:None
```

11. Enter interface configuration mode.

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

12. Bind the policy map to the port.

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

13. Return to privileged exec mode.

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

14. Verify the configuration.

```
device# show policy-map detail policyAclFilter
Policy-Map policyAclFilter
  Class default
      Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

  Bound To: Eth 0/3(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 180000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

Configuring use case 2 - protection against TCP RST attacks

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create or invoke an extended IP ACL.

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the ACL.

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

Protection against TCP RST attacks - ACL configuration example

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

Configuring use case 2 - bind the TCP RST ACL to an interface

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

A TCP RST matching ACL has been configured.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create a class map.

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

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the class map to ACL association.

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

6. Create a policy map with a policer.

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

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

7. Associate a class map with the policy map.

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

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

8. Populate the class map policer parameters.

   ```
   device(config-policymap-class)# police cir 180000 cbs 50000 eir 36000 ebs 400000
   ```

   CIR and EIR are in increments of 18000 bps.

9. Return to privileged exec mode.

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

10. Enter interface configuration mode.

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

11. Verify the configuration.

```
device(conf-if-eth-0/3)# do show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class default
      Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

  Bound To:None
```

12. Bind the policy map to the port.

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

13. Verify the configuration.

```
device(conf-if-eth-0/3)# do show policy-map detail policyAclFilter
Policy-Map policyAclFilter
 Class default
   Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

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

14. Return to privileged exec mode.

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

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 180000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

Configuring use case 3 - protection against ping flood attacks

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create or invoke an extended IP ACL.

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

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

3. Configure the extended ACL to filter ICMP packets.

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

4. Return to privileged exec mode.

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

5. Verify the ACL.

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

Protection against ping attacks - ACL configuration example

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

Configuring use case 3 - bind the ping flood attack ACL to an interface

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

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a class map.

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

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the class map to ACL association.

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

6. Create a policy map with a policer.

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

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

7. Associate a class map with the policy map.

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

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

8. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 180000 cbs 50000 eir 36000 ebs 400000
```

CIR and EIR are in increments of 18000 bps.

9. Enter interface configuration mode.

```
device(config)# interface ethernet 0/33
```

10. Verify the configuration.

```
device(conf-if-eth-0/3)# do show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class default
      Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

  Bound To:None
```

11. Bind the policy map to the port.

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

12. Verify the configuration.

```
device# do show policy-map detail policyAclFilter
Policy-Map policyAclFilter
  Class default
       Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

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

13. Return to privileged exec mode.

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

14. 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 180000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

Configuring use case 4 - protection against UDP flood attacks

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

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create or invoke an extended IP ACL.

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

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

3. Configure the extended ACL to filter UDP packets.

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

4. Return to privileged exec mode.

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

5. Verify the ACL.

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

Protection against UDP flood attacks - ACL configuration example

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

Configuring use case 4 - bind the UDP ACL to an interface

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

You have configured an extended Layer 3 UDP ACL.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a class map.

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

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the class map to ACL association.

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

6. Create a policy map with a policer.

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

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

7. Associate a class map with the policy map.

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

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

8. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 180000 cbs 50000 eir 36000 ebs 400000
```

   CIR and EIR are in increments of 18000 bps.

9. Return to privileged exec mode.

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

10. Enter interface configuration mode.

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

11. Verify the configuration.

```
device(conf-if-eth-0/3)# do show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class default
      Police cir 1000000000 eir 1000000000 classification-type color remark-profile default

  Bound To:None
```

12. Bind the policy map to the port.

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

13. Return to privileged exec mode.

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

14. Verify the configuration.

```
device# show policy-map detail policyAclFilter
Policy-Map policyAclFilter
 Class aclFilter
 Police cir 180000 cbs 50000 eir 36000 ebs 400000 classification-type color remark-profile default

 Bound To: Et 0/3(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 180000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

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

Follow these steps to apply ACLs for traffic filtering.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create an ACL.

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

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

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

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

   This step provides protection from TCP SYN attacks.

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

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

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

   This step provides protection from TCP RST attacks.

5. Configure the extended ACL to filter ICMP packets.

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

   This step protects against ping flood attacks.

6. Configure the extended ACL to filter UDP packets.

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

   This step protects against UDP flood attacks.

7. Return to global configuration mode.

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

8. Verify the ACL.

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

9. Create a class map.

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

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

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

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

11. Return to global configuration mode.

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

12. Verify the class map to ACL association.

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

13. Create a policy map with a policer.

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

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

14. Associate a class map with the policy map.

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

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

15. Populate the class map policer parameters.

```
device(config-policymap-class)# police cir 180000 cbs 50000 eir 36000 ebs 400000
```

CIR and EIR are in increments of 18000 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 4000000 cbs 50000 eir 800000 ebs 400000 classification-type color remark-profile
default

  Bound To:None
```

18. Enter global configuration mode.

```
device# configure terminal
```

19. Enter interface configuration mode.

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

20. Bind the policy map to the port.

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

21. Return to privileged exec mode.

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

22. Verify the configuration.

```
device# show policy-map detail policyAclFilter

Policy-Map policyAclFilter
    Class aclFilter
      Police cir 180000 cbs 50000 eir 36000 ebs 400000 classification-type color remark-profile
default

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

23. Save the configuration.

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

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

```
device# configure terminal
device(config)# ip access-list extended acl1
device(conf-ipacl-ext)# permit tcp any any sync
device(conf-ipacl-ext)# permit tcp any any rst
device(conf-ipacl-ext)# permit icmp any any
device(conf-ipacl-ext)# permit udp any any
device(config)# do show running-config ip access-list extended acl1
device(config)# class-map aclFilter
device(config-classmap)# match access-group acl1
device(config-classmap)# exit
device(config)# do show running-config class-map aclFilter
device(config)# policy-map policyAclFilter
device(config-policymap)# class aclFilter
device(config-policymap-class)# police cir 180000 cbs 50000 eir 36000 ebs 400000
device(config-policymap-class-police)# end
device# show policy-map detail policyAclFilter
device# configure terminal
device(config)# interface ethernet 0/3
device(conf-if-eth-0/3)# service-policy in policyAclFilter
device(conf-if-eth-0/3)# end
device# show policy-map detail policyAclFilter
device# copy running-config startup-config
```

Configuring storm control

This example configures BUM storm control on an Ethernet interface.

1. Enter global configuration mode.

```
device# configure terminal
```

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

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

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

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

   In this example you set a control on the inbound broadcast traffic, limiting the rate to 400000 bits per second (bps), with a parameter set to shutdown the port in case of a limit violation.

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

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

   In this example you set a control on the inbound unknown-unicast traffic, limiting the rate to 50000000 bps, with a parameter set to monitor the port in case of a limit violation.

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

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

   In this example you set a control on the inbound multicast traffic, limiting the rate to 3% of traffic, with a parameter set to monitor the port in case of a limit violation.

6. Return to privileged exec mode.

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

7. Verify the storm control configuration.

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

8. Save the configuration.

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

BUM storm control configuration example

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

Configuring a policer with the default policer remarking profile

To configure the settings for a policer, you modify the default policer remarking profile and apply it to a policer.

To use the default policer remarking profile to configure a policer, perform the following procedures:

- Modifying the color classification type in the default policer remarking profile on page 40.

  In this procedure, you use the **police-remark-profile**, **action**, and **set** commands.

- Modifying the color-and-cos classification type in the default policer remarking profile on page 40.

  In this procedure, you use the **police-remark-profile**, **action**, and **map** commands.

- Modifying the color-and-dscp classification type in the default policer remarking profile on page 41.

  In this procedure, you use the **police-remark-profile**, **action**, and **map** commands.

- Modifying the color-and-traffic-class classification type in the default policier remarking profile on page 42.

In this procedure, you use the **police-remark-profile**, **action**, and **map** commands.

- Applying the default policer remarking profile to a policer on page 42.

  In this procedure, you use the **policy-map**, **class**, and **police cir** commands.

- Verifying the policer remarking configuration on page 43.

  In this procedure, you use the **show policy-map** command.

Modifying the color classification type in the default policer remarking profile

Follow these steps to modify the color classification type of the default profile.

The default policer remarking profile was created during initialization. Only the default policer remarking profile can be modified; you cannot create a separate profile.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Indicate the default profile.

```
device(config)# police-remark-profile default
```

3. Specify that you are modifying the color classification type for conforming traffic.

```
device(police-remark-profile-default)# action color conform
```

   (Alternatively, you can specify the color classification type for exceeding traffic by substituting "conform" with "exceed.")

4. Set the remarked values for the color classification type.

```
device(police-remark-profile-default)# set cos 3
device(police-remark-profile-default)# set traffic-class 5
device(police-remark-profile-default)# set dscp 10
```

Color classification type modification example

```
device# configure terminal
device(config)# police-remark-profile default
device(police-remark-profile-default)# action color conform
device(police-remark-profile-default)# set cos 3
device(police-remark-profile-default)# set traffic-class 5
device(police-remark-profile-default)# set dscp 10
```

Modifying the color-and-cos classification type in the default policer remarking profile

Follow these steps to modify the color-and-cos classification type of the default profile.

The default policer remarking profile was created during initialization. Only the default policer remarking profile can be modified; you cannot create a separate profile. Before using this procedure, create and bind policy maps that contain the remarking values you need.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Indicate the default profile.

```
device(config)# police-remark-profile default
```

3. Specify that you are modifying the color-and-cos classification type for conforming traffic.

```
device(police-remark-profile-default)# action color-and-cos conform
```

(Alternatively, you can specify the color-and-cos classification type for exceeding traffic by substituting "conform" with "exceed.")

4. Specify the maps that contain the remarking values for the traffic. (In this example, "cm1," "ct1," and "cd1" are sample map names. You specify your particular set of map names.)

```
device(police-remark-profile-default)# map cos-mutation cm1
device(police-remark-profile-default)# map cos-traffic-class ct1
device(police-remark-profile-default)# map cos-dscp cd1
```

Color-and-cos classification type modification example

```
device# configure terminal
device(config)# police-remark-profile default
device(police-remark-profile-default)# action color-and-cos conform
device(police-remark-profile-default)# map cos-mutation cm1
device(police-remark-profile-default)# map cos-traffic-class ct1
device(police-remark-profile-default)# map cos-dscp cd1
```

Modifying the color-and-dscp classification type in the default policer remarking profile

Follow these steps to modify the color-and-dscp classification type of the default profile.

The default policer remarking profile was created during initialization. Only the default policer remarking profile can be modified; you cannot create a separate profile. Before using this procedure, create and bind policy maps that contain the remarking values you need.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Indicate the default profile.

```
device(config)# police-remark-profile default
```

3. Specify that you are modifying the color-and-dscp classification type for conforming traffic.

```
device(police-remark-profile-default)# action color-and-dscp conform
```

(Alternatively, you can specify the color-and-dscp classification type for exceeding traffic by substituting "conform" with "exceed.")

4. Specify the maps that contain the remarking values for the traffic. (In this example, "dm1," "dc1," and "dt1" are sample map names. You specify your particular set of map names.)

```
device(police-remark-profile-default)# map dscp-mutation dm1
device(police-remark-profile-default)# map dscp-cos dc1
device(police-remark-profile-default)# map dscp-traffic-class dt1
```

Color-and-dscp classification type modification example

```
device# configure terminal
device(config)# police-remark-profile default
device(police-remark-profile-default)# action color-and-dscp conform
device(police-remark-profile-default)# map dscp-mutation dm1
device(police-remark-profile-default)# map dscp-cos dc1
device(police-remark-profile-default)# map dscp-traffic-class dt1
```

Modifying the color-and-traffic-class classification type in the default policier remarking profile

Follow these steps to modify the color-and-traffic-class classification type of the default profile.

The default policer remarking profile was created during initialization. Only the default policer remarking profile can be modified; you cannot create a separate profile. Before using this procedure, create and bind policy maps that contain the remarking values you need.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Indicate the default profile.

   ```
   device(config)# police-remark-profile default
   ```

3. Specify that you are modifying the color-and-traffic-class classification type for conforming traffic.

   ```
   device(police-remark-profile-default)# action color-and-traffic-class conform
   ```

   (Alternatively, you can specify the color-and-traffic-class classification type for exceeding traffic by substituting "conform" with "exceed.")

4. Specify the maps that contain the remarking values for the traffic. (In this example, "tm1," "tc1," and "td1" are sample map names. You specify your particular set of map names.)

   ```
   device(police-remark-profile-default)# map traffic-class-mutation tm1
   device(police-remark-profile-default)# map traffic-class-cos tc1
   device(police-remark-profile-default)# map traffic-class-dscp td1
   ```

Color-and-traffic-class classification type modification example

```
device# configure terminal
device(config)# police-remark-profile default
device(police-remark-profile-default)# map traffic-class-mutation tm1
device(police-remark-profile-default)# map traffic-class-cos tc1
device(police-remark-profile-default)# map traffic-class-dscp td1
```

Applying the default policer remarking profile to a policer

After you modify the policer remarking default profile, you must apply it to a policer.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter policy map configuration mode.

   In this example, "p1" is used for the policy map name. You specify your policy map name.

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

3. Specify the default policer remarking profile.

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

4. Specify the classification type that you want to include in the policer.

   ```
   device(config-policymap-class)# police cir10000000 eir200000000
                                    classification-type color-and-cos remark-profile default
   ```

Default policer remarking profile application example

```
device# configure terminal
device(config)# policy-map p1
device(config-policymap)# class default
device(config-policymap-class)# police cir10000000 eir200000000
                                 classification-type color-and-cos remark-profile default
```

Verifying the policer remarking configuration

After you apply the default policer remarking profile to a policer, you can verify the configuration of the policer with the following procedure.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Display the policy map configuration for the appropriate port.

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

   The following information is displayed:

   ```
   Ingress Direction:
   Policy-Map p1
   Class default
   Police cir10000000 eir200000000 classification-type color-and-cos remark-profile default
   Stats:
   Operational cir:10000000 cbs:1499999 eir:200000000 ebs:29999999
   Conform Byte:3455765768 Exceed Byte:434456546 Violate Byte:654768799
   ```

3. Display the policer details.

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

   The following information is displayed:

   ```
   Policy-Map p1
   Class default
   Police cir10000000 eir200000000 classification-type color-and-cos remark-profile default
   Bound To: Eth 0/3(in)
   ```

Policer remarking configuration verification example

```
device# configure terminal
device(config)# show policy-map interface ethernet 0/3

Ingress Direction:
Policy-Map p1
Class default
Police cir10000000 eir200000000 classification-type color-and-cos remark-profile default
Stats:
Operational cir:10000000 cbs:1499999 eir:200000000 ebs:29999999
Conform Byte:3455765768 Exceed Byte:434456546 Violate Byte:654768799

device(config)# show policy-map detail p1

Policy-Map p1
Class default
Police cir10000000 eir200000000 classification-type color-and-cos remark-profile default

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

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 Brocade 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 for multicast traffic

While managing multicast traffic, consider the following:

- There are eight multicast queues for multicast traffic.
- The Traffic Manager (TM) maps incoming packets to these queues based on the traffic class (TC) or drop precedence (DP) received from the packet processor (PP).
- The ingress TM pushes multicast traffic either by strict priority (SP) or by mixed SP and weighted priority.
- The egress TM maps these multicast packets to the egress queues (EGQ).
- EGQs share memory from a 2.4 MB pool.

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 4 Default DSCP mappings

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

Congestion control

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

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

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

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 Brocade 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 Brocade 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:

Statistical Average-Q-Size - The statistical average size of the queue calculated over time on the device.

Current-Q-Size - The current size of the queue as calculated on the device.

Wq - This variable specifies the weights that should be given to the current queue size and the statistical average-q-size when calculating the size for WRED calculations.

Max-Instantaneous-Q-Size - The maximum size up to which a queue is allowed to grow. Packets that cause the queue to grow beyond this point are unconditionally dropped. This variable is user configured.

Min-Average-Q-Size - The average queue size below which all packets are accepted. This variable is user configured.

Max-Average-Q-Size - The average queue size above which all packets are dropped. This variable is user configured.

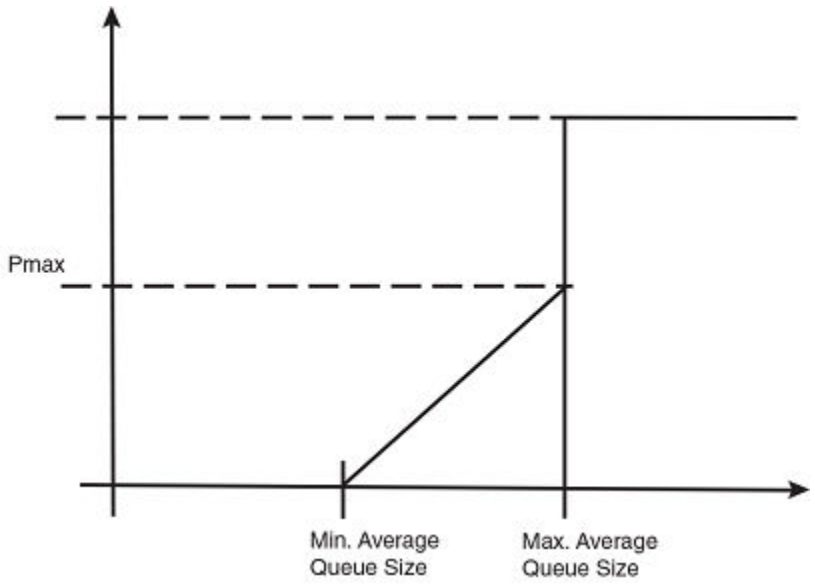
Pmax - The maximum drop probability when queue-size is at Max-Average-Q-Size. This variable is user configured.

Pkt-Size-Max - The packet size to which the current packet's size is compared as shown in the algorithm below. This variable is user configured.

How WRED works

The WRED operation graph below describes the interaction of the previously described variables in the operation of WRED. When a packet arrives at a device, the average queue size (*avg-q-size*) is calculated as described below (note that this is not the statistical average queue size). If *avg-q-size* as calculated, is below the configured Min. Average Queue Size, then the packet is accepted. If the average queue size is above the Max. configured Average Queue Size threshold, the packet is dropped. If the instantaneous queue size exceeds the value configured for the Max-Instantaneous-Q-Size, the packet is dropped. If the Average Queue size falls between the Min. Average Queue Size and the Max. Average Queue Size, packets are dropped according to the calculated probability described below.

FIGURE 3 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 *Wq* value is instrumental to the calculation and can be:

- equal to the statistical average queue size (*Wq* == 0), or
- equal to the current queue size (*Wq* == 1) or
- be between 0 and 1 (0 < *Wq* < 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 *Wq* 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 *Wq* 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 = (pkt-size ÷ pkt-size-max) × Pmax × [(avg-q-size - min-avg-q size) ÷ (max-avg-q-size - min-avg-q size)]
```

Apply WRED

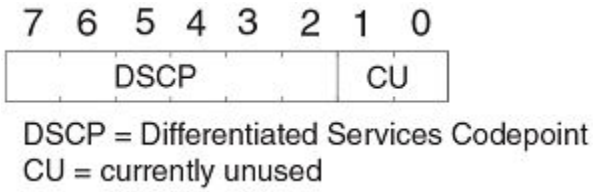
Packets are assigned to an ingress queue type based on their individual destination port and one of the 8 (0 - 7) internal priorities. Each of these priorities is assigned a queue type from 0 - 7 according to the internal priority it belongs to.

TABLE 5 Internal priority to queue type mapping

Internal priority	0	1	2	3	4	5	6	7
Queue type	0	1	2	3	4	5	6	7

The WRED algorithm is applied to traffic on these individual queues based upon parameters configured for its assigned queue type. When traffic arrives at a queue, it is passed or dropped as determined by the WRED algorithm. Packets in an individual queue are further differentiated by one of four drop precedence values which are determined by the value of bits 3:2 of the DSCP bits in the IPv4 or IPv6 packet header.

FIGURE 4 DSCP bits in packet header



The user configurable values applied per queue type and per drop precedence value are:

- Maximum Drop Probability
- Minimum and Maximum Average Queue Size
- Maximum Packet Size

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 specified period of time. This feature is available per port in all front ports and applies to all the traffic on the link. The Brocade device supports the receive direction only and supports pause flow control and priority flow control.

By default, the feature is disabled for both directions.

See the procedure, Configuring link level flow control on page 72 for the steps to configure LLFC.

Scheduling

Scheduling arbitrates among multiple queues waiting to transmit a frame.

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

Scheduling types

TABLE 6 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.

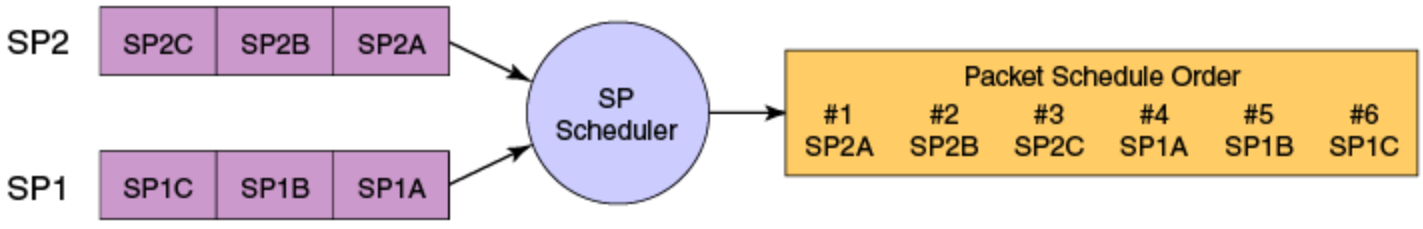
QoS strict priority egress traffic scheduling

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 Brocade devices classify packets into one of eight internal 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.

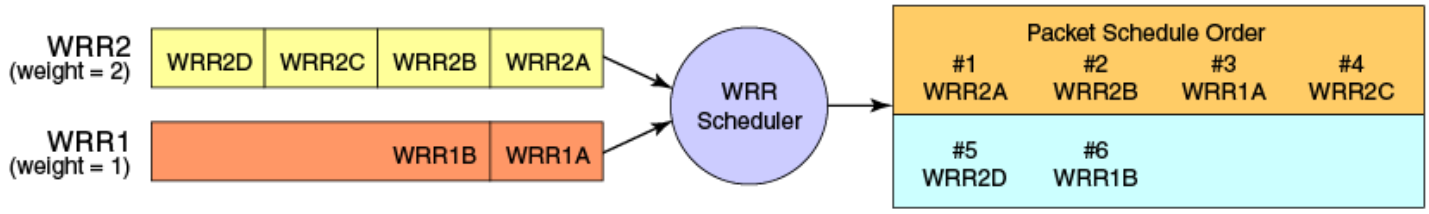
Weighted round robin egress traffic scheduling

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.

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 7 Multicast traffic class equivalence mapping

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

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

Control protocol packet prioritization

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

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

TABLE 8 Default prioritized protocol table

Protocol Packets
ARP
BFD (Bidirectional Forwarding Detection)
BGP / BGP in 6to4
BGP / BGP over GRE
BOOTP/DHCP
ES-IS
GARP
IGMP
IPv4 Router Alert
IPv4/L2

TABLE 8 Default prioritized protocol table (continued)

Protocol Packets
IPv6
Keep Alive Packets
LACP
LDP basic
LDP extended
MLD
MRP
MSDP / MSDP over GRE
ND6 / ND6 in 6to4
OSPF / OSPF in 6to4
OSPF / OSPF over GRE
RIP
RIPNG
RSVP
STP/RSTP/BPDU
VRRP
VRRPE
VSRP
Y.1731

Configure flow-based QoS

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

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

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

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

Match access-group — class map policing

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

ACL-based rate limiting is built on top of ACL and policer features, it rate limits the Layer 3 traffic that matches the permit conditions specified in an IPv4 access list. The ACL-based policer feature controls the amount of bandwidth consumed by an individual flow or aggregate of inbound flows by limiting the traffic rate on an individual port according to criteria defined by the **match access-group** class map. This ACL-based rate limiting feature can serve as a hardware solution to prevent DoS attacks.

Match access-group - class map policing rules and limitations

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

- You can configure:
  - 2000 policy maps
  - 50 class maps for each policy map

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

- For protection against:
  - PING attacks
  - TCP Reset attacks
  - TCP SYN attacks
  - UDP attack
- Layer 3 IPv4 and IPv6 ACL-based rate limiting and MAC-based rate limiting are supported.
- ACL-based rate limiting is applicable only to ingress traffic.
- There is one policer per ACL, it applies to all the rules for that ACL
- Control protocols are rate-limited if they match the configured ACL clause.
- When a **match access-group** class map rate limit is applied to a LAG logical port, and all LAG ports belong to the same tower, then MAX CIR value is the interface speed × number of physical ports. For example: if 0/1, 0/2 are LAG member ports, then MAX CIR will be 2 × 10Gbps.
- When a **match access-group** class map rate limit is applied to LAG logical port, MAX rate on that port is the number of the tower in that LAG × CIR. For example: if 0/1, 0/2 , 0/4, 0/5 are LAG member ports, then MAX rate is 3 × CIR.

Policy maps

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

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

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

When using the traffic policing policies available from previous versions, the policy parameters are provided explicitly for each port during port configuration. In this version, the policies must be defined using a policy map. One policy map can be specified per service policy. You can configure up to 2000 policy maps per system.

Policy map configuration rules

Follow these rules when configuring traffic policing:

- A policer map (policy map or class map) name must be unique among all maps of that type.
- A policer name must begin with a-z or A-Z . An underscore, hyphen, and numeric values 0-9 can be used in the body of the name but not as the first character.
- You can configure a policer by modifying the remarking values in the default policer remarking profile and then applying the profile to the policer. See Configuring a policer with the default policer remarking profile on page 39 for more information.
- You can configure a maximum of 2000 policy maps.
- ACL-based class maps and default class maps can be used in a single policy map.

- Flow-based QoS is not supported in the egress direction. You can apply QoS to the flow using the policy map on the interface only in the ingress direction.
- For an ingress or egress service policy, one default class map can be specified per policy map.
- For an ingress-only service policy, 50 class maps, including the default class map, are supported per policy map.
- Broadcast, unknown unicast, and multicast (BUM) policies are counted separately.
- You cannot delete a policy map if is referenced in an active service policy (applied on an interface).

QoS shaping rate

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

QoS ingress data buffer management

Buffer management consists of the following.

- Packets arrived at ingress are stored in a 24 MB data buffer (DB).
- 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.
  - The non-guaranteed BD and DB are allocated from a shared pool.

QoS maps

The SLX-OS device provides support for the following QoS maps:

- Default maps for all map types created by default as soon as switch comes up.
- Layer 2 maps including the cos-mutation and cos-traffic class maps that must be applied on a port. Both cos-mutation and cos-traffic-class maps must be applied for a Layer 2 QoS map to work.
- Layer 3 maps including the dscp-cos, dscp-traffic class, dscp mutation maps that must be applied on a port. All three of these maps must to be applied for a Layer 3 QoS map to work.

The device also provides for the following maps that can be used in a remarking profile only:

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

For applying these maps to a remarking profile, see the *Traffic Policing* chapter.

Default Maps

By default, as soon as device comes up, maps for each map type are created with the names of default and all-zero-map.

The default and all-zero-map maps for Layer 2 and Later 3 maps can be applied on interfaces with other user-defined maps. However, they cannot be deleted or modified. The CoS-to-DSCP, TC Mutation map, TC-to-CoS, and TC-to-DSCP maps cannot be applied on interfaces. They can be used only in a policer-remark profile.

You can display the default maps by using the **show qos maps** command.

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

  CoS Mutation Map: all-zero-map
            In-CoS: 0  1  2  3  4  5  6  7
      ------------------------------------
           Out-CoS: 0  0  0  0  0  0  0  0

      Enabled on the following interfaces:

  CoS Mutation Map: default
            In-CoS: 0  1  2  3  4  5  6  7
      ------------------------------------
           Out-CoS: 0  1  2  3  4  5  6  7

      Enabled on the following interfaces:

  CoS-to-TC Map: all-zero-map
            In-CoS: 0  1  2  3  4  5  6  7
      ------------------------------------
            Out-TC: 0  0  0  0  0  0  0  0
            Out-DP: 0  0  0  0  0  0  0  0

      Enabled on the following interfaces:

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

      Enabled on the following interfaces:

  CoS-to-DSCP Map: all-zero-map
            In-CoS:  0   1   2   3   4   5   6   7
      ---------------------------------------------
          Out-DSCP: 00  00  00  00  00  00  00  00

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

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

  TC Mutation Map: default
             In-TC: 0  1  2  3  4  5  6  7
      ------------------------------------
            Out-TC: 0  1  2  3  4  5  6  7
```

```
TC-to-DSCP Map: all-zero-map
            In-TC:  0   1   2   3   4   5   6   7
    ------------------------------------------------
         Out-DSCP: 00  00  00  00  00  00  00  00

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

DSCP Mutation Map: all-zero-map (DSCP = d1d2)
    d1 :  d2 0  1  2  3  4  5  6  7  8  9
    -------------------------------------
     0 :    00 00 00 00 00 00 00 00 00 00
     1 :    00 00 00 00 00 00 00 00 00 00
     2 :    00 00 00 00 00 00 00 00 00 00
     3 :    00 00 00 00 00 00 00 00 00 00
     4 :    00 00 00 00 00 00 00 00 00 00
     5 :    00 00 00 00 00 00 00 00 00 00
     6 :    00 00 00 00

    Enabled on the following interfaces:

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

    Enabled on the following interfaces:

DSCP-to-TC Map: all-zero-map (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 0/0 0/0
     1 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
     2 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
     3 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
     4 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
     5 :     0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0 0/0
     6 :     0/0 0/0 0/0 0/0

    Enabled on the following interfaces:

DSCP-to-TC Map: default (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:

DSCP-to-CoS Map: all-zero-map (DSCP = d1d2)
    d1 :  d2 0  1  2  3  4  5  6  7  8  9
    -------------------------------------
     0 :    00 00 00 00 00 00 00 00 00 00
     1 :    00 00 00 00 00 00 00 00 00 00
     2 :    00 00 00 00 00 00 00 00 00 00
     3 :    00 00 00 00 00 00 00 00 00 00
     4 :    00 00 00 00 00 00 00 00 00 00
     5 :    00 00 00 00 00 00 00 00 00 00
     6 :    00 00 00 00
```

```
    Enabled on the following interfaces:

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

    Enabled on the following interfaces:
```

Layer 2 QoS maps

The SLX-OS device supports the Layer 2 cos-mutation and cos-traffic class maps.

The device support eight unique combinations of cos-mutation and cos-traffic class maps. You can apply these maps on multiple ports.

Both cos-mutation and cos-traffic class maps must be applied on a port, otherwise the map configuration is not active. If you want to have a user-defined cos-mutation configuration and default cos-traffic class configuration on a port, you must explicitly apply the default cos-traffic class map on a port, otherwise the cos-mutation configuration is not active.

Layer 3 QoS maps

The SLX-OS device supports the Layer 3 dscp-cos, dscp-traffic class, and dscp mutation maps.

The device support eight unique combinations of dscp mutation, dscp-traffic class, and dscp-cos maps. These maps can be applied on multiple ports.

All DSCP map types must be applied on a port, otherwise the map configuration is not active. If you want to have default configuration on a port for one of the Layer 3 maps, you must explicitly apply the default configuration of the Layer 3 map type.

Ingress QoS mutation

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

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

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

Configuring QoS

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

Configuring QoS CoS-to-CoS mutation mapping

Follow these tasks to configure QoS CoS-to-CoS mutation mapping.

Configuring a CoS-to-CoS mutation map

Follow these steps to configure QoS CoS-to-CoS mutation maps.

1. Enter global configuration mode.

```
device# configure terminal
```

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

```
device(config)# qos map cos-mutation cos_map_1
```

3. Configure the CoS-to-CoS map value.

```
device(cos-mutation-cos_map_1)# map cos 2 to cos 4
```

4. Return to privileged exec mode.

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

5. Verify the configuration.

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

Cos-to-Cos Mutation map  cos_map_1
    In-Cos: 0  1  2  3  4  5  6  7
   -------------------------------
   Out-Cos: 0  0  4  0  0  0  0  0

   Enabled on the following interfaces:
```

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

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

QoS CoS-to-CoS mutation map configuration example

```
device# configure terminal
device(config)# qos map cos-mutation cos_map_1
device(cos-mutation-cos_map_1)# map cos 2 to cos 4
device(config)# end
device# show qos maps cos-mutation
device# copy running-config startup-config
```

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

Follow these steps to apply a CoS-to-CoS mutation map to either an ingress or egress interface.

You have created a QoS CoS-to-CoS mutation map to apply.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

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

3. Apply the map to the interface.

```
device(conf-if-eth-0/5)# qos cos-mutation cos_map_1
```

4. Return to privileged exec mode.

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

5. Verify the configuration.

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

Cos-to-Cos Mutation map  cos_map_1
    In-Cos: 0  1  2  3  4  5  6  7
   -------------------------------
   Out-Cos: 0  0  4  0  0  0  0  0

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

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

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

QoS CoS-to-CoS mutation map applied to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/5
device(conf-if-eth-0/5)# qos cos-mutation cos_map_1
device(conf-if-eth-0/5)# end
device# show qos maps cos-mutation
device# copy running-config startup-config
```

Configuring CoS-to-traffic class mappings

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

Configuring a CoS-to-traffic class mutation map

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

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

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

1. Enter global configuration mode.

```
device# configure terminal
```

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the configuration.

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

 Cos-to-TC map: cosTCMap
             In-CoS: 0  1  2  3  4  5  6  7
        ------------------------------------
             Out-TC: 0  1  2  3  3  6  6  6
             Out-DP: 0  0  0  0  0  0  0  0

        Enabled on the following interfaces:
```

QoS CoS-to-traffic class map configuration example

```
device# configure terminal
device(config)# qos map cos-traffic-class cosTCMap
device(config-cos-traffic-class-cosTCMap)# map cos 4 to traffic-class 3
device(config-cos-traffic-class-cosTCMap)# map cos 5 to traffic-class 5
device(config-cos-traffic-class-cosTCMap)# map cos 6 to traffic-class 6
device(config-cos-traffic-class-cosTCMap)# map cos 7 to traffic-class 6
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.

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

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 ingress interface to effect the priority remapping. This feature only maps the incoming priority to outgoing priority.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode..

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

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

```
device(conf-if-eth-0/5)# qos cos-traffic-class cosTCMap
```

4. Return to privileged exec mode.

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

5. Verify the configuration.

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

 Cos-to-TC map: cosTCMap
             In-CoS: 0  1  2  3  4  5  6  7
      ------------------------------------
             Out-TC: 0  1  2  3  3  6  6  6
             Out-DP: 0  0  0  0  0  0  0  0

          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 CoS-to-traffic class mutation map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/5
device(conf-if-eth-0/5)# qos cos-traffic-class cosTCMap
device(conf-if-eth-0/5)# end
device# show qos maps cos-traffic-class cosTCMap
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.

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
  [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]

  DSCP Mutation Map: dscpMap (DSCP = d1d2)
      d1 :  d2 0  1  2  3  4  5  6  7  8  9
      ------------------------------------
       0 :    00 00 00 00 00 00 00 00 00 00
       1 :    00 00 00 00 00 00 00 00 00 00
       2 :    00 00 00 00 50 00 00 00 00 00
       3 :    00 00 00 35 00 00 00 00 00 00
       4 :    00 00 00 00 00 00 00 00 00 00
       5 :    00 00 00 61 00 00 00 00 00 00
       6 :    40 00 00 00

       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 ingress interface

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

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

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

This configuration is effective only when the dscp-cos and dscp-traffic-class maps are also applied to same interface.

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 map dscp-mutation dscpMap
  [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]

  DSCP Mutation Map: dscpMap (DSCP = d1d2)
      d1 :  d2 0  1  2  3  4  5  6  7  8  9
      ----------------------------------------
       0 :    00 00 00 00 00 00 00 00 00 00
       1 :    00 00 00 00 00 00 00 00 00 00
       2 :    00 00 00 00 50 00 00 00 00 00
       3 :    00 00 00 35 00 00 00 00 00 00
       4 :    00 00 00 00 00 00 00 00 00 00
       5 :    00 00 00 61 00 00 00 00 00 00
       6 :    40 00 00 00

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

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

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

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

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

Configuring DSCP-to-CoS mappings

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

Configuring a DSCP-to-CoS mutation map

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

1. Enter global configuration mode.

```
device# configure terminal
```

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

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

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

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

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

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

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

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the configuration.

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


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

   Enabled on the following interfaces:
```

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

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

QoS DSCP-to-CoS mutation map configuration example

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

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

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

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

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

3. Enable the DSCP mutation map on the interface.

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

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

4. Return to privileged exec mode.

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

5. Verify the configuration.

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


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


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

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

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

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

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

Configuring DSCP-to-traffic class mappings

Configuring a DSCP-to-traffic class mutation map globally

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

1. Enter global configuration mode.

```
device# configure terminal
```

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

4. Return to privileged exec mode.

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

5. Verify the configuration.

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

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

      Enabled on the following interfaces:
```

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

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

QoS DSCP to traffic class mutation map global 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(dscp-traffic-class-dscpTcMap)# end
device# show qos maps dscp-traffic-class dscpTcMap
device# copy running-config startup-config
```

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

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

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

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

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

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

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

4. Return to privileged exec mode.

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

5. Verify the configuration

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

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

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

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

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

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

Configuring a CoS-to-DSCP mutation map

Follow these steps to configure QoS CoS-to-DSCP mutation map.

1. Enter global configuration mode.

```
device# configure terminal
```

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

```
device(config)# qos map cos-dscp cosDSCP1
```

3. Map the ingress CoS values to egress DSCP values.

```
device(cos-dscp-cosDSCP1)# map cos 2 to dscp 24
```

4. Return to privileged exec mode.

```
device(cos-dscp-cosDSCP1)# end
```

5. Verify the configuration.

```
device# show qos maps cos-dscp cosDSCP1
 [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]

  CoS-to-DSCP Map: test1
             In-CoS:  0   1   2   3   4   5   6   7
         ----------------------------------------------
              Out-DSCP: 00  08  24  24  32  40  48  56
```

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

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

CoS-to-DSCP mutation map configuration example

```
device# configure terminal
device(config)# qos map cos-dscp cosDSCP1
device(cos-dscp-cosDSCP1)# map cos 2 to dscp 24
device(cos-dscp-cosDSCP1)# end
device# show qos maps cos-dscp cosDSCP1
device# copy running-config startup-config
```

Apply this map to a remarking profile only. For more information on the policer remarking profile, refer to the *Traffic Policing* chapter.

Configuring a traffic class mutation map

Follow these steps to configure QoS traffic class mutation map.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a QoS traffic class mutation map.

```
device(config)# qos map traffic-class-mutation tcm1
```

3. Map the ingress traffic class value to an egress traffic class value.

```
device(traffic-class-mutation-tcm1)# map traffic-class 4 to traffic-class 0
```

4. Return to privileged exec mode.

```
device(traffic-class-mutation-tcm1)# end
```

5. Verify the configuration.

```
device# show qos maps traffic-class-mutation tcm1
   [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]

  TC Mutation Map: tcm1
             In-TC: 0  1  2  3  4  5  6  7
         ------------------------------------
            Out-TC: 0  1  2  3  0  5  6  7
```

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

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

Traffic class mutation map configuration example

```
device# configure terminal
device(config)# qos map traffic-class-mutation tcm1
device(traffic-class-mutation-tcm1)# map traffic-class 4 to traffic-class 0
device(traffic-class-mutation-tcm1)# end
device# show qos maps traffic-class-mutation tcm1
device# copy running-config startup-config
```

Apply this map to a remarking profile only. For more information on the policer remarking profile, refer to the *Traffic Policing* chapter.

Configuring a traffic class-to-CoS mutation map

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a QoS traffic class-to-CoS mutation map.

```
device(config)# qos map traffic-class-cos tc1
```

3. Map ingress traffic class values to egress CoS values.

```
device(traffic-class-cos-tc1)# map traffic-class 5 to cos 6
```

4. Return to privileged exec mode.

```
device(traffic-class-cos-tc1)# end
```

5. Verify the configuration.

```
device# show qos maps traffic-class-cos tc1
  [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]

  TC-to-CoS Map: test1
             In-TC: 0  1  2  3  4  5  6  7
         ------------------------------------
         Out-CoS(DP=0): 0  1  2  3  4  6  6  7
         Out-CoS(DP=1): 0  1  2  3  4  6  6  7
         Out-CoS(DP=2): 0  1  2  3  4  6  6  7
         Out-CoS(DP=3): 0  1  2  3  4  6  6  7
```

6. 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 tc1
device(traffic-class-cos-tc1)# map traffic-class 5 to cos 6
device(traffic-class-cos-tc1)# end
device# show qos maps traffic-class-cos tc1
device# copy running-config startup-config
```

Apply this map to a remarking profile only. For more information on the policer remarking profile, refer to the *Traffic Policing* chapter.

Configuring a traffic class-to-DSCP mutation map

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

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Create a QoS traffic class-to-DSCP mutation map.

   ```
   device(config)# qos map traffic-class-dscp tcdscp1
   ```

3. Map ingress traffic class values to egress DSCP values.

   ```
   device(traffic-class-dscp-tcdscp1)# map traffic-class 4 to dscp 55
   ```

4. Return to privileged exec mode.

   ```
   device(traffic-class-dscp-tcdscp1)# end
   ```

5. Verify the configuration.

   ```
   device# show qos maps traffic-class-dscp tcdscp1
     [Note: CoS = Class of Service, TC = Traffic Class, DP = Drop Precedence]

     TC-to-DSCP Map: tcdscp1
                In-TC:  0   1   2   3   4   5   6   7
         ----------------------------------------------
              Out-DSCP: 00  08  16  24  55  40  48  56
   ```

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

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

Traffic class-to-DSCP mutation map configuration example

```
device# configure terminal
device(config)# qos map traffic-class-dscp tcdscp1
device(traffic-class-dscp-tcdscp1)# map traffic-class 4 to dscp 55
device(traffic-class-dscp-tcdscp1)# end
device# show qos maps traffic-class-dscp tcdscp1
device# copy running-config startup-config
```

Apply this map to a remarking profile only. For more information on the policer remarking profile, refer to the *Traffic Policing* chapter.

Configuring congestion control

Refer to the section Congestion control on page 46.

Configuring WRED

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

1. Enter configuration mode.

```
device# configure terminal
```

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

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

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

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

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

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

5. Return to privileged exec mode.

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

6. Verify the WRED configuration.

```
device# show qos red profiles 1

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


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

7. Save the configuration.

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

WRED configuration example

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

Displaying WRED statistics for an interface

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

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

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

Configuring link level flow control

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

Link level flow control can only be configured at the interface level.

1. Enter global configuration mode.

   ```
   device# configure terminal
   ```

2. Enter interface configuration mode.

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

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

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

4. Return to privileged exec mode.

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

5. Verify the configuration

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

6. Save the configuration

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

Link level flow control configuration example

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

Configuring scheduling

Configuring strict priority egress scheduling

Follow these steps to configure strict priority scheduling.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Select the policy map.

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

3. Select the classification.

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

4. Specify the scheduling attributes.

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

   For complete information for this command, refer to the *Brocade SLX-OS Layer 2 Configuration Guide* for the Brocade SLX 9140 and Brocade SLX 9240 Switches.

5. Return to privileged exec mode.

```
device(config-policymap)# 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. 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# copy running-config startup-config
```

Configuring flow-based QoS

Follow these tasks to configure ingress flow-based QoS.

Configuring a class map using an ACL

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

1. Enter global configuration mode.

```
device# configure terminal
```

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

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

   b) Allow traffic from a specific IP address.

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

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

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

   For details on creating access lists, refer to the *Brocade 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# do show running-config class-map class_1
class-map class_1
 match access-group ip_acl
```

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

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

Class map using an ACL configuration example

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

Configuring a policy map

Follow these steps to create a policy map.

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

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create and name a policy map.

```
device(config)# policy-map policyMap1
```

3. Return to privileged exec mode.

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

4. Verify the configuration

```
device# show policy-map

Number of policy maps : 2

Policy-Map policy
  Bound To:None

Policy-Map policyMap1
  Bound To:None
```

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

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

Policy map configuration example

```
device# configure terminal
device(config)# policy-map policyMap1
device(config-policymap)# end
device# show policy-map
device# copy running-config startup-config
```

Binding the policy map at the system level

Follow these steps to apply policing parameters to an interface.

1. Enter global configuration mode.

.

```
device# configure terminal
```

2. Bind the policy map to inbound traffic.

```
device(config)# qos service-policy in policyMap1
```

You cannot use a policy map that is bound to class maps, default, or CEE maps.

3. Return to privileged exec mode.

```
device(config-service-policy-in/policyMap1)# end
```

4. Verify the configuration.

```
device# show policy-map detail policyMap1

Policy-Map policyMap1
    Class class_1
      Police cir 40000 cbs 5000 eir 40000 ebs 3000

  Bound To: none
```

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

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

Bind a policy map globally configuration example

```
device# configure terminal
device(config)# qos service-policy in policyMap1
device(config-service-policy-in/policyMap1)# end
device# show policy-map detail policyMap1
device# copy running-config startup-config
```

Binding the policy map to an interface

Follow these step to configure the default remapping priorities.

Consider the following rules when binding a policy map to an interface:

- You can bind the same policy map to multiple interfaces but only one policy per interface per direction is allowed.
- You cannot bind policy maps to an interface if the policy map has no class map associations.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface configuration mode.

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

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

```
device(config-if-eth-0/4)# service-policy in policyMap1
```

4. Return to privileged exec mode.

```
device(config-if-eth-0/4)# end
```

5. Verify the configuration.

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

Ingress Direction :
 Policy-Map policyMap1
    Class class_1
      matches 0 packets
      Police cir 40000 cbs 5000 eir 40000 ebs 3000
        Stats:
          Operational cir:39856 cbs:5000 eir:39856 ebs:3000
          Conform Byte:0 Exceed Byte:0 Violate Byte:0
```

6. Verify the configuration.

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

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

Bind the policy map to an interface configuration example

```
device# configure terminal
device(config)# interface ethernet 0/4
device(conf-if-eth-0/4)# service-policy out policyMap1
device(conf-if-eth-0/4)# service-policy in policyMap1
device(conf-if-eth-0/4)# end
device# show policy-map interface ethernet 0/4
device# copy running-config startup-config
```

Configuring QoS mutation map actions

Follow these steps to configure a QoS mutation map.

A policy map and a class map have been configured.

Different kinds of mutations can be used depending on the command. For complete information, refer to relevant Command Reference guide. The available commands are **cos-mutation**, **cos-traffic-class**, **dscp-cos**, **dscp-mutation**, and **dscp-traffic-class**.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Select the policy map.

```
device(config)# policy-map policyMap1
```

3. Select the class.

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

4. Specify the mutation map.

```
device(config-policyclass)# map dscp-cos all-zero-map
```

In this example a DSCP-to-CoS mutation is configured.

> NOTE
> The dscp-cos map will be effective only if all three DSCP maps are applied.

5. Return to privileged exec mode.

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

6. Verify the configuration.

```
device# show run policy-map
policy-map policyMap1
 class default
  map dscp-cos all-zero-map
 !
```

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

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

QoS mutation map configuration example

```
device# configure terminal
device(config)# policy-map policyMap1
device(config-policymap)# class default
device(config-policyclass)# map dscp-cos all-zero-map
device(config-policyclass)# end
device# show run policy-map
device# copy running-config startup-config
```

Applying QoS mutation maps to an interface

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

A mutation map has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Select the policy map.

```
device(config)# policy-map policyMap1
```

3. Enter interface configuration mode.

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

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

```
device(conf-if-eth-0/1)# qos dscp-mutation dscpMutMap
```

> **NOTE**
> The dscp-mutation map will be effective only if all three DSCP maps are applied to the interface.

5. Return to privileged exec mode.

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

6. Verify the configuration.

```
device# show qos map dscp-mutation dscpMutMap

Dscp-to-Dscp Mutation map 'dscpMutMap' (dscp= d1d2)
d1 :  d2 0  1  2  3  4  5  6  7  8  9
---------------------------------------
 0 :    11 11 11 03 11 11 11 11 11 11
 1 :    11 11 11 11 11 11 11 11 11 11
 2 :    11 11 11 23 24 25 26 27 28 29
 3 :    30 31 32 33 34 35 36 37 38 39
 4 :    40 41 42 43 44 45 46 47 48 49
 5 :    50 51 52 53 54 11 11 11 11 11
 6 :    11 11 11 11

   Enabled on the following interfaces:
    Eth 0/1
```

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

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

Apply a QoS mutation map to an interface configuration example

```
device# configure terminal
device(config)# policy-map policyMap1
device(config-policymap)# interface ethernet 0/1
device(conf-if-eth-0/1)# qos dscp-mutation dscpMutMap
device(conf-if-eth-0/1)# end
device# show qos map dscp-mutation dscpMutMap
device# copy running-config startup-config
```

Configuring the QoS policing rate

To configure QoS for rate policing on an interface, you apply a policy map top the interface.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Create a policy map and enter policy map configuration mode.

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

3. Under policy map configuration mode, attach the classification map to the policy map.

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

4. Set the QoS action.

```
device(config-policymap-class)# police cir 40000
```

5. Return to privileged exec mode.

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

6. Verify the configuration.

```
device# show policy-map detail policy_1

Policy-Map policy_1
    Class default
      Police cir 400000 classification-type color remark-profile default

  Bound To:None
```

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

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

QoS policing rate configuration example

```
device# configure terminal
device(config)# policy-map policy_1
device(config-policymap)# class default
device(config-policymap-class)# police cir 40000
device(config-policymap-class)# end
device# show policy-map detail policy_1
device# copy running-config startup-config
```

Applying the QoS policing rate to an interface

Follow these steps to apply the policing rate to an interface.

A policy map has been configured.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Access interface configuration mode.

```
device(config-policymap-class)# interface ethernet 0/4
```

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

```
device(conf-if-eth-0/4)# service-policy in policy_1
```

4. Return to privileged exec mode.

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

5. Verify the configuration.

```
device# show policy-map

Number of policy maps : 2
...
Policy-Map policy_1
  Bound To: Et 0/4(in)
```

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

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

QoS policing rate on an interface configuration example

```
device# configure terminal
device(config-policymap-class)# interface ethernet 0/4
device(conf-if-eth-0/4)# service-policy in policy_1
device(conf-if-eth-0/4)# end
device# show policy-map
device# copy running-config startup-config
```

Configuring CoS trust

CoS trust can be configured on an interface to honor the incoming CoS value of the ingress packet.

You can enable CoS trust on an interface by using the **qos trust cos** command.

```
device(conf-if-eth-0/2)# qos trust cos
```

NOTE

When Layer 2 maps are active on an interface, CoS on this interface is trusted implicitly.

Configuring DSCP trust

Like CoS trust, DSCP trust can be configured on an interface to honor the incoming IP DSCP settings for deciding the CoS queue priority value of the ingress packet. When DSCP trust is not enabled, the DSCP value in the packet is ignored.

You can enable DSCP trust on an interface by using the **qos trust dscp** command.

```
device(conf-if-eth-0/2)# qos trust dscp
```

When DSCP trust is enabled, the following table shows the default DSCP to queue priority mapping.

TABLE 9 Default DSCP priority mappings

DSCP values	Traffic class (queue priority)
0-7	0
8-15	1
16-23	2
24-31	3
32-39	4
40-47	5
48-55	6
56-63	7

Converged Enhanced Ethernet (CEE) provisioning

Converged Enhanced Ethernet (CEE) provisioning allows you to configure the CEE parameters for Priority to Priority Group mapping, High (Strict) and Low (DWRR) priority groups, percentage of bandwidth for the low priority groups and PFC settings for each of the groups.

Under the CEE provisioning model all of these features are configured using two configuration tables, Priority Group Table and Priority Table.

The CEE Priority Group Table defines each Priority Group ID (PGID) and its scheduling policy (Strict Priority versus DWRR, DWRR weight, relative priority), and partially defines the congestion control (PFC) configuration. There are 16 rows in the CEE Priority Group Table. The following table is the default CEE Priority Group Table configuration for each PGID.

PGID	Bandwidth%	PFC
15.0	-	N
15.1	-	N
15.2	-	N
15.3	-	N
15.4	-	N
15.5	-	N
15.6	-	N
15.7	-	N
0	0	N
1	40	Y
2	60	N
3	0	N

PGID	Bandwidth%	PFC
4	0	N
5	0	N
6	0	N
7	0	N

Strict Priority versus DWRR is derived directly from the PGID value. All PGID with prefix 15 receive Strict Priority scheduling policy and all PGID in the range 0 through 7 receive DWRR scheduling policy. When any of the PGID 0 through 7 is activated, a bandwidth percentage must be specified. The bandwith is the minimum percentage of link bandwidth that the Priority Group should receive during periods of link oversubscription (after all Strict Priority Group have been serviced) and is used to derive DWRR weight. Relative priority between Priority Groups is the exact ordering of entries listed in the table, with PGID 15.0 being highest priority and PGID 15.7 being lowest priority.

Congestion control configuration is partially specified by enabling or disabling PFC. It is only a partial configuration of congestion control because the set of priorities mapped to the Priority Group is not known.

The CEE Priority Table defines each Priority (CoS) mapping to Priority Group, and completes the PFC configuration. There are eight rows in the CEE Priority Table. The following table presents the default CEE Priority Table configuration.

Priority (CoS) mapping	PGID
0	2
1	2
2	2
3	1
4	2
5	2
6	2
7	2

NOTE
Do not mix lossless traffic with lossy traffic.

Configuring a CEE provisioning map

Perform the following step to implement CEE provisioning.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Access the CEE map.

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

The only map name allowed is default.

3. Define the priority group table maps.

```
device(config-cee-map-default)# priority-group-table 0 weight 50 pfc on
device(config-cee-map-default)# priority-group-table 1 weight 50 pfc off
```

This step defines the maps for PGID 0 and 1.

4. Define a priority-table map.

```
device(config-cee-map-default)# priority-table 1 1 1 0 1 1 1 15.0
```

5. Set the lossless priority.

```
device(config-cee-map-default)# remap lossless-priority priority 2
```

   In this step, the CoS is remapped to 2.

6. Return to global configuration mode.

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

7. Specify an interface to apply the CEE provisioning map.

```
device(config)# interface Ethernet 0/6
```

8. Apply the map.

```
device(conf-if-eth-0/6)# cee default
```

   This command also activates and configures QoS priority flow control on the interface.

9. Return to privileged EXEC mode.

```
device(config-if-eth-0/6)# end
```

10. Verify the map.

```
device# show cee maps default
CEE Map 'default'
   Precedence: 1
   Remap Lossless-Priority to Priority 2
   Priority Group Table
     1:  Weight 50, PFC Enabled, BW% 40
     2:  Weight 50, 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:    1    1    1    0    1    1    1 15.0
   Enabled on the following interfaces: Eth 0/6
```

The following example is the steps in the previous configuration.

```
device# configure terminal
device(config)# cee-map default
device(config-cee-map-default)# priority-group-table 0 weight 50 pfc on
device(config-cee-map-default)# priority-group-table 1 weight 50 pfc off
device(config-cee-map-default)# priority-table 1 1 1 0 1 1 1 15.0
device(config-cee-map-default)# remap lossless-priority priority 2
device(config-cee-map)# exit
device(config)# interface Ethernet 0/6
device(conf-if-eth-0/6)# cee default
device(conf-if-eth-0/6)# qos flowcontrol pfc CoS tx on rx on
device(conf-if-eth-0/6)# end
device# show cee maps default
```

Dynamic buffer sharing

Buffer sharing among lossy unicast queues is supported. Buffer accounting is done in ingress at the port level and it is done in egress at the queue level.

The size of the shared buffer pool on the Brocade SLX 9140 is 14,615 pages and the Brocade SLX 9240 is 37,143 pages, where each page is a size of 256 bytes.

You can limit the buffer usage for egress lossy unicast queues, as shown in the following example.

```
device(config)# qos tx-queue limit 256
```

When setting the TX-queue limit, the queue limit is set to one of the following values (the next higher value above the configured value). For example, if the value is default of 512 Kbytes, the queue limit is set to 748288 bytes on the Brocade SLX 9140 or 570368 bytes on the Brocade SLX 9240.