Extreme ONE OS Switching v22.2.1.0 QoS Configuration Guide

Traffic Management

9039431-00 Rev AA
December 2025

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks® and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, see: https://www.extremenetworks.com/about-extreme-networks/company/legal/trademarks

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: https://www.extremenetworks.com/support/policies/open-source-declaration/

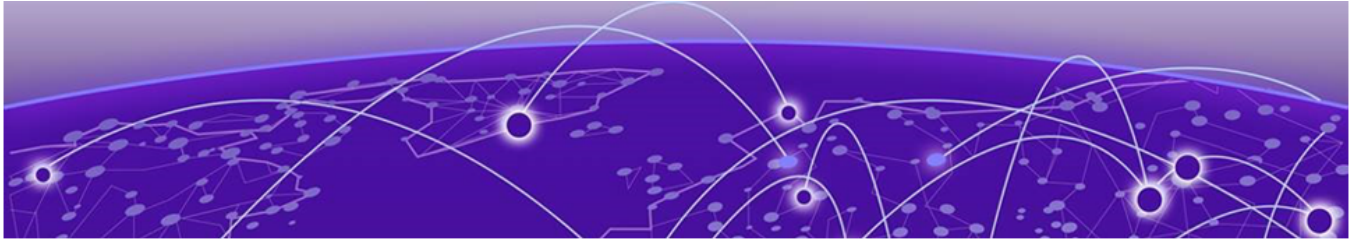
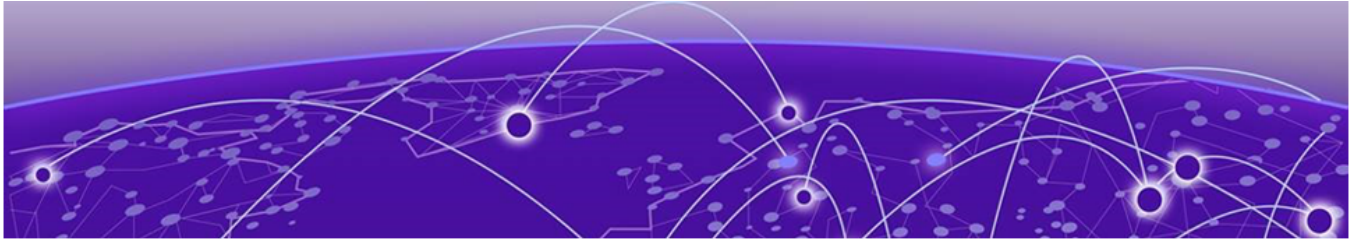


Table of Contents



Abstract

The *Extreme ONE OS Switching QoS Configuration Guide* version 22.2.1.0 is designed for IT professionals and network engineers and provides advanced procedures for deploying and managing Quality of Service (QoS) across complex, multiplatform networks. This guide includes procedures for fabric QoS for both access and fabric traffic to provide comprehensive traffic classification, marking, and prioritization capabilities for both traditional access traffic and VxLAN fabric traffic. This guide also includes procedures for egress port scheduling, which intelligently prioritizes and schedules packet transmission at egress ports, ensuring optimal network performance and traffic flow management.

Preface

Read the following topics to learn about:

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

Text Conventions

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

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

Table 1: Notes and warnings

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

Table 2: Text

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

Table 3: Command syntax

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

Documentation and Training

Find Extreme Networks product information at the following locations:

Current Product Documentation
Release Notes
Hardware and Software Compatibility for Extreme Networks products
Extreme Optics Compatibility
Other Resources such as articles, white papers, and case studies

Open Source Declarations

Some software files have been licensed under certain open source licenses. Information is available on the Open Source Declaration page.

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit the Extreme Networks Training page.

Help and Support

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

Extreme Portal

Search the GTAC (Global Technical Assistance Center) knowledge base; manage support cases and service contracts; download software; and obtain product licensing, training, and certifications.

The Hub

A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

Call GTAC

For immediate support: (800) 998 2408 (toll-free in U.S. and Canada) or 1 (408) 579 2800. For the support phone number in your country, visit www.extremenetworks.com/support/contact.

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

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

Subscribe to Product Announcements

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

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

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

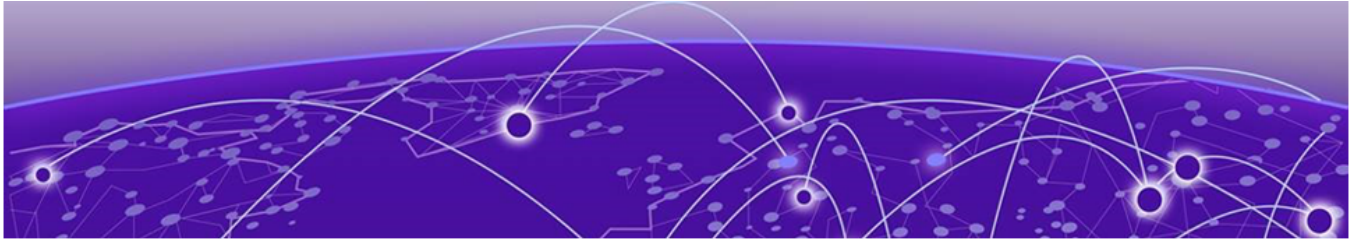
Send Feedback

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

- Content errors, or confusing or conflicting information.
- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, email us at Product-Documentation@extremenetworks.com.

Provide as much detail as possible including the publication title, topic heading, and page number (if applicable), along with your comments and suggestions for improvement.



About This Document

What's New in This Document

The *Extreme ONE OS Switching QoS Configuration Guide* is a new document for release 22.2.1.0.

For additional information, refer to the *Extreme ONE OS Switching 22.2.1.0 Release Notes*.

Supported Platforms

Extreme ONE OS Switching 22.2.1.0 supports Extreme 8520, Extreme 8720, Extreme 8730, and Extreme 8820 hardware platforms.

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

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

Fabric QoS

This chapter explains how to configure and manage Quality of Service (QoS) for both access and fabric traffic in your Extreme Networks environment. The Fabric QoS feature provides comprehensive traffic classification, marking, and prioritization capabilities for both traditional access traffic and VxLAN fabric traffic.

This feature supports QoS configuration for Multi-Chassis Link Aggregation Group (MLAG) tunnels using the same NVO-level configuration. MLAG tunnel QoS follows the same uniform and pipe mode behaviors as standard VxLAN tunnels.

Fabric QoS Overview

You can use this feature to:

- Control traffic prioritization based on packet content.
- Manage VxLAN tunnel QoS behavior.
- Ensure proper traffic class derivation and packet remarking.

Key Capabilities

The Fabric QoS implementation delivers three core functionalities.

Internal Traffic Class Derivation

The internal traffic class (TC) derivation feature:

- Automatically derives internal traffic classes from packet headers.
- Supports both Layer 2 (CoS/PCP) and Layer 3 (DSCP) classification.
- Provides flexible mapping options for different traffic types.

Egress Packet Remarking

The egress packet remarking feature:

- Modifies VLAN Priority Code Point (PCP) values on outgoing packets.
- Updates IP Differentiated Services Code Point (DSCP) values.
- Maintains QoS markings consistent with your network policies.

VxLAN Tunnel QoS

The VxLAN tunnel QoS feature:

- Controls QoS behavior during VxLAN encapsulation and decapsulation.
- Supports both uniform and pipe tunnel modes.
- Manages outer header DSCP marking for fabric traffic.

Access QoS and Fabric QoS Limitations

This feature has the following platform-related limitations, feature-related limitations, and scope.

Access QoS and Fabric QoS Platform Limitations

Access QoS (which is the foundation for fabric QoS) and fabric QoS have the following platform-related limitations.

Extreme 8520 and Extreme 8720

- Egress Layer 2 remarking is always enabled in hardware.
- DSCP remarking (in egress) in switched IP packets must be configured only at the interface level, not at the logical interface (subinterface) level.
- Trust DSCP without Trust CoS on a Layer 2 logical interface will use the hardware default map (traffic class = PCP) for Layer 2 tagged packets without IP headers.
- Tunnel decapsulation traffic class derivation is applied at the physical interface level, so any routed packets on this interface will have traffic classes derived using this map.

- After tunnel decapsulation, if a packet undergoes routing, then traffic class derivation is done from the ingress QoS map applied on the ingress Layer 3 interface on the tunnel terminating node.
- Untagged routed traffic with Trust DSCP on untagged logical interfaces without a user configured map will derive the traffic class as 1 instead of basing it on IP packet DSCP. The workaround enables Trust DSCP on a Layer 2 logical interface.

Extreme 8730

- For unknown-unicast and multicast traffic, classes are mapped as below (because the hardware is limited to only four multicast queues).

Traffic Class	Queue Number
0	0
1	0
2	1
3	1
4	2
5	2
6	3
7	3

- Updating the default-traffic-class value overwrites the CoS 0 (default/best-effort) to traffic-class mapping in hardware for default-mode bridge domain members.
- Egress Layer 2 remarking is always enabled in hardware.
- For IP packet forwarding in a bridge domain, the Layer 2 logical interface QoS configuration is not applied to packets. Instead, it uses the Virtual Ethernet (VE) interface QoS configuration (if the VE interface is present).

Extreme 8820-40C and Extreme 8820-80C

- Broadcast, unknown-unicast, and multicast (BUM) traffic shows per-queue statistics on source ports.
- Egress packet remarking is not supported because of hardware limitations.
- Shared user QoS maps affect all associated interfaces.
- User configuration of tunnel decapsulation mode is not supported.
- After tunnel decapsulation, if a packet undergoes routing, then the traffic class derivation is done from the ingress QoS map applied on the ingress Layer 3 interface on the tunnel terminating node.

Feature Limitations

Fabric QoS does not support the following features:

- User-configurable maps for fabric (VxLAN) QoS
- Policers, schedulers, shapers, and rate limiters

- Pause frames and flow control

Other Limitations

Access QoS and fabric QoS have the following additional limitations:

- (Non Extreme 8730 platforms only) Configuration for egress packet remarking is not supported for VLAN mode bridge domains. For logical interfaces (LIFs) associated to a VLAN mode bridge domain, a shared LIF is created internally. So specific LIF level behavior is shared across such LIFs on a port and cannot be controlled individually per LIF.
- To check the functionality of ingress maps on a VLAN mode bridge domain, you must configure QoS maps under the Ethernet or port channel interface (this will loop through all LIFs under this interface and apply the same QoS configuration as that of the Ethernet or port channel interface to the LIFs).
- Any QoS configuration on VLAN mode bridge domain LIFs is ignored and is replayed if the same LIF is moved to a default mode bridge domain.
- Trust and remark DSCP settings on a Layer 2 LIF is not effective until a user map is configured and attached to the Layer 2 LIF.
- If any specific QoS configuration exists under a specific LIF that belongs to an Ethernet or port channel interface, then the QoS map applied on the Ethernet or port channel interface is not applied to this LIF until the specific configuration is deleted from the LIF.
- When QoS configuration is deleted on a LIF, configurations on Ethernet or port channel interfaces are applied to the LIF.
- Extreme 8730, Extreme 8820-40C and Extreme 8820-80C platforms only) User configuration of tunnel decapsulation mode is not supported.

Scope

The fabric QoS implementation in this release focuses on per hop behavior (PHB) including:

- Packet classification and marking.
- Traffic class derivation.
- Basic remarking functionality.

Traffic Flow Processing

The system processes packets through these stages:

1. Ingress classification: Analyzes incoming packets and assigns internal traffic classes
2. Internal processing: Routes or switches packets while preserving QoS information

3. Egress processing: Applies remarking policies and queue assignments before transmission
4. Tunnel processing: Handles VxLAN encapsulation and decapsulation with appropriate QoS treatment

Prerequisites

Before configuring fabric QoS, make sure that you have:

- Administrative access to the switch management interface.
- An understanding of your network's QoS requirements.
- Existing VLAN and interface configurations (for access QoS).
- VxLAN tunnel configurations (for fabric QoS).

Planning Considerations

Before configuring fabric QoS, make sure that you:

- Identify traffic types requiring specific QoS treatment.
- Determine appropriate traffic class mappings for your environment.
- Plan DSCP and CoS marking strategies.
- Consider bandwidth and latency requirements for different traffic classes.

Best Practices

Configuration Management

- Document your QoS policies and mappings.
- Test configurations in a lab environment before deployment.
- Use consistent naming conventions for maps and policies.
- Regularly review and validate QoS behavior.

Performance Optimization

- Monitor traffic patterns and adjust classifications as needed.
- Consider platform specific queue limitations.
- Balance QoS granularity with operational simplicity.
- Plan for growth in traffic volumes and types.

Troubleshooting Preparation

- Establish baseline QoS statistics before making changes.
- Keep configuration backups for quick restoration.
- Document known platform limitations for your environment.
- Train operations staff on QoS verification procedures.

CLI Commands

To enable fabric QoS, you use the CLI to configure access QoS and then to configure fabric QoS. Access QoS and fabric QoS are different but related levels of QoS enforcement.

Access QoS is the foundation for fabric QoS. Access QoS sets the traffic classification and markings, and fabric QoS depends on those markings to enforce the correct forwarding behavior.

If traffic is not classified or marked properly at the access level, the fabric cannot give it the intended priority. Access QoS defines and enforces the contract at the edge, and fabric QoS honors that contract across the network.

Configuring Access QoS

Access QoS controls how the system classifies and marks traffic on traditional Layer 2 and Layer 3 interfaces.

Creating a Traffic Class Map

Traffic class maps define how incoming packet markings translate to internal traffic classes.

1. From privileged EXEC mode, access global configuration mode.
   ```
   device# configure terminal
   ```
2. Enter QoS configuration mode.
   ```
   device(config)# qos
   ```
3. Configure a QoS map for packet classification.
   ```
   device(config-qos)# traffic-class-map tcmap1
   ```
4. Define the CoS-to-traffic-class entries in the QoS traffic-class map.
   ```
   device(config-qos-traffic-class-map-tcmap1)# cos 2 traffic-class 6
   ```
5. Define the DSCP-to-traffic-class entries in the QoS traffic-class map.
   ```
   device(config-qos-traffic-class-map-tcmap1)# dscp 24 traffic-class 6
   ```
6. (Optional) Verify the traffic class map configuration.
   ```
   device# show running-config qos

   qos
     traffic-class-map tcmap1
       cos 2 traffic-class 6
   ```

```
    dscp 24 traffic-class 6
  !
!
device#
```

The following example creates a traffic class map named tcmap1:

```
device# configure terminal
device(config)# qos
device(config-qos)# traffic-class-map tcmap1
device(config-qos-traffic-class-map-tcmap1)# cos 2 traffic-class 6
device(config-qos-traffic-class-map-tcmap1)# dscp 24 traffic-class 6
device(config-qos-traffic-class-map-tcmap1)#
```

The following example shows how to display a QoS configuration. In this example, two traffic class maps (tcmap1 and tcl3map) are configured:

```
device# show running-config qos

qos
  traffic-class-map tcmap1
    cos 2 traffic-class 6
    dscp 24 traffic-class 6
  !
  traffic-class-map tcl3map
    dscp 10 traffic-class 7
  !
  remark-map rmrkmap1
    traffic-class 6 cos 2
    traffic-class 6 dscp 1
  !
  interface ethernet 0/2
    output
      remark cos
      remark-map rmrkmap1
    !
  !
  interface ethernet 0/3
    input
      trust dscp
      traffic-class-map tcl3map
    !
  !
  interface ethernet 0/1 subinterface vlan 100
    input
      trust cos
      traffic-class-map tcmap1
    !
  !
!
device#
```

Creating a Remark Map

Remark maps specify how internal traffic classes map to outgoing packet markings.

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

   ```
   device# configure terminal
   ```

2. Enter QoS configuration mode.

   ```
   device(config)# qos
   ```

3. Configure a map for remarking the QoS markings on packets.

```
device(config-qos)# remark-map rmrkmap1
```

4. Define the traffic-class-to-CoS entries for remarking.

```
device(config-qos-remark-map-rmrkmap1)# traffic-class 6 cos 2
```

5. Define the traffic-class-to-DSCP entries for remarking

```
device(config-qos-remark-map-rmrkmap1)# traffic-class 6 dscp 1
```

6. (Optional) Verify the remark map configuration.

```
device# show running-config qos

qos
  remark-map rmrkmap1
    traffic-class 6 cos 2
    traffic-class 6 dscp 1
  !
!
device#
```

The following example creates a remark map named rmrkmap1:

```
device# configure terminal
device(config)# qos
device(config-qos)# remark-map rmrkmap1
device(config-qos-remark-map-rmrkmap1)# traffic-class 6 cos 2
device(config-qos-remark-map-rmrkmap1)# traffic-class 6 dscp 1
device(config-qos-remark-map-rmrkmap1)#
```

The following example shows how to display a QoS configuration. In this example, a remark map named rmrkmap1 is configured:

```
device# show running-config qos

qos
  traffic-class-map tcmap1
    cos 2 traffic-class 6
    dscp 24 traffic-class 6
  !
  traffic-class-map tcl3map
    dscp 10 traffic-class 7
  !
  remark-map rmrkmap1
    traffic-class 6 cos 2
    traffic-class 6 dscp 1
  !
  interface ethernet 0/2
    output
      remark cos
      remark-map rmrkmap1
    !
  !
  interface ethernet 0/3
    input
      trust dscp
      traffic-class-map tcl3map
    !
  !
  interface ethernet 0/1 subinterface vlan 100
    input
      trust cos
      traffic-class-map tcmap1
    !
  !
```

```
!
device#
```

Applying a Traffic Class Map to an Interface

You can attach a QoS traffic class map to an interface under the input or output directions.

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

   ```
   device# configure terminal
   ```

2. Enter QoS configuration mode.

   ```
   device(config)# qos
   ```

3. Access the interface or subinterface to which you are assigning the traffic class map.

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

4. Enter the mode for configuring access QoS in the input direction or the output direction on an interface.

   ```
   device(config-qos-if-eth-0/3)# input
   ```

5. Enable the interface to trust the QoS CoS or QoS DSCP markings (using the **cos** or **dscp** keywords respectively) on ingress (incoming) packets for traffic class mapping. You can enable the interface to trust both marking types simultaneously.

   ```
   device(config-qos-if-eth-0/3-input)# trust dscp
   ```

6. Attach the traffic class map to the interface.

   ```
   device(config-qos-if-eth-0/3-input)# traffic-class-map tcl3map
   ```

7. (Optional) Verify the interface configuration.

   ```
   device# show running-config qos interface ethernet 0/3

   qos
     interface ethernet 0/3
       input
         trust dscp
         traffic-class-map tcl3map
       !
     !
   !
   device#
   ```

The following example attaches a traffic class map named tcl3map to Ethernet interface 0/3:

```
device# configure terminal
device(config)# qos
device(config-qos)# interface ethernet 0/3
device(config-qos-if-eth-0/3)# input
device(config-qos-if-eth-0/3-input)# trust dscp
device(config-qos-if-eth-0/3-input)# traffic-class-map tcl3map
device(config-qos-if-eth-0/3-input)#
```

The following example shows how to display a QoS configuration. In this example, a traffic class map named tcl3map is attached to Ethernet interface 0/3 in the input direction:

```
device# show running-config qos
```

```
qos
  traffic-class-map tcmap1
    cos 2 traffic-class 6
    dscp 24 traffic-class 6
  !
  traffic-class-map tcl3map
    dscp 10 traffic-class 7
  !
  remark-map rmrkmap1
    traffic-class 6 cos 2
    traffic-class 6 dscp 1
  !
  interface ethernet 0/2
    output
      remark cos
      remark-map rmrkmap1
    !
  !
  interface ethernet 0/3
    input
      trust dscp
      traffic-class-map tcl3map
    !
  !
  interface ethernet 0/1 subinterface vlan 100
    input
      trust cos
      traffic-class-map tcmap1
    !
  !
!
device#
```

Applying a Remark Map to an Interface

You can attach a QoS map to an interface for remarking the QoS markings on egress (outgoing) packets.

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

   ```
   device# configure terminal
   ```

2. Enter QoS configuration mode.

   ```
   device(config)# qos
   ```

3. Access the interface (or subinterface) to which you are assigning the remark map.

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

4. Enter the mode for configuring access QoS in the output direction or the input direction on an interface.

   ```
   device(config-qos-if-eth-0/2)# output
   ```

5. Enable the interface to remark the packet CoS or DSCP values (using the **cos** or **dscp** keyword respectively) on egress (outgoing) packets based on remark mapping.

   ```
   device(config-qos-if-eth-0/2-output)# remark cos
   ```

6. Attach the remark map to the interface.

   ```
   device(config-qos-if-eth-0/2-output)# remark-map rmrkmap1
   ```

7. (Optional) Verify the interface configuration.

   ```
   device# show running-config qos interface ethernet 0/2
   ```

```
qos
  interface ethernet 0/2
    output
      remark cos
      remark-map rmrkmap1
    !
  !
!
device#
```

The following example attaches a remark map named rmrkmap1 to Ethernet interface 0/2:

```
device# configure terminal
device(config)# qos
device(config-qos)# interface ethernet 0/2
device(config-qos-if-eth-0/2)# output
device(config-qos-if-eth-0/2-output)# remark cos
device(config-qos-if-eth-0/2-output)# remark-map rmrkmap1
device(config-qos-if-eth-0/2-output)#
```

The following example shows how to display a QoS configuration. In this example, a remark map named rmrkmap1 is attached to Ethernet interface 0/2 in the output direction:

```
device# show running-config qos

qos
  traffic-class-map tcmap1
    cos 2 traffic-class 6
    dscp 24 traffic-class 6
  !
  traffic-class-map tcl3map
    dscp 10 traffic-class 7
  !
  remark-map rmrkmap1
    traffic-class 6 cos 2
    traffic-class 6 dscp 1
  !
  interface ethernet 0/2
    output
      remark cos
      remark-map rmrkmap1
    !
  !
  interface ethernet 0/3
    input
      trust dscp
      traffic-class-map tcl3map
    !
  !
  interface ethernet 0/1 subinterface vlan 100
    input
      trust cos
      traffic-class-map tcmap1
    !
  !
!
device#
```

Configuring Fabric QoS

Fabric QoS manages QoS behavior for VxLAN tunnel traffic, controlling how the system handles encapsulation and decapsulation. Fabric QoS in Extreme ONE OS supports uniform tunnel mode and pipe tunnel mode.

Uniform Tunnel Mode

In uniform mode, the fabric honors and propagates host markings fabric-wide. Uniform mode provides the following functionality:

- Copies QoS markings between inner and outer headers
- Maintains end-to-end QoS visibility
- Modifies customer packet markings based on fabric treatment

Pipe Tunnel Mode

In pipe mode, the fabric uses its own QoS policies, independent of the original packet's markings. The fabric applies its own markings internally, and host markings pass through untouched. Pipe mode provides the following functionality:

- Isolates customer traffic QoS from fabric QoS
- Preserves original customer packet markings
- Uses fixed or mapped values for outer header markings

Configuring Tunnel QoS on an NVO Endpoint

You configure QoS behavior at the Network Virtualization Overlay (NVO) level, which applies to all tunnels under that NVO.

1. From privileged EXEC mode, access global configuration mode.
   ```
   device# configure terminal
   ```
2. Enter QoS configuration mode.
   ```
   device(config)# qos
   ```
3. Associate an NVO instance with the QoS node.
   ```
   device(config-qos)# nvo nvo1
   ```
4. Enter QoS NVO tunnel configuration mode.
   ```
   device(config-qos-nvo1)# tunnel-mode
   ```
5. Set the fabric QoS tunnel encapsulation mode for the NVO endpoint (using the **encapsulation pipe uniform** or **encapsulation pipe** [**dscp** [*dscp_value*]] commands).
   ```
   device(config-qos-nvo-nvo1-tunnel-mode)# encapsulation pipe dscp 44
   ```

6. Set the fabric QoS tunnel decapsulation mode for the NVO endpoint (using the **uniform** or **pipe** keywords).

```
device(config-qos-nvo-nvo1-tunnel-mode)# decapsulation uniform
```

> **Note**
> Tunnel decapsulation on the Extreme 8730 and Extreme 8820 platforms supports only the default mode (pipe) and is not configurable.

7. (Optional) Verify the fabric tunnel QoS configuration.

```
device# show qos nvo

nvo nvo1
  tunnel-mode:
    encapsulation pipe dscp 44
    decapsulation uniform
  !
!
device#
```

The following example configures fabric tunnel QoS encapsulation and decapsulation for an NVO named nvo1:

```
device# configure terminal
device(config)# qos
device(config-qos)# nvo nvo1
device(config-qos-nvo1)# tunnel-mode
device(config-qos-nvo-nvo1-tunnel-mode)# encapsulation pipe dscp 44
device(config-qos-nvo-nvo1-tunnel-mode)# decapsulation uniform
device(config-qos-nvo-nvo1-tunnel-mode)#
```

The following example shows how to display a QoS configuration. In this example, fabric tunnel QoS encapsulation and decapsulation for an NVO named nvo1 are configured:

```
device# show running-config qos

qos
  traffic-class-map tcmap1
    cos 2 traffic-class 6
    dscp 24 traffic-class 6
  !
  traffic-class-map tcl3map
    dscp 10 traffic-class 7
  !
  remark-map rmrkmap1
    traffic-class 6 cos 2
    traffic-class 6 dscp 1
  !
  interface ethernet 0/2
    output
      remark cos
      remark-map rmrkmap1
    !
  !
  interface ethernet 0/3
    input
      trust dscp
      traffic-class-map tcl3map
    !
  !
  interface ethernet 0/1 subinterface vlan 100
    input
      trust cos
```

```
      traffic-class-map tcmap1
    !
  !
  nvo nvo1
    tunnel-mode
      encapsulation pipe dscp 44
      decapsulation uniform
    !
  !
!
device#
```

Displaying Mappings in a Traffic Class Map

To display the CoS-to-traffic-class and DSCP-to-traffic-class mappings in a traffic class map, use the **show qos traffic-class-map** command.

The following example displays the CoS-to-traffic-class and DSCP-to-traffic-class mappings in a traffic class map named tcmap1 on the device:

```
device# show qos traffic-class-map tcmap1

  Default Traffic Class for map tcmap1: 1
  COS-to-Traffic-Class Map: tcmap1
    COS                 : 0  1  2  3  4  5  6  7
    ------------------------------------------
    Traffic-Class       : 1  0  6  3  4  5  6  7

  DSCP-to-Traffic-Class Map: tcmap1 (DSCP = d1d2)
    d1 : d2 0   1       2       3       4       5       6       7       8       9
    ---------------------------------------------------------------------------

    0 :     1   1       1       1       1       1       1       1       0       0
    1 :     0   0       0       0       0       0       2       2       2       2
    2 :     2   2       2       2       6       3       3       3       3       3
    3 :     3   3       4       4       4       4       4       4       4       4
    4 :     5   5       5       5       5       5       5       5       6       6
    5 :     6   6       6       6       6       6       7       7       7       7
    6 :     7   7       7       7
device#
```

The following example displays all CoS-to-traffic-class and DSCP-to-traffic-class mappings on the device:

```
device# sh qos traffic-class-map

  Default Traffic Class for map default: 1
  COS-to-Traffic-Class Map: default
    COS                 : 0  1  2  3  4  5  6  7
    ------------------------------------------
    Traffic-Class       : 1  0  2  3  4  5  6  7

  DSCP-to-Traffic-Class Map: default (DSCP = d1d2)
    d1 : d2 0   1       2       3       4       5       6       7       8       9
    ---------------------------------------------------------------------------

    0 :     1   1       1       1       1       1       1       1       0       0
    1 :     0   0       0       0       0       0       2       2       2       2
    2 :     2   2       2       2       3       3       3       3       3       3
    3 :     3   3       4       4       4       4       4       4       4       4
    4 :     5   5       5       5       5       5       5       5       6       6
```

```
    5 :     6   6       6       6       6       6       7       7       7       7
    6 :     7   7       7       7

  Default Traffic Class for map tcmap1: 1
  COS-to-Traffic-Class Map: tcmap1
    COS                 : 0  1  2  3  4  5  6  7
    ----------------------------------------
    Traffic-Class       : 1  0  6  3  4  5  6  7

  DSCP-to-Traffic-Class Map: tcmap1 (DSCP = d1d2)
    d1 : d2 0   1       2       3       4       5       6       7       8       9
    ---------------------------------------------------------------------------

    0 :     1   1       1       1       1       1       1       1       0       0
    1 :     0   0       0       0       0       0       2       2       2       2
    2 :     2   2       2       2       6       3       3       3       3       3
    3 :     3   3       4       4       4       4       4       4       4       4
    4 :     5   5       5       5       5       5       5       5       6       6
    5 :     6   6       6       6       6       6       7       7       7       7
    6 :     7   7       7       7

  Default Traffic Class for map tcl3map: 1
  COS-to-Traffic-Class Map: tcl3map
    COS                 : 0  1  2  3  4  5  6  7
    ----------------------------------------
    Traffic-Class       : 1  0  2  3  4  5  6  7

  DSCP-to-Traffic-Class Map: tcl3map (DSCP = d1d2)
    d1 : d2 0   1       2       3       4       5       6       7       8       9
    ---------------------------------------------------------------------------

    0 :     1   1       1       1       1       1       1       1       0       0
    1 :     7   0       0       0       0       0       2       2       2       2
    2 :     2   2       2       2       3       3       3       3       3       3
    3 :     3   3       4       4       4       4       4       4       4       4
    4 :     5   5       5       5       5       5       5       5       6       6
    5 :     6   6       6       6       6       6       7       7       7       7
    6 :     7   7       7       7
device#
```

Displaying Traffic Class Mappings for Remarking

To display the traffic-class-to-COS and traffic-class-to-DSCP mappings for remarking the QoS markings on packets, use the **`show qos remark-map`** command.

The following example displays the mappings within a QoS remark map named rmrkmap1 on the device:

```
device# show qos remark-map rmrkmap1

Traffic-Class-to-COS Map: rmrkmap1
    Traffic-Class       : 0  1  2  3  4  5  6  7
    ----------------------------------------
    COS                 : 1  0  2  3  4  5  2  7

  Traffic-Class-to-DSCP Map: rmrkmap1
    Traffic-Class       : 0     1       2       3       4       5       6       7
    ---------------------------------------------------------------------------
    DSCP                : 15    7       23      31      39      47      1       63
device#
```

The following example displays the mappings within all QoS remark maps on the device:

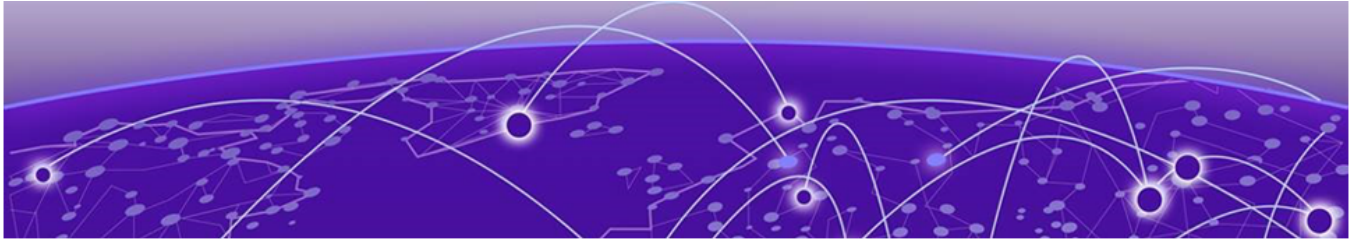
```
device# show qos remark-map

  Traffic-Class-to-COS Map: default
    Traffic-Class       : 0  1  2  3  4  5  6  7
    ------------------------------------------
    COS                 : 1  0  2  3  4  5  6  7

  Traffic-Class-to-DSCP Map: default
    Traffic-Class       : 0     1       2       3       4       5       6       7
    ------------------------------------------------------------------------------
    DSCP                : 15    7       23      31      39      47      55      63

  Traffic-Class-to-COS Map: rmrkmap1
    Traffic-Class       : 0  1  2  3  4  5  6  7
    ------------------------------------------
    COS                 : 1  0  2  3  4  5  2  7

  Traffic-Class-to-DSCP Map: rmrkmap1
    Traffic-Class       : 0     1       2       3       4       5       6       7
    ------------------------------------------------------------------------------
    DSCP                : 15    7       23      31      39      47      1       63
device#
```



Egress Port Scheduler

The following topics describe the egress port scheduling functionality in Extreme ONE OS Switching.

Egress Port Scheduler Overview

The Egress Port Scheduler provides essential Quality of Service (QoS) capabilities for your Extreme One OS network infrastructure by managing network traffic. By understanding the queue architecture, traffic mapping, and scheduling behaviors described in this guide, you can effectively manage network traffic priorities and ensure optimal performance for your applications.

This feature intelligently prioritizes and schedules packet transmission at egress ports, ensuring optimal network performance and traffic flow management. When packets arrive at your network switch, the system routes them to specific output ports and assigns them to priority queues. The Egress Port Scheduler determines which packets transmit first based on configured scheduling disciplines to provide precise control over traffic prioritization.

Key Benefits

The Egress Port Scheduler delivers several important advantages:

- Traffic Prioritization: You can ensure critical traffic receives higher priority over less important data flows.
- Bandwidth Management: The system efficiently allocates available bandwidth across different traffic classes.

- Performance Optimization: Your network maintains consistent performance even under heavy traffic loads.
- Hardware-Based Processing: The feature leverages dedicated hardware acceleration for maximum throughput

Queue Architecture

The system implements a hierarchical queue structure with the following characteristics.

Queue Distribution

Traffic is distributed among twelve total hardware queues that are split between unicast and multicast traffic types:

- Eight unicast queues that handle known unicast traffic (destinations with learned MAC addresses)
- Four multicast queues that process broadcast, unknown unicast, and multicast (BUM) traffic

Scheduling Modes

The scheduler operates using two distinct modes. These modes are deficit weighted round robin (DWRR) and strict priority (SP).

- Queues 0-5 use DWRR scheduling: This provides proportional bandwidth allocation based on configured weights. This ensures fair access to network resources.
- Queues 6-7 use strict priority scheduling: Higher priority traffic always transmits before lower priority traffic. This guarantees minimal latency for critical applications.

Default Traffic Class Mapping

The system maps traffic classes to queues according to the following configuration. :

Traffic Class	Unicast Queue	Multicast Queue	Scheduling Mode	Weight
0	0	0	DWRR	5
1	1	0	DWRR	20
2	2	1	DWRR	30
3	3	1	DWRR	40
4	4	2	DWRR	50
5	5	2	DWRR	50 (highest priority traffic)
6	6	3	SP	None
7	7	3	SP	None

Architecture and Operation

The system implements a two-level hierarchical scheduler. The scheduler processes traffic through a four-step process as described below.

Scheduler Hierarchy

The scheduler hierarchy consists of level 1 (L1) and level 0 (L0) nodes.

- For the L1 nodes, 12 L1 scheduler nodes are split into 8 unicast and 4 multicast queues. Each L1 node connects to specific traffic classes.
- For the L0 nodes, 8 L0 nodes apply weights and scheduling disciplines. These nodes aggregate traffic from L1 nodes before final port transmission.

Traffic Flow Process

The scheduler processes traffic through the following steps:

1. Classification: The system classifies incoming packets into traffic classes.
2. Queue Assignment: Packets route to appropriate unicast or multicast queues.
3. Scheduling Decision: The scheduler selects the next packet for transmission based on strict priority for queues 6–7 and deficit weighted round-robin for queues 0–5.
4. Transmission: Selected packets transmit through the egress port.

Egress Port Scheduler Limitations

Be aware of the following egress port scheduler limitations.

Unsupported Features

The following features are not supported:

- Dynamic Scheduler Profiles: You cannot configure custom scheduler profiles.
- Additional Platforms: Only the Extreme 8730 platform supports this feature. Other platforms (Extreme 8520, Extreme 8720, and Extreme 8820, do not support this feature in this release.
- Traffic Shaping: Egress queue shaping and port shaping are not available.

Hardware Constraints

Be aware of the following hardware constraints.

- Multicast Queue Sharing: Traffic classes 6-7 share multicast queues, which might affect strict priority behavior for multicast traffic.
- DWRR Multicast Behavior: Multicast traffic in DWRR queues might not follow configured weights exactly.

Monitoring Queue Statistics

You can monitor queue performance as well as QoS queue performance by examining packet statistics per interface and packet statistics per interface subject to the QoS policy respectively.

Monitoring Queue Performance

You can monitor queue performance by using the **show counters interface ethernet** { *interface-name* |**all** | **brief** } command. For details, see the *Extreme ONE OS Switching Command Reference*.

For each interface, this command displays the:

- Packet counts per queue.
- Byte counts per queue.
- Drop statistics (if applicable).

The following example displays statistics for Ethernet interface 0/80:

```
device# show counters interface ethernet 0/80

Interface Statistics: ethernet 0/80
 Carrier Transitions: 2
           LastClear: 0s
Input:
               Total pkts: 24832
           Broadcast pkts: 24101
             Discard pkts: 505
              Errors pkts: 503
               FCS Errors: 0
               MCast pkts: 413
                   Octets: 3104
               UCast pkts: 1922
                Runt pkts: 502
               CRC Errors: 0
Input Distribution:
             64 byte pkts: 24002
         65-127 byte pkts: 131
        128-255 byte pkts: 232
        256-511 byte pkts: 396
       512-1023 byte pkts: 29
      1024-1518 byte pkts: 42
               Jumbo pkts: 0
Out:
               Total pkts: 2746
           Broadcast pkts: 2002
             Discard pkts: 101
              Errors pkts: 21
               MCast pkts: 42
                   Octets: 343
               UCast pkts: 237
Rate Info:
                    Input: 301.000000 Mbits/sec, 25402 pkts/sec 91.00% of line-rate
                   Output: 322.000000 Mbits/sec, 26312 pkts/sec 93.00% of line-rate
device#
```

The following example displays statistics for all Ethernet interfaces on the device:

```
device# show counters interface ethernet all

Interface Statistics: ethernet 0/80
 Carrier Transitions: 2
           LastClear: 0s
Input:
              Total pkts: 24832
          Broadcast pkts: 24101
            Discard pkts: 505
             Errors pkts: 503
              FCS Errors: 0
              MCast pkts: 413
                  Octets: 3104
              UCast pkts: 1922
               Runt pkts: 502
              CRC Errors: 0
Input Distribution:
            64 byte pkts: 24002
        65-127 byte pkts: 131
       128-255 byte pkts: 232
       256-511 byte pkts: 396
      512-1023 byte pkts: 29
     1024-1518 byte pkts: 42
              Jumbo pkts: 0
Out:
              Total pkts: 2746
          Broadcast pkts: 2002
            Discard pkts: 101
             Errors pkts: 21
              MCast pkts: 42
                  Octets: 343
              UCast pkts: 237
Rate Info:
                   Input: 301.000000 Mbits/sec, 25402 pkts/sec 91.00% of line-rate
                  Output: 322.000000 Mbits/sec, 26312 pkts/sec 93.00% of line-rate

Interface Statistics: ethernet 0/90
 Carrier Transitions: 2
           LastClear: 0s
Input:
              Total pkts: 24822
          Broadcast pkts: 24099
            Discard pkts: 503
             Errors pkts: 501
              FCS Errors: 0
              MCast pkts: 411
                  Octets: 3104
              UCast pkts: 1920
               Runt pkts: 502
              CRC Errors: 0
Input Distribution:
            64 byte pkts: 23992
        65-127 byte pkts: 129
       128-255 byte pkts: 230
       256-511 byte pkts: 394
      512-1023 byte pkts: 27
     1024-1518 byte pkts: 40
              Jumbo pkts: 0
Out:
              Total pkts: 2736
          Broadcast pkts: 2000
            Discard pkts: 91
```

```
             Errors pkts: 19
              MCast pkts: 40
                  Octets: 343
              UCast pkts: 235
Rate Info:
                   Input: 319.000000 Mbits/sec, 25702 pkts/sec 92.00% of line-rate
                  Output: 344.000000 Mbits/sec, 26912 pkts/sec 94.00% of line-rate
device#
```

Monitoring QoS Queue Performance

You can validate QoS queue statistics by using the **show counters qos interface ethernet** { *interface-name* | **all** } **queue** { *queue-name* | **all** } command. For details, see the *Extreme ONE OS Switching Command Reference*.

For each interface, this command displays the number of:

- Received (Rx) packets per queue.
- Received (Rx) octets per queue.
- Packets that were dropped per queue according to the QoS policy.
- Octets (bytes) that were dropped per queue according to the QoS policy.

The following example displays statistics for Ethernet interface 0/1:1 QoS queue q0:

```
device# show counters qos interface ethernet 0/1:1 queue q0

-----------------------------------------------------------
interface-id := ethernet 0/1:1
-----------------------------------------------------------
Queue      Tx Packets Tx Octets  Drop Packets Drop Octets
========== ========== ========== ============ ===========
q0         1120       77818      0            0
device#
```

The following example displays statistics for Ethernet interface 0/1:1 and all QoS queues:

```
device# show counters qos interface ethernet 0/1:1 queue all

-----------------------------------------------------------------------
interface-id := ethernet 0/1:1
-----------------------------------------------------------------------
Queue    Rx Packets              Rx Octets               Drop Packets            Drop Octets
======== ====================== ====================== ======================
======================
q0       1120                   182560                 0                       0
q1       0                      0                      0                       0
q2       6                      564                    0                       0
q3       0                      0                      0                       0
q4       0                      0                      0                       0
q5       0                      0                      0                       0
q6       949                    77818                  0                       0
q7       0                      0                      0                       0
device#
```

Verification Methods

To verify proper scheduler operation:

1. Traffic Pattern Analysis: Monitor queue utilization during different traffic loads.
2. Performance Testing: Measure latency and throughput for different traffic classes.
3. Priority Verification: Confirm that strict priority queues receive preference during congestion.

Implementation Recommendations

Traffic Classification

- Map Critical Applications: Assign mission-critical traffic to strict priority queues (6-7).
- Balance DWRR Weights: Configure DWRR weights based on application requirements and expected traffic volumes.
- Monitor Utilization: Regularly check queue utilization to identify potential bottlenecks.

Network Design

- Plan for Growth: Consider future traffic growth when designing QoS policies.
- Test Thoroughly: Validate scheduler behavior under various traffic conditions.
- Document Configuration: Maintain clear documentation of traffic class assignments.