

# Junos® OS

---

## Hierarchical Class of Service User Guide

Published  
2024-12-17

Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, California 94089  
USA  
408-745-2000  
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

*Junos® OS Hierarchical Class of Service User Guide*  
Copyright © 2024 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

## YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

## END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

About This Guide | viii

1

## Hierarchical Class of Service

**Configuring Hierarchical Class of Service on MX Series 5G Universal Routing Platforms | 2**

Hierarchical Class of Service Overview | 2

Hierarchical Class of Service Network Scenarios | 6

Understanding Hierarchical Scheduling | 7

Priority Propagation in Hierarchical Scheduling | 11

Hierarchical CoS for Metro Ethernet Environments | 14

Hierarchical Schedulers and Traffic Control Profiles | 15

Example: Building a Four-Level Hierarchy of Schedulers | 18

Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels | 24

Example: Performing Output Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels | 26

Requirements | 26

Overview | 27

Configuration | 27

Verification | 41

Configuring Ingress Hierarchical CoS | 45

Hierarchical Class of Service for Network Slicing | 47

**Configuring Hierarchical Class of Service on MICs, MPCs, MLCs, and Aggregated Ethernet Interfaces | 52**

Understanding Hierarchical Scheduling for MIC and MPC Interfaces | 52

Configuring Ingress Hierarchical CoS on MIC and MPC Interfaces | 55

Per-Unit Scheduling and Hierarchical Scheduling for MPC Interfaces | 57

Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview | 61

Jitter Reduction in Hierarchical CoS Queues | 64

Example: Reducing Jitter in Hierarchical CoS Queues | 67

Requirements | 68

Overview | 68

Configuration | 68

Verification | 74

Hierarchical Schedulers on Aggregated Ethernet Interfaces Overview | 76

Configuring Hierarchical Schedulers on Aggregated Ethernet Interfaces | 77

Example: Configuring Scheduling Modes on Aggregated Interfaces | 78

Increasing Available Bandwidth on Rich-Queuing MPCs by Bypassing the Queuing Chip | 84

2

## Hierarchical CoS for Subscriber Management

**Hierarchical Class of Service for Subscriber Management | 88**

Hierarchical Class of Service for Subscriber Management Overview | 88

Understanding Hierarchical CoS for Subscriber Interfaces | 90

Hardware Requirements for Dynamic Hierarchical CoS | 98

Configuring Static Hierarchical Scheduling in a Dynamic Profile | 100

Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links | 101

Configuring Hierarchical CoS on a Static PPPoE Subscriber Interface | 103

Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access | 104

Requirements | 104

Overview | 105

Configuration | 106

Verification | 119

**Applying CoS to Groups of Subscriber Interfaces | 121**

CoS for Interface Sets of Subscribers Overview | 121

Configuring an Interface Set of Subscribers in a Dynamic Profile | 124

Example: Configuring a Dynamic Interface Set of VLAN Subscribers | 125

Requirements | 125

- Overview | **125**
- Configuring the Dynamic VLANs | **126**
- Configuring Dynamic Traffic Scheduling and Shaping | **129**
- Configuring the Interface Set in the Dynamic Profile | **134**
- Configuring DHCP Access | **137**
- Configuring RADIUS Authentication | **139**
- Verification | **145**

Example: Configuring a Dynamic Service VLAN Interface Set of Subscribers in a Dynamic Profile | **146**

- Requirements | **147**
- Overview | **147**
- Configuration | **148**
- Verification | **151**

## **Configuring Hierarchical Scheduling for MPLS Pseudowire Interfaces | 153**

Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | **153**

CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | **154**

CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | **156**

Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces | **158**

CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | **160**

Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces  
(Logical Interfaces over a Transport Logical Interface) | **165**

Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces  
(Logical Interfaces over a Pseudowire Interface Set) | **167**

## **Configuring Hierarchical Scheduling for L2TP | 171**

CoS for L2TP LAC Subscriber Interfaces Overview | **171**

Configuring Dynamic CoS for an L2TP LAC Tunnel | **174**

CoS for L2TP LNS Inline Services Overview | **176**

Configuring an Inline Service Interface for L2TP LNS | **177**

Configuring Dynamic CoS for an L2TP LNS Inline Service | **178**

## **Preventing Bandwidth Contention on Subscriber Interfaces | 182**

Hierarchical CoS Shaping-Rate Adjustments Overview | **182**

Shaping Rate Adjustments for Subscriber Local Loops Overview	185
Guidelines for Configuring Shaping-Rate Adjustments for Subscriber Local Loops	186
Configuring the Minimum Adjusted Shaping Rate on Scheduler Nodes for Subscribers	187
Overview	187
Configuring a Static Minimum Adjusted Shaping Rate on Scheduler Nodes	188
Configuring a Dynamic Minimum Adjusted Shaping Rate on Scheduler Nodes	188
Configuring Shaping-Rate Adjustments on Queues	188
Overview	189
Configuring a Static Shaping-Rate Adjustment for Queues	189
Configuring a Dynamic Shaping-Rate Adjustment for Queues	190
Enabling Shaping-Rate Adjustments for Subscriber Local Loops	191
Configuring Static Logical Interface Sets to Serve as CoS Hierarchical Scheduler Nodes for Subscriber Loops	191
Configuring the Logical Interfaces That Compose the Static Logical Interface Sets	192
Configuring Hierarchical CoS on the Static Logical Interface Sets That Serve as Hierarchical Scheduler Nodes for Subscriber Local Loops	193
Configuring ANCP Functionality That Supports and Drives Shaping-Rate Adjustments for Subscriber Local Loops	195
Disabling Shaping-Rate Adjustments for Subscriber Local Loops	197
Disabling Hierarchical Bandwidth Adjustment for Subscriber Interfaces with Reverse-OIF Mapping	198
Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops	198
Verifying the Configuration of Shaping-Rate Adjustments for Subscriber Local Loops	202
Verifying the Configuration of ANCP for Shaping-Rate Adjustments	203
Using Hierarchical CoS to Adjust Shaping Rates Based on Multicast Traffic	204
<b>Configuring Targeted Distribution of Subscribers on Aggregated Ethernet Interfaces  </b>	<b>208</b>
Distribution of Demux Subscribers in an Aggregated Ethernet Interface	208
Providing Accurate Scheduling for a Demux Subscriber Interface of Aggregated Ethernet Links	212
Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces	213
Configuring Link and Module Redundancy for Demux Subscribers in an Aggregated Ethernet Interface	214

Configuring Rebalancing of Demux Subscribers in an Aggregated Ethernet Interface | 215

Configuring Periodic Rebalancing of Subscribers in an Aggregated Ethernet Interface | 215

Configuring Manual Rebalancing of Subscribers on an Aggregated Ethernet Interface | 216

Example: Separating Targeted Multicast Traffic for Demux Subscribers on Aggregated Ethernet Interfaces | 216

Requirements | 217

Overview | 217

Configuration | 218

Verification | 226

Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 231

Configuring the Distribution Type for PPPoE Subscribers on Aggregated Ethernet Interfaces | 232

Verifying the Distribution of PPPoE Subscribers in an Aggregated Ethernet Interface | 233

**Applying CoS Using Parameters Received from RADIUS | 235**

Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS | 235

Changing CoS Services Overview | 240

CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview | 244

Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions | 246

Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | 247

Configuring Static Default Values for Traffic Scheduling and Shaping | 248

Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions | 250

CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets | 253

Example: Configuring Dynamic Hierarchical Scheduling for Subscribers | 259

## **Configuration Statements and Operational Commands**

Junos CLI Reference Overview | 270

# About This Guide

Use this guide to understand and configure hierarchical class of service (CoS) features in Junos OS to define service levels that provide different delay, jitter, and packet loss characteristics to particular applications served by specific traffic flows. Applying CoS features to each device in your network ensures quality of service (QoS) for traffic throughout your entire network.

## RELATED DOCUMENTATION

[The Day One: Deploying Basic QoS](#)

[Day One: Dynamic Subscriber Management](#)



# 1

PART

## Hierarchical Class of Service

---

Configuring Hierarchical Class of Service on MX Series 5G Universal Routing  
Platforms | 2

Configuring Hierarchical Class of Service on MICs, MPCs, MLCs, and Aggregated  
Ethernet Interfaces | 52

---

# Configuring Hierarchical Class of Service on MX Series 5G Universal Routing Platforms

## IN THIS CHAPTER

- Hierarchical Class of Service Overview | 2
- Hierarchical Class of Service Network Scenarios | 6
- Understanding Hierarchical Scheduling | 7
- Priority Propagation in Hierarchical Scheduling | 11
- Hierarchical CoS for Metro Ethernet Environments | 14
- Hierarchical Schedulers and Traffic Control Profiles | 15
- Example: Building a Four-Level Hierarchy of Schedulers | 18
- Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels | 24
- Example: Performing Output Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels | 26
- Configuring Ingress Hierarchical CoS | 45
- Hierarchical Class of Service for Network Slicing | 47

## Hierarchical Class of Service Overview

Hierarchical class of service (HCoS) is the ability to apply traffic schedulers and shapers to a hierarchy of *scheduler nodes*. Each level of the scheduler hierarchy can be used to shape traffic based on different criteria such as application, user, VLAN, slice, and physical port.

This allows you to support the requirements of different services, applications, and users on the same physical device and physical infrastructure.

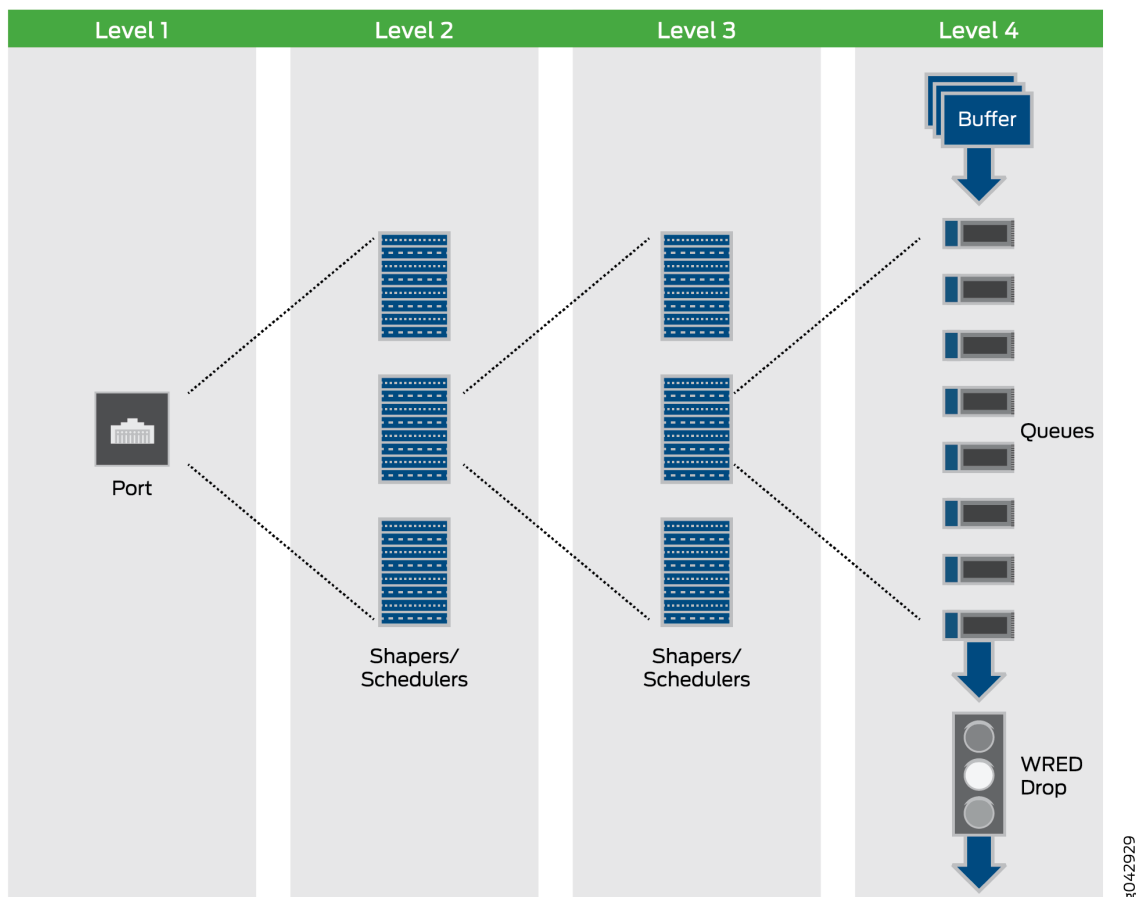
HCoS is implemented primarily using traffic classifiers at the ingress and hierarchical schedulers and shapers at the egress.

A classifier is a filter that labels traffic at the device ingress based on configurable parameters such as application or destination. Traffic is classified into what is called a forwarding equivalence class (FEC). The FEC defines a class of traffic that receives common treatment.

Schedulers, and their associated shapers, are the functions that control the traffic bandwidth, *jitter* (delay variation), and packet loss priority at the egress of the device.

Hierarchical schedulers are used to apply multiple levels of scheduling and shaping with each level applied to different classifications such as forwarding equivalence class, VLAN, and physical interface (port) as shown in [Figure 1 on page 3](#).

**Figure 1: Hierarchical Scheduling Architecture**



**NOTE:** Hierarchical class of service is also referred to as Hierarchical Quality of Service (HQoS) in other vendor's documentation.

A typical application of HCoS is to configure multiple levels of egress schedulers and shapers, at the subscriber edge, using dynamic profiles to provide traffic shaping and prioritization at the subscriber VLAN level and for multiple classes of traffic.

Dynamic profiles are a mechanism that allows you to dynamically apply schedulers and shapers to individual subscribers or groups of subscribers.

To learn more about HCoS, the following topics are very helpful:

- *Junos CoS on MX Series 5G Universal Routing Platforms Overview*
- *CoS Features and Limitations on MX Series Routers*
- *CoS Features of the Router Hardware, PIC, MIC, and MPC Interface Families*
- *How Schedulers Define Output Queue Properties*
- *Subscriber Access Network Overview*
- *CoS for Subscriber Access Overview*
- "[Hierarchical Class of Service for Subscriber Management Overview](#)" on page 88

The Junos OS hierarchical schedulers support up to five levels of scheduler hierarchies on MX Series devices when using enhanced queuing Dense Port Concentrators (DPCs) or fine-grained queuing Modular Port Concentrators (MPCs), and Modular Interface Cards (MICs). It is important to know the capabilities of your hardware with respect to HCoS. The following are a few tips to help you:

- Only certain hardware supports the five-level scheduler hierarchy of HCoS.
- The number of queues and logical interfaces supported is dependent upon exactly what hardware you are using.
- The MX Series Packet Forwarding Engine handles guaranteed bandwidth and scheduler node weight differently than other Packet Forwarding Engines.
- The fine-grained queuing MPCs and MICs have a certain granularity with respect to the shaping and delay buffer values. The values used are not necessarily exactly the values configured.

To learn more about platform support for HCoS, use the Juniper Networks Feature Explorer (<https://pathfinder.juniper.net/feature-explorer/>). In the Feature Explorer, search on *hierarchical schedulers*.

In addition, it is important to note the following:

- HCoS is most frequently used to enforce service level agreements at the subscriber edged using dynamic traffic control profiles.
- Hierarchical schedulers can also be applied to Ethernet pseudowire interfaces, aggregated Ethernet interfaces, Layer 2 Tunnel Protocol (L2TP) network server (LNS) inline services, and GRE tunnels.

- Hierarchical ingress policing is a feature that is complimentary to and often used in conjunction with HCoS.
- There are other features in Junos OS that have similar sounding names.



**NOTE:** The *hierarchical scheduler and shaper* feature supported on the SRX Series Firewalls is not the HCoS feature described here.

Before planning HCoS for you network, you should learn about HCoS, define you needs, plan how you want to implement HCoS, and test the operation in a simulated environment.

**Table 1: Resources for Learning More About HCoS**

Document	Description
<a href="#">Day One: Deploying Basic QoS</a> Juniper Networks Books	This book is a good resource for learning the basics of CoS on Juniper Networks devices.
Juniper MX-Series O'Reilly Media	Learn about the advanced features of HCoS. This book provides an in-depth description of how HCoS works and how it can be deployed. It also provides a lab tested topology and configuration example.
<a href="#">Day One: Dynamic Subscriber Management</a> Juniper Networks Books	Learn how to use HCoS in conjunction with dynamic traffic control profiles for subscriber management. This book also includes troubleshooting.
QoS Enabled Networks John Wiley & Sons	This book is an additional source for studying QoS.

Documentation related to HCoS is consolidated in the [Hierarchical Class of Service User Guide](#).

## RELATED DOCUMENTATION

[Hierarchical Class of Service for Subscriber Management Overview](#) | 88

*Hierarchical Class of Service Network Scenarios*

*Understanding Hierarchical Scheduling*

## Hierarchical Class of Service Network Scenarios

### IN THIS SECTION

- Services to Subscribers | 6
- Services to Businesses | 7
- Wireless Backhaul | 7

Hierarchical class of service (HCoS) can be used to provide granular control of traffic for a variety of different applications.



**NOTE:** Hierarchical class of service is also referred to as Hierarchical Quality of Service (HQoS) in other vendor's documentation.

Hierarchical class of service is most frequently used in the following scenarios:

### Services to Subscribers

Multiservice network operators face a challenge to provide different types of services on the same infrastructure to residential and business subscribers. The network operator needs to make sure each subscriber gets the network resources they paid for and each service gets the network resources it needs to operate properly.

If no CoS is applied, one service could consume most of the bandwidth of the transmission infrastructure and starve the other services.

Using hierarchical class of service, the network edge device can have up to five levels of scheduling and prioritization. So the traffic can be shaped and prioritized per customer and per service type. Controlling traffic in this way provides the ability to deliver the required service level for each subscriber for each service type.

By allowing network operators to consolidate different services and multiple customers on the same physical infrastructure, hierarchical class of service helps maximize the ability to offer revenue generating services while simultaneously minimizing capital cost.

## Services to Businesses

Hierarchical class of service is a valuable tool for service providers that support business customers who are running applications with different prioritization and scheduling requirements over the same infrastructure. In this scenario hierarchical class of service allows lower priority traffic to fully utilize the available bandwidth on a port, while simultaneously ensuring low latency and guaranteed bandwidth to higher priority traffic on the same port.

This allows a provider to consolidate different services on the same physical device and physical infrastructure thus optimizing network resources while maintaining the required level of service.

All of this maximizes revenue and minimizes cost

## Wireless Backhaul

In a cellular network the operator might want to offer business services along with its cell tower traffic. One of the main challenges is to make sure that the time-sensitive cell traffic is not affected by the business services running on the same infrastructure. Each type of traffic has its own priority flows and bandwidth constraints. For example, wireless backhaul is very sensitive to fluctuations in the packet stream (Jitter) because it relies on synchronization.

In this scenario, hierarchical class of service allows each type of traffic to receive the required resources and quality of service while being delivered over the same infrastructure.

By consolidate different services on the same physical infrastructure, HCoS helps maximize revenue and minimize cost.

## RELATED DOCUMENTATION

*Hierarchical Class of Service Overview*

[Hierarchical Class of Service for Subscriber Management Overview | 88](#)

## Understanding Hierarchical Scheduling

### IN THIS SECTION

- [Hierarchical Scheduling Terminology | 8](#)
- [Scheduler Node-Level Designations in Hierarchical Scheduling | 9](#)

Hierarchical class of service (HCoS) is a set of capabilities that enable you to apply unique CoS treatment for network traffic based on criteria such as user, application, VLAN, and physical port.

This allows you to support the requirements of different services, applications, and users on the same physical device and physical infrastructure.

This topic covers the following information:

## Hierarchical Scheduling Terminology

Hierarchical scheduling introduces some new CoS terms and also uses some familiar terms in different contexts:

- **Customer VLAN (C-VLAN)**—A C-VLAN, defined by IEEE 802.1ad. A stacked VLAN contains an outer tag corresponding to the S-VLAN, and an inner tag corresponding to the C-VLAN. A C-VLAN often corresponds to CPE. Scheduling and shaping is often used on a C-VLAN to establish minimum and maximum bandwidth limits for a customer. See also *S-VLAN*.
- **Interface set**—A logical group of interfaces that describe the characteristics of set of service VLANs, logical interfaces, customer VLANs, or aggregated Ethernet interfaces. Interface sets establish the set and name the traffic control profiles. See also *Service VLAN*.
- **Scheduler**— A scheduler defines the scheduling and queuing characteristics of a queue. Transmit rate, scheduler priority, and buffer size can be specified. In addition, a drop profile may be referenced to describe WRED congestion control aspects of the queue. See also *Scheduler map*.
- **Scheduler map**—A scheduler map is referenced by traffic control profiles to define queues. The scheduler map establishes the queues that comprise a scheduler node and associates a forwarding class with a scheduler. See also *Scheduler*.
- **Stacked VLAN**—An encapsulation on an S-VLAN with an outer tag corresponding to the S-VLAN, and an inner tag corresponding to the C-VLAN. See also *Service VLAN* and *Customer VLAN*.
- **Service VLAN (S-VLAN)**—An S-VLAN, defined by IEEE 802.1ad, often corresponds to a network aggregation device such as a DSLAM. Scheduling and shaping is often established for an S-VLAN to provide CoS for downstream devices with little buffering and simple schedulers. See also *Customer VLAN*.
- **Traffic control profile**—Defines the characteristics of a scheduler node. Traffic control profiles are used at several levels of the CLI, including the physical interface, interface set, and *logical interface*



levels. Scheduling and queuing characteristics can be defined for the scheduler node using the `shaping-rate`, `guaranteed-rate`, and `delay-buffer-rate` statements. Queues over these scheduler nodes are defined by referencing a scheduler map. See also *Scheduler* and *Scheduler map*.

- VLAN—Virtual LAN, defined on an Ethernet logical interface.

## Scheduler Node-Level Designations in Hierarchical Scheduling

Scheduler hierarchies are composed of nodes and queues. Queues terminate the hierarchy. Nodes can be either root nodes, leaf nodes, or internal (non-leaf) nodes. Internal nodes are nodes that have other nodes as “children” in the hierarchy.

Scheduler hierarchies consist of levels, starting with Level 1 at the physical port. This chapter establishes a four-level scheduler hierarchy which, when fully configured, consists of the physical interface (Level 1), the interface set (Level 2), one or more logical interfaces (Level 3), and one or more queues (Level 4).



**NOTE:** Beginning with Junos OS Release 16.1, certain MPCs on MX Series devices support up to five levels of scheduler hierarchies. The concepts presented in this topic apply similarly to five scheduler hierarchy levels.

[Table 2 on page 9](#) describes the possible combinations of scheduler nodes and their corresponding node level designations for a hierarchical queuing MIC or MPC.

**Table 2: Node Levels Designations in Hierarchical Scheduling**

Scheduler Configuration for Hierarchical CoS	Hierarchical CoS Scheduler Nodes			
	Root Node	Internal (Non-Leaf) Nodes		Leaf Node
	Level 1	Level 2	Level 3	Level 4
One or more traffic control profiles configured on logical interfaces, but no interface-sets configured	Physical interface	—	One or more logical interfaces	One or more queues
Interface-sets (collections of logical interfaces) configured, but no traffic-control profiles configured on logical interfaces	Physical interface	—	Interface-set	One or more queues

**Table 2: Node Levels Designations in Hierarchical Scheduling (Continued)**

Scheduler Configuration for Hierarchical CoS	Hierarchical CoS Scheduler Nodes			
	Root Node	Internal (Non-Leaf) Nodes		Leaf Node
	Level 1	Level 2	Level 3	Level 4
Fully configured scheduler nodes	Physical interface	Interface-set	One or more logical interfaces	One or more queues

The table illustrates how the configuration of an interface set or logical interface affects the terminology of hierarchical scheduler nodes. For example, suppose you configure an interface-set statement with logical interfaces (such as `unit 0` and `unit 2`) and a queue. In this case, the interface-set is an internal node at Level 2 of the scheduler node hierarchy. However, if there are no traffic control profiles attached to logical interfaces, then the interface set is at Level 3 of the hierarchy.

### Hierarchical Scheduling at Non-Leaf Nodes

Whereas standard CoS scheduling is based on the scheduling and queuing characteristics of a router's egress ports and their queues, hierarchical CoS scheduling is based on the scheduling and queuing characteristics that span a hierarchy of *scheduler nodes* over a port. The hierarchy begins at Level 1, a *root node* at the physical interface (port) level of the CLI hierarchy and terminates at Level 4, a *leaf node* at the queue level. Between the root and leaf nodes of any scheduler hierarchy are one or more *internal nodes*, which are non-root nodes that have other nodes as "children" in the hierarchy.

Whereas you configure standard CoS scheduling by applying a *scheduler map* to each egress port to specify a forwarding class and a queue priority level, you configure hierarchical CoS scheduling with additional parameters. To configure hierarchical CoS scheduling, you apply a scheduler map to the queue level (Level 4) of a scheduler hierarchy, and you can apply a different *traffic control profile* at each of the other levels. A traffic control profile specifies not only a scheduler map (forwarding class and queue priority level) but also optional shaping rate (PIR), guaranteed transmit rate (CIR), burst rate, delay buffer rate, and drop profile.

### Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
16.1	Beginning with Junos OS Release 16.1, certain MPCs on MX Series devices support up to five levels of scheduler hierarchies.

## Priority Propagation in Hierarchical Scheduling

Priority propagation is useful for mixed traffic environments when, for example, you want to make sure that the voice traffic of one customer does not suffer due to the data traffic of another customer. Nodes and queues are serviced in the order of their priority. The default priority of a queue is low, and you can explicitly configure a queue priority by including the `priority` statement at the `[edit class-of-service schedulers scheduler-name]` hierarchy level.

You cannot directly configure the priorities of all hierarchical scheduling elements. The priorities of internal nodes, for example, are determined as follows:

- The highest priority of an active child, that is, a child currently containing traffic. (Interface sets only take the highest priority of their active children.)
- Whether the node is above its configured guaranteed rate (CIR) or not (this is only relevant if the physical interface is in CIR mode).

Each queue has a configured priority and a hardware priority. The usual mapping between the configured priority and the hardware priority is shown in [Table 3 on page 11](#).

**Table 3: Queue Priority**

Configured Priority	Hardware Priority
Strict-high	0
High	0
Medium-high	1
Medium-low	1

**Table 3: Queue Priority (Continued)**

Configured Priority	Hardware Priority
Low	2

MPCs also have configurable CLI priorities of excess-priority high, excess-priority medium-high, excess-priority medium-low, and excess-priority low. These priorities only take effect above the guaranteed rate.

In CIR mode, the priority for each internal node depends on whether the highest active child node is above or below the guaranteed rate. The mapping between the highest active child's priority and the hardware priority below and above the guaranteed rate is shown in [Table 4 on page 12](#).

**Table 4: Internal Node Queue Priority for CIR Mode**

Configured Priority of Highest Active Child Node	Hardware Priority Below Guaranteed Rate	Hardware Priority Above Guaranteed Rate
Strict-high	0	0
High	0	3
Medium-high	1	3
Medium-low	1	3
Low	2	3
Excess-priority high*	N/A	3
Excess-priority medium-high*	N/A	3
Excess-priority medium-low*	N/A	4
Excess-priority low*	N/A	4

**Table 4: Internal Node Queue Priority for CIR Mode (Continued)**

Configured Priority of Highest Active Child Node	Hardware Priority Below Guaranteed Rate	Hardware Priority Above Guaranteed Rate
--	---	---

\* MPCs only

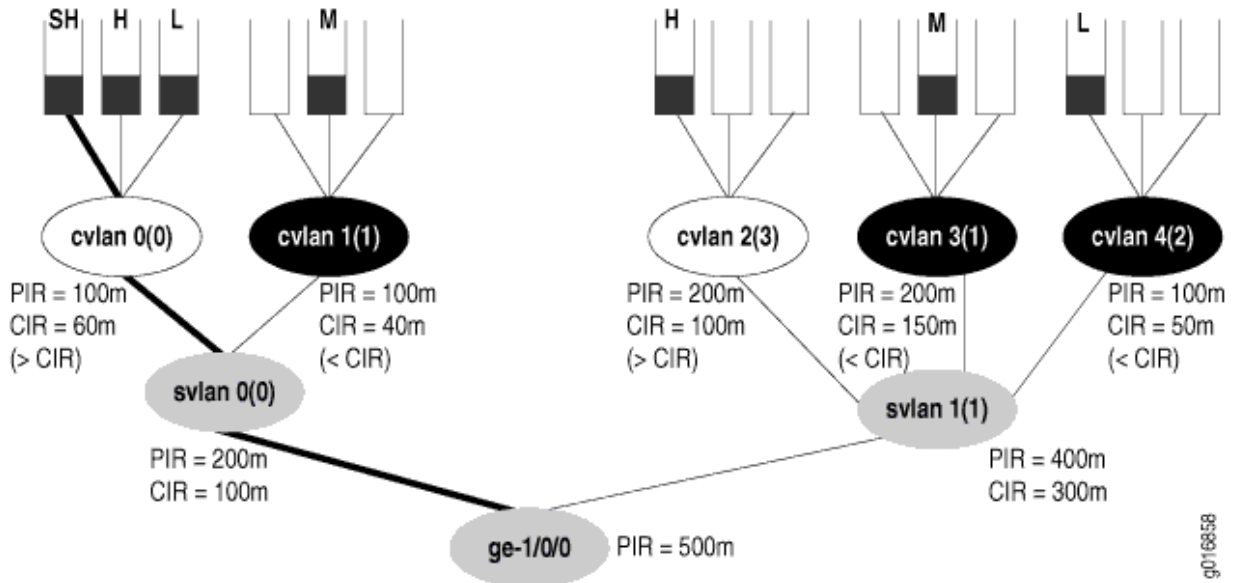
In PIR-only mode, nodes cannot send if they are above the configured shaping rate. The mapping between the configured priority and the hardware priority is for PIR-only mode is shown in [Table 5 on page 13](#).

**Table 5: Internal Node Queue Priority for PIR-Only Mode**

Configured Priority	Hardware Priority
Strict-high	0
High	0
Medium-high	1
Medium-low	1
Low	2

A physical interface with hierarchical schedulers configured is shown in [Figure 2 on page 14](#). The configured priorities are shown for each queue at the top of the figure. The hardware priorities for each node are shown in parentheses. Each node also shows any configured shaping rate (PIR) or guaranteed rate (CIR) and whether or not the queues is above or below the CIR. The nodes are shown in one of three states: above the CIR (clear), below the CIR (dark), or in a condition where the CIR does not matter (gray).

Figure 2: Hierarchical Schedulers and Priorities



In the figure, the strict-high queue for customer VLAN 0 (cvlan 0) receives service first, even though the customer VLAN is above the configured CIR (see [Table 4 on page 12](#) for the reason: strict-high always has hardware priority 0 regardless of CIR state). Once that queue has been drained, and the priority of the node has become 3 instead of 0 (due to the lack of strict-high traffic), the system moves on to the medium queues next (cvlan 1 and cvlan 3), draining them in a round robin fashion (empty queue lose their hardware priority). The low queue on cvlan 4 (priority 2) is sent next, because that mode is below the CIR. Then the high queues on cvlan 0 and cvlan 2 (both now with priority 3) are drained in a round robin fashion, and finally the low queue on cvlan 0 is drained (thanks to svlan 0 having a priority of 3).

## RELATED DOCUMENTATION

*CoS on Enhanced IQ2 PICs Overview*

*Enhanced Queuing DPC CoS Properties*

*CoS Features and Limitations on MIC and MPC Interfaces*

*Understanding Hierarchical Scheduling for MIC and MPC Interfaces*

## Hierarchical CoS for Metro Ethernet Environments

In metro Ethernet environments, a virtual LAN (VLAN) typically corresponds to a customer premises equipment (CPE) device and the VLANs are identified by an inner VLAN tag on Ethernet frames (called the customer VLAN, or C-VLAN, tag). A set of VLANs can be grouped at the DSL access multiplexer

(DSLAM) and identified by using the same outer VLAN tag (called the service VLAN, or S-VLAN, tag). The service VLANs are typically gathered at the Broadband Remote Access Server (B-RAS) level. Hierarchical schedulers let you provide shaping and scheduling at the service VLAN level as well as other levels, such as the physical interface. In other words, you can group a set of logical interfaces and then apply scheduling and shaping parameters to the logical interface set as well as to other levels.

You can apply CoS shaping and scheduling at one of four different levels, including the VLAN set level. (Some devices support up to five levels of scheduler hierarchies.)

The supported scheduler hierarchy is as follows:

- The physical interface (level 1)
- The service VLAN (level 2)
- The logical interface or customer VLAN (level 3)
- The queue (level 4)

Users can specify a traffic control profile (`output-traffic-control-profile`) that can specify a shaping rate, a guaranteed rate, and a scheduler map with transmit rate and buffer delay. The scheduler map contains the mapping of queues (forwarding classes) to their respective schedulers (schedulers define the properties for the queue). Queue properties can specify a transmit rate and buffer management parameters such as buffer size and drop profile.

To configure CoS hierarchical scheduling, you must enable hierarchical scheduling by including the `hierarchical-scheduler` statement at the physical interface.

## RELATED DOCUMENTATION

---

*[Understanding Hierarchical Scheduling](#)*

---

*[Understanding Hierarchical Scheduling for MIC and MPC Interfaces](#)*

---

*[CoS on Enhanced IQ2 PICs Overview](#)*

---

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

## Hierarchical Schedulers and Traffic Control Profiles

When used, the interface set level of the hierarchy falls between the physical interface level (Level 1) and the *logical interface* (Level 3). Queues are always Level 4 of the hierarchy.



**NOTE:** Beginning with Junos OS Release 16.1, certain MPCs on MX Series devices support up to five levels of scheduler hierarchies. The concepts presented in this topic apply similarly to five scheduler hierarchy levels.

Hierarchical schedulers add CoS parameters to the interface-set level of the configuration. They use traffic control profiles to set values for parameters such as shaping rate (the peak information rate [PIR]), guaranteed rate (the committed information rate [CIR] on these interfaces), scheduler maps (assigning queues and resources to traffic), and so on.

The following CoS configuration places the following parameters in traffic control profiles at various levels:

- Traffic control profile at the port level (tcp-port-level1):
  - A shaping rate (PIR) of 100 Mbps
  - A delay buffer rate of 100 Mbps
- Traffic control profile at the interface set level (tcp-interface-level2):
  - A shaping rate (PIR) of 60 Mbps
  - A guaranteed rate (CIR) of 40 Mbps
- Traffic control profile at the logical interface level (tcp-unit-level3):
  - A shaping rate (PIR) of 50 Mbps
  - A guaranteed rate (CIR) of 30 Mbps
  - A scheduler map called `smap1` to hold various queue properties (level 4)
  - A delay buffer rate of 40 Mbps

In this case, the traffic control profiles look like this:

```
[edit class-of-service traffic-control-profiles]
tcp-port-level1 { # This is the physical port level
  shaping-rate 100m;
  delay-buffer-rate 100m;
}
tcp-interface-level2 { # This is the interface set level
  shaping-rate 60m;
  guaranteed-rate 40m;
}
```



```

tcp-unit-level3 { # This is the logical interface level
    shaping-rate 50m;
    guaranteed-rate 30m;
    scheduler-map smap1;
    delay-buffer-rate 40m;
}

```

Once configured, the traffic control profiles must be applied to the proper places in the CoS interfaces hierarchy.

```

[edit class-of-service interfaces]
interface-set level-2 {
    output-traffic-control-profile tcp-interface-level-2;
}
ge-0/1/0 {
    output-traffic-control-profile tcp-port-level-1;
    unit 0 {
        output-traffic-control-profile tcp-unit-level-3;
    }
}

```

In all cases, the properties for level 4 of the hierarchical schedulers are determined by the scheduler map.

### Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
16.1	Beginning with Junos OS Release 16.1, certain MPCs on MX Series devices support up to five levels of scheduler hierarchies.

### RELATED DOCUMENTATION

[Oversubscribing Interface Bandwidth](#)

[Providing a Guaranteed Minimum Rate](#)

[Configuring Scheduler Maps](#)

[Configuring Traffic Control Profiles for Shared Scheduling and Shaping](#)

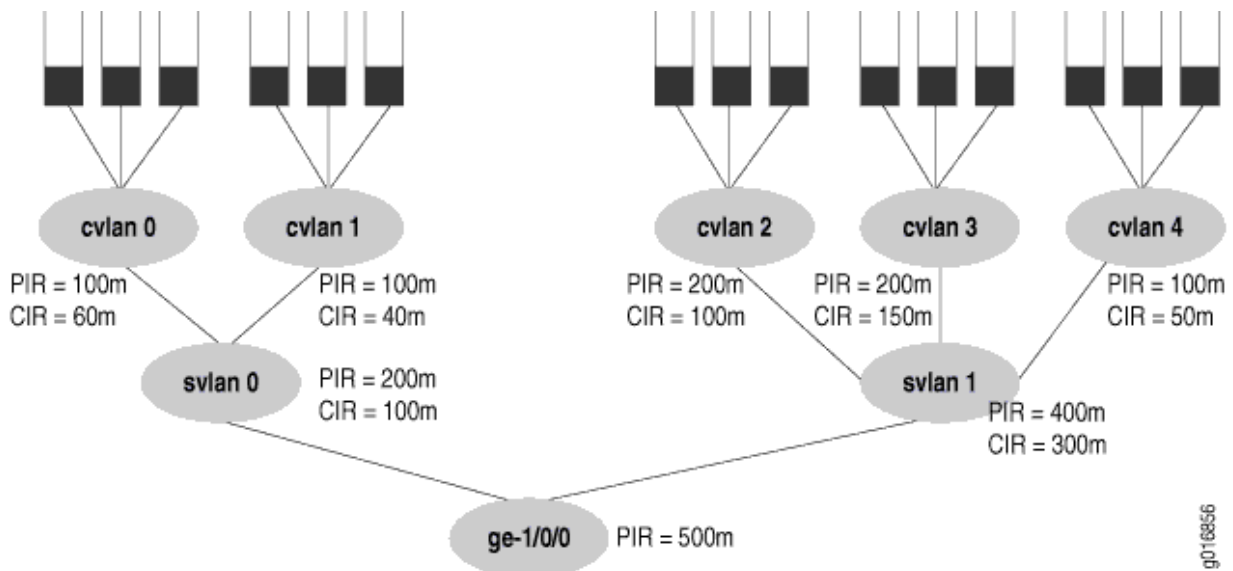
## Example: Building a Four-Level Hierarchy of Schedulers

### IN THIS SECTION

- [Configuring the Interface Sets | 19](#)
- [Configuring the Interfaces | 19](#)
- [Configuring the Traffic Control Profiles | 20](#)
- [Configuring the Schedulers | 21](#)
- [Configuring the Drop Profiles | 22](#)
- [Configuring the Scheduler Maps | 22](#)
- [Applying the Traffic Control Profiles | 23](#)

This section provides a more complete example of building a 4-level hierarchy of schedulers. The configuration parameters are shown in [Figure 3 on page 18](#). The queues are shown at the top of the figure with the other three levels of the hierarchy below.

**Figure 3: Building a Scheduler Hierarchy**



The figure's PIR values are configured as the shaping rates and the CIRs are configured as the guaranteed rate on the Ethernet interface `ge-1/0/0`. The PIR can be oversubscribed (that is, the sum of the children PIRs can exceed the parent's, as in `svlan 1`, where  $200 + 200 + 100$  exceeds the parent rate

of 400)). However, the sum of the children node level's CIRs must never exceed the parent node's CIR, as shown in all the service VLANs (otherwise, the guaranteed rate could never be provided in all cases).

This configuration example presents all details of the CoS configuration for the interface in the figure (ge-1/0/0), including:

## Configuring the Interface Sets

```
[edit interfaces]
interface-set svlan-0 {
  interface ge-1/0/0 {
    unit 0;
    unit 1;
  }
}
interface-set svlan-1 {
  interface ge-1/0/0 {
    unit 2;
    unit 3;
    unit 4;
  }
}
```

## Configuring the Interfaces

The keyword to configure hierarchical schedulers is at the physical interface level, as is VLAN tagging and the VLAN IDs. In this example, the interface sets are defined by logical interfaces (units) and not outer VLAN tags. All VLAN tags in this example are customer VLAN tags.

```
[edit interface ge-1/0/0]
hierarchical-scheduler;
vlan-tagging;
unit 0 {
  vlan-id 100;
}
unit 1 {
  vlan-id 101;
}
unit 2 {
  vlan-id 102;
```

```

}
unit 3 {
    vlan-id 103;
}
unit 4 {
    vlan-id 104;
}

```

## Configuring the Traffic Control Profiles

The traffic control profiles hold parameters for levels above the queue level of the scheduler hierarchy. This section defines traffic control profiles for both the service VLAN level (logical interfaces) and the customer VLAN (VLAN tag) level.

```

[edit class-of-service traffic-control-profiles]
tcp-500m-shaping-rate {
    shaping-rate 500m;
}
tcp-svlan0 {
    shaping-rate 200m;
    guaranteed-rate 100m;
    delay-buffer-rate 300m; # This parameter is not shown in the figure.
}
tcp-svlan1 {
    shaping-rate 400m;
    guaranteed-rate 300m;
    delay-buffer-rate 100m; # This parameter is not shown in the figure.
}
tcp-cvlan0 {
    shaping-rate 100m;
    guaranteed-rate 60m;
    scheduler-map tcp-map-cvlan0; # Applies scheduler maps to customer VLANs.
}
tcp-cvlan1 {
    shaping-rate 100m;
    guaranteed-rate 40m;
    scheduler-map tcp-map-cvlan1; # Applies scheduler maps to customer VLANs.
}
tcp-cvlan2 {
    shaping-rate 200m;
    guaranteed-rate 100m;
}

```

```

    scheduler-map tcp-map-cvlanx; # Applies scheduler maps to customer VLANs.
}
tcp-cvlan3 {
    shaping-rate 200m;
    guaranteed-rate 150m;
    scheduler-map tcp-map-cvlanx; # Applies scheduler maps to customer VLANs
}
tcp-cvlan4 {
    shaping-rate 100m;
    guaranteed-rate 50m;
    scheduler-map tcp-map-cvlanx; # Applies scheduler maps to customer VLANs
}

```

## Configuring the Schedulers

The schedulers hold the information about the queues, the last level of the hierarchy. Note the consistent naming schemes applied to repetitive elements in all parts of this example.

```

[edit class-of-service schedulers]
sched-cvlan0-qx {
    priority low;
    transmit-rate 20m;
    buffer-size temporal 100ms;
    drop-profile loss-priority low dp-low;
    drop-profile loss-priority high dp-high;
}
sched-cvlan1-q0 {
    priority high;
    transmit-rate 20m;
    buffer-size percent 40;
    drop-profile loss-priority low dp-low;
    drop-profile loss-priority high dp-high;
}
sched-cvlanx-qx {
    transmit-rate percent 30;
    buffer-size percent 30;
    drop-profile loss-priority low dp-low;
    drop-profile loss-priority high dp-high;
}
sched-cvlan1-qx {
    transmit-rate 10m;
}

```

```

buffer-size temporal 100ms;
drop-profile loss-priority low dp-low;
drop-profile loss-priority high dp-high;
}

```

## Configuring the Drop Profiles

This section configures the drop profiles for the example. For more information about interpolated drop profiles, see *RED Drop Profiles for Congestion Management*.

```

[edit class-of-service drop-profiles]
dp-low {
    interpolate fill-level 80 drop-probability 80;
    interpolate fill-level 100 drop-probability 100;
}
dp-high {
    interpolate fill-level 60 drop-probability 80;
    interpolate fill-level 80 drop-probability 100;
}

```

## Configuring the Scheduler Maps

This section configures the scheduler maps for the example. Each one references a scheduler configured in ["Configuring the Schedulers" on page 21](#).

```

[edit class-of-service scheduler-maps]
tcp-map-cvlan0 {
    forwarding-class voice scheduler sched-cvlan0-qx;
    forwarding-class video scheduler sched-cvlan0-qx;
    forwarding-class data scheduler sched-cvlan0-qx;
}
tcp-map-cvlan1 {
    forwarding-class voice scheduler sched-cvlan1-q0;
    forwarding-class video scheduler sched-cvlan1-qx;
    forwarding-class data scheduler sched-cvlan1-qx;
}
tcp-map-cvlanx {
    forwarding-class voice scheduler sched-cvlanx-qx;
    forwarding-class video scheduler sched-cvlanx-qx;
}

```

```

forwarding-class data scheduler sched-cvlanx-qx;
}

```

## Applying the Traffic Control Profiles

This section applies the traffic control profiles to the proper levels of the hierarchy.



**NOTE:** Although a shaping rate can be applied directly to the physical interface, hierarchical schedulers must use a traffic control profile to hold this parameter.

```

[edit class-of-service interfaces]
ge-1/0/0 {
  output-traffic-control-profile tcp-500m-shaping-rate;
  unit 0 {
    output-traffic-control-profile tcp-cvlan0;
  }
  unit 1 {
    output-traffic-control-profile tcp-cvlan1;
  }
  unit 2 {
    output-traffic-control-profile tcp-cvlan2;
  }
  unit 3 {
    output-traffic-control-profile tcp-cvlan3;
  }
  unit 4 {
    output-traffic-control-profile tcp-cvlan4;
  }
}
interface-set svlan0 {
  output-traffic-control-profile tcp-svlan0;
}
interface-set svlan1 {
  output-traffic-control-profile tcp-svlan1;
}

```



**NOTE:** You should be careful when using a `show interfaces queue` command that references nonexistent class-of-service logical interfaces. When multiple logical interfaces (units)

are not configured under the same interface set or physical interface, but are referenced by a command such as `show interfaces queue ge-10/0/1.12 forwarding-class be` or `show interfaces queue ge-10/0/1.13 forwarding-class be` (where logical units 12 and 13 are not configured as a class-of-service interfaces), these interfaces display the same traffic statistics for each logical interface. In other words, even if there is no traffic passing through a particular unconfigured logical interface, as long as one or more of the other unconfigured logical interfaces under the same interface set or physical interface is passing traffic, this particular logical interface displays statistics counters showing the total amount of traffic passed through all other unconfigured logical interfaces together.

## Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels

### IN THIS SECTION

- [Understanding Scheduling and Shaping of Traffic Routed to GRE Tunnels | 24](#)
- [Configuration Overview | 25](#)
- [Configuration Caveats | 25](#)

This topic covers the following information:

### Understanding Scheduling and Shaping of Traffic Routed to GRE Tunnels

On MX Series routers running Junos OS Release 12.3R4 or later revisions, 13.2R2 or later revision, or 13.3R1 or later, you can manage CoS scheduling and shaping of traffic routed to generic route encapsulation (GRE) tunnel interfaces configured on [MPC1 Q](#), [MPC2 Q](#), or [MPC2 EQ](#) modules.

A single egress logical interface can be converted to multiple GRE tunnel interfaces. A GRE tunnel physical interface can support many logical interfaces, but one or more of those logical interfaces might not have an output traffic control profiles attached. If a GRE tunnel logical interface is not attached to an output traffic control profile, the router does not assign the interface a dedicated scheduler. Instead, the interface uses a reserved scheduler intended for all *unshaped tunnel traffic* (traffic entering a GRE tunnel logical interface that does not have an explicit traffic control profile configuration).



## Configuration Overview

At GRE tunnel interfaces, the `output-traffic-control-profile` configuration statement can apply an output traffic scheduling and shaping profile at the physical or logical interface level, while the `output-traffic-control-profile-remaining` configuration statement can apply an output traffic scheduling and shaping profile for remaining traffic at the physical interface level only. Interface sets (sets of interfaces used to configure hierarchical CoS schedulers on supported Ethernet interfaces) are not supported on GRE tunnel interfaces.

By default—if you do not attach an output traffic control profile to the GRE tunnel physical interface—traffic entering the interface is scheduled and shaped using the default 95/5 scheduler with parameters as specified in the `tunnel-services` configuration.

If you use an output traffic control profile to configure the shaping rate at the GRE tunnel physical interface, the `shaping-rate` specified by the attached traffic control profile overrides the `bandwidth` specified as the tunnel services default value.

## Configuration Caveats

When configuring hierarchical CoS scheduling and shaping of traffic routed to GRE tunnels, keep the following guidelines in mind:

- You must first configure and commit a hierarchical scheduler on the GRE tunnel physical interface, specifying a maximum of two hierarchical scheduling levels for node scaling. After you commit the `hierarchical-scheduler` configuration, you can configure scheduling and queuing parameters at the GRE tunnel physical or logical interfaces.
- GRE tunnel interfaces support eight egress queues only. For interfaces on MPC1 Q, MPC2 Q, and MPC2 EQ modules, you can include the `max-queues-per-interface 4` statement at the `[edit fpc slot-number pic pic-number]` hierarchy level to configure four-queue mode for the interface. However, any GRE tunnel interfaces configured on those ports have eight queues.
- Queuing and scheduling calculations include Layer 3 fields. For GRE interfaces, Layer 3 fields include the delivery header (the outer IP header), the 4-byte GRE header, and the payload protocol header and data.

## RELATED DOCUMENTATION

*Example: Performing Output Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels*

*Per-Unit Queuing and Hierarchical Queuing for MIC and MPC Interfaces*

*Understanding Hierarchical Scheduling for MIC and MPC Interfaces*

## Example: Performing Output Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels

### IN THIS SECTION

- [Requirements | 26](#)
- [Overview | 27](#)
- [Configuration | 27](#)
- [Verification | 41](#)

This example shows how to configure a generic routing encapsulation (GRE) tunnel device to perform CoS output scheduling and shaping of IPv4 traffic routed to GRE tunnels. This feature is supported on MX Series routers running Junos OS Release 12.3R4 or later revisions, 13.2R2 or later revision, or 13.3R1 or later, with GRE tunnel interfaces configured on [MPC1 Q](#), [MPC2 Q](#), or [MPC2 EQ](#) modules.

### Requirements

This example uses the following Juniper Networks hardware and Junos OS software:

- Transport network—An IPv4 network running Junos OS Release 13.3.
- GRE tunnel device—One MX80 router installed as an ingress provider edge (PE) router.
- Input and output logical interfaces configurable on two ports of the built-in 10-Gigabit Ethernet Modular Interface Card (MIC):
  - Input logical interface `ge-1/1/0.0` for receiving traffic that is to be transported across the network.
  - Output logical interfaces `ge-1/1/1.0`, `ge-1/1/1.1`, and `ge-1/1/1.2` to convert to GRE tunnel source interfaces `gr-1/1/10.1`, `gr-1/1/10.2`, and `gr-1/1/10.3`.

For information about interfaces hosted on modules in MX80 routers, see the following topics:

- [MX5, MX10, MX40, and MX80 Modular Interface Card Description](#)
- [MX5, MX10, MX40, and MX80 Port and Interface Numbering](#)

## Overview

In this example, you configure the router with input and output logical interfaces for IPv4 traffic, and then you convert the output logical interface to four GRE tunnel source interfaces. You also install static routes in the routing table so that input traffic is routed to the four GRE tunnels.



**NOTE:** Before you apply a traffic control profile with a scheduler-map and shaping rate to a GRE tunnel interface, you must configure and commit a hierarchical scheduler on the GRE tunnel physical interface, specifying a maximum of two hierarchical scheduling levels for node scaling.

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 27](#)
- [Configuring Interfaces, Hierarchical Scheduling on the GRE Tunnel Physical Interface, and Static Routes | 30](#)
- [Measuring GRE Tunnel Transmission Rates Without Shaping Applied | 33](#)
- [Configuring Output Scheduling and Shaping at GRE Tunnel Physical and Logical Interfaces | 35](#)

To configure scheduling and shaping in hierarchical CoS queues for traffic routed to GRE tunnel interfaces configured on MPC1Q, MPC2Q, or MPC2 EQ modules on an MX Series router, perform these tasks:

### CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

### Configuring Interfaces, Hierarchical Scheduling on the GRE Tunnel Physical Interface, and Static Routes

```
set chassis fpc 1 pic 1 tunnel-services bandwidth 1g
set interfaces ge-1/1/0 unit 0 family inet address 10.6.6.1/24
set interfaces ge-1/1/1 unit 0 family inet address 10.70.1.1/24 arp 10.70.1.3 mac
00:00:03:00:04:00
set interfaces ge-1/1/1 unit 0 family inet address 10.80.1.1/24 arp 10.80.1.3 mac
```

```

00:00:03:00:04:01
set interfaces ge-1/1/1 unit 0 family inet address 10.90.1.1/24 arp 10.90.1.3 mac
00:00:03:00:04:02
set interfaces ge-1/1/1 unit 0 family inet address 10.100.1.1/24 arp 10.100.1.3 mac
00:00:03:00:04:04
set interfaces gr-1/1/10 unit 1 family inet address 10.100.1.1/24
set interfaces gr-1/1/10 unit 1 tunnel source 10.70.1.1 destination 10.70.1.3
set interfaces gr-1/1/10 unit 2 family inet address 10.200.1.1/24
set interfaces gr-1/1/10 unit 2 tunnel source 10.80.1.1 destination 10.80.1.3
set interfaces gr-1/1/10 unit 3 family inet address 10.201.1.1/24
set interfaces gr-1/1/10 unit 3 tunnel source 10.90.1.1 destination 10.90.1.3
set interfaces gr-1/1/10 unit 4 family inet address 10.202.1.1/24
set interfaces gr-1/1/10 unit 4 tunnel source 10.100.1.1 destination 10.100.1.3
set interfaces gr-1/1/10 hierarchical-scheduler
set routing-options static route 10.2.2.0/24 next-hop gr-1/1/10.1
set routing-options static route 10.3.3.0/24 next-hop gr-1/1/10.2
set routing-options static route 10.4.4.0/24 next-hop gr-1/1/10.3
set routing-options static route 10.5.5.0/24 next-hop gr-1/1/10.4

```

### Configuring Output Scheduling and Shaping at GRE Tunnel Physical and Logical Interfaces

```

set class-of-service forwarding-classes queue 0 be
set class-of-service forwarding-classes queue 1 ef
set class-of-service forwarding-classes queue 2 af
set class-of-service forwarding-classes queue 3 nc
set class-of-service forwarding-classes queue 4 be1
set class-of-service forwarding-classes queue 5 ef1
set class-of-service forwarding-classes queue 6 af1
set class-of-service forwarding-classes queue 7 nc1
set class-of-service classifiers inet-precedence gr-inet forwarding-class be loss-priority low
code-points 000
set class-of-service classifiers inet-precedence gr-inet forwarding-class ef loss-priority low
code-points 001
set class-of-service classifiers inet-precedence gr-inet forwarding-class af loss-priority low
code-points 010
set class-of-service classifiers inet-precedence gr-inet forwarding-class nc loss-priority low
code-points 011
set class-of-service classifiers inet-precedence gr-inet forwarding-class be1 loss-priority low
code-points 100
set class-of-service classifiers inet-precedence gr-inet forwarding-class ef1 loss-priority low
code-points 101
set class-of-service classifiers inet-precedence gr-inet forwarding-class af1 loss-priority low

```

```
code-points 110
set class-of-service classifiers inet-precedence gr-inet forwarding-class nc1 loss-priority low
code-points 111
set class-of-service interfaces ge-1/1/0 unit 0 classifiers inet-precedence gr-inet
set class-of-service schedulers be_sch transmit-rate percent 30
set class-of-service schedulers ef_sch transmit-rate percent 40
set class-of-service schedulers af_sch transmit-rate percent 25
set class-of-service schedulers nc_sch transmit-rate percent 5
set class-of-service schedulers be1_sch transmit-rate percent 60
set class-of-service schedulers be1_sch priority low
set class-of-service schedulers ef1_sch transmit-rate percent 40
set class-of-service schedulers ef1_sch priority medium-low
set class-of-service schedulers af1_sch transmit-rate percent 10
set class-of-service schedulers af1_sch priority strict-high
set class-of-service schedulers nc1_sch shaping-rate percent 10
set class-of-service schedulers nc1_sch priority high
set class-of-service scheduler-maps sch_map_1 forwarding-class be scheduler be_sch
set class-of-service scheduler-maps sch_map_1 forwarding-class ef scheduler ef_sch
set class-of-service scheduler-maps sch_map_1 forwarding-class af scheduler af_sch
set class-of-service scheduler-maps sch_map_1 forwarding-class nc scheduler nc_sch
set class-of-service scheduler-maps sch_map_2 forwarding-class be scheduler be1_sch
set class-of-service scheduler-maps sch_map_2 forwarding-class ef scheduler ef1_sch
set class-of-service scheduler-maps sch_map_3 forwarding-class af scheduler af_sch
set class-of-service scheduler-maps sch_map_3 forwarding-class nc scheduler nc_sch
set class-of-service traffic-control-profiles gr-ifl-tcp3 guaranteed-rate 5m
set class-of-service traffic-control-profiles gr-ifd-tcp shaping-rate 10m
set class-of-service traffic-control-profiles gr-ifd-tcp-remain shaping-rate 7m
set class-of-service traffic-control-profiles gr-ifd-tcp-remain guaranteed-rate 4m
set class-of-service traffic-control-profiles gr-ifl-tcp1 scheduler-map sch_map_1
set class-of-service traffic-control-profiles gr-ifl-tcp1 shaping-rate 8m
set class-of-service traffic-control-profiles gr-ifl-tcp1 guaranteed-rate 3m
set class-of-service traffic-control-profiles gr-ifl-tcp2 scheduler-map sch_map_2
set class-of-service traffic-control-profiles gr-ifl-tcp2 guaranteed-rate 2m
set class-of-service traffic-control-profiles gr-ifl-tcp3 scheduler-map sch_map_3
set class-of-service interfaces gr-1/1/10 output-traffic-control-profile gr-ifd-tcp
set class-of-service interfaces gr-1/1/10 output-traffic-control-profile-remaining gr-ifd-remain
set class-of-service interfaces gr-1/1/10 unit 1 output-traffic-control-profile gr-ifl-tcp1
set class-of-service interfaces gr-1/1/10 unit 2 output-traffic-control-profile gr-ifl-tcp2
set class-of-service interfaces gr-1/1/10 unit 3 output-traffic-control-profile gr-ifl-tcp3
```

## Configuring Interfaces, Hierarchical Scheduling on the GRE Tunnel Physical Interface, and Static Routes

### Step-by-Step Procedure

To configure GRE tunnel interfaces (including enabling hierarchical scheduling) and static routes:

1. Configure the amount of bandwidth for tunnel services on the physical interface.

```
[edit]
user@host# set chassis fpc 1 pic 1 tunnel-services bandwidth 1g
```

2. Configure the GRE tunnel device output logical interface.

```
[edit]
user@host# set interfaces ge-1/1/0 unit 0 family inet address 10.6.6.1/24
```

3. Configure the GRE tunnel device output logical interface.

```
[edit]
user@host# set interfaces ge-1/1/1 unit 0 family inet address 10.70.1.1/24 arp 10.70.1.3 mac
00:00:03:00:04:00
user@host# set interfaces ge-1/1/1 unit 0 family inet address 10.80.1.1/24 arp 10.80.1.3 mac
00:00:03:00:04:01
user@host# set interfaces ge-1/1/1 unit 0 family inet address 10.90.1.1/24 arp 10.90.1.3 mac
00:00:03:00:04:02
user@host# set interfaces ge-1/1/1 unit 0 family inet address 10.100.1.1/24 arp 10.100.1.3
mac 00:00:03:00:04:04
```

4. Convert the output logical interface to four GRE tunnel interfaces.

```
[edit]
user@host# set interfaces gr-1/1/10 unit 1 family inet address 10.100.1.1/24
user@host# set interfaces gr-1/1/10 unit 1 tunnel source 10.70.1.1 destination 10.70.1.3
user@host# set interfaces gr-1/1/10 unit 2 family inet address 10.200.1.1/24
user@host# set interfaces gr-1/1/10 unit 2 tunnel source 10.80.1.1 destination 10.80.1.3
user@host# set interfaces gr-1/1/10 unit 3 family inet address 10.201.1.1/24
user@host# set interfaces gr-1/1/10 unit 3 tunnel source 10.90.1.1 destination 10.90.1.3
user@host# set interfaces gr-1/1/10 unit 4 family inet address 10.202.1.1/24
user@host# set interfaces gr-1/1/10 unit 4 tunnel source 10.100.1.1 destination 10.100.1.3
```

5. Enable the GRE tunnel interfaces to use hierarchical scheduling.

```
[edit]
user@host# set interfaces gr-1/1/10 hierarchical-scheduler
```

6. Install static routes in the routing table so that the device routes IPv4 traffic to the GRE tunnel source interfaces.

Traffic destined to the subnets 10.2.2.0/24, 10.3.3.0/24, 10.4.4.0/24, and 10.5.5.0/24 is routed to the tunnel interfaces at IP addresses 10.70.1.1, 10.80.1.1, 10.90.1.1, and 10.100.1.1, respectively.

```
[edit]
user@host# set routing-options static route 10.2.2.0/24 next-hop gr-1/1/10.1
user@host# set routing-options static route 10.3.3.0/24 next-hop gr-1/1/10.2
user@host# set routing-options static route 10.4.4.0/24 next-hop gr-1/1/10.3
user@host# set routing-options static route 10.5.5.0/24 next-hop gr-1/1/10.4
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Results

From configuration mode, confirm your configuration by entering the `show chassis fpc 1 pic 1`, `show interfaces ge-1/1/0`, `show interfaces ge-1/1/1`, `show interfaces gr-1/1/10`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Confirm the configuration of interfaces, hierarchical scheduling on the GRE tunnel physical interface, and static routes.

```
user@host# show chassis fpc 1 pic 1
tunnel-services {
  bandwidth 1g;
}

user@host# show interfaces ge-1/1/0
unit 0 {
```

```
family inet {
    address 10.6.6.1/24;
}
}

user@host# show interfaces ge-1/1/1
unit 0 {
    family inet {
        address 10.70.1.1/24 {
            arp 10.70.1.3 mac 00:00:03:00:04:00;
        }
        address 10.80.1.1/24 {
            arp 10.80.1.3 mac 00:00:03:00:04:01;
        }
        address 10.90.1.1/24 {
            arp 10.90.1.3 mac 00:00:03:00:04:02;
        }
        address 10.100.1.1/24 {
            arp 10.100.1.3 mac 00:00:03:00:04:04;
        }
    }
}

user@host# show interfaces gr-1/1/10
hierarchical-scheduler;
unit 1 {
    tunnel {
        destination 10.70.1.3;
        source 10.70.1.1;
    }
    family inet {
        address 10.100.1.1/24;
    }
}
unit 2 {
    tunnel {
        destination 10.80.1.3;
        source 10.80.1.1;
    }
    family inet {
        address 10.200.1.1/24;
    }
}
```



```

unit 3 {
  tunnel {
    destination 10.90.1.3;
    source 10.90.1.1;
  }
  family inet {
    address 10.201.1.1/24;
  }
}
unit 4 {
  tunnel {
    destination 10.100.1.3;
    source 10.100.1.1;
  }
  family inet {
    address 10.202.1.1/24;
  }
}

user@host# show routing-options
static {
  route 10.2.2.0/24 next-hop gr-1/1/10.1;
  route 10.3.3.0/24 next-hop gr-1/1/10.2;
  route 10.4.4.0/24 next-hop gr-1/1/10.3;
  route 10.5.5.0/24 next-hop gr-1/1/10.4;
}

```

## Measuring GRE Tunnel Transmission Rates Without Shaping Applied

### Step-by-Step Procedure

To establish a baseline measurement, note the transmission rates at each GRE tunnel source.

1. Pass traffic through the GRE tunnel at logical interfaces `gr-1/1/10.1`, `gr-1/1/10.2`, and `gr-1/1/10.3`.
2. To display the traffic rates at each GRE tunnel source, use the `show interfaces queue operational mode` command.

The following example command output shows detailed CoS queue statistics for logical interface gr-1/1/10.1 (the GRE tunnel from source IP address 10.70.1.1 to destination IP address 10.70.1.3).

```

user@host> show interfaces queue gr-1/1/10.1
Logical interface gr-1/1/10.1 (Index 331) (SNMP ifIndex 4045)
Forwarding classes: 16 supported, 8 in use
Egress queues: 8 supported, 8 in use
Burst size: 0
Queue: 0, Forwarding classes: be
  Queued:
    Packets      :           31818312           102494 pps
    Bytes        :           6522753960       168091936 bps
  Transmitted:
    Packets      :           1515307           4879 pps
    Bytes        :           310637935         8001632 bps
    Tail-dropped packets :           21013826         68228 pps
    RED-dropped packets :           9289179         29387 pps
      Low        :           9289179         29387 pps
      Medium-low :                0              0 pps
      Medium-high:                0              0 pps
      High       :                0              0 pps
    RED-dropped bytes :           1904281695       48194816 bps
      Low        :           1904281695       48194816 bps
      Medium-low :                0              0 bps
      Medium-high:                0              0 bps
      High       :                0              0 bps
  ...

```



**NOTE:** This step shows command output for queue 0 (forwarding class be) only.

The command output shows that the GRE tunnel device transmits traffic from queue 0 at a rate of 4879 pps. Allowing for 182 bytes per Layer 3 packet, preceded by 24 bytes of GRE overhead (a 20-byte delivery header consisting of the IPv4 packet header followed by 4 bytes for GRE flags plus encapsulated protocol type), the traffic rate received at the tunnel destination device is 8,040,592 bps:

The command output shows that the GRE tunnel device transmits traffic from queue 0 at a rate of 4879 pps. Allowing for 182 bytes per Layer 3 packet, preceded by 24 bytes of GRE overhead (a 20-byte delivery header consisting of the IPv4 packet header followed by 4 bytes for GRE flags plus

encapsulated protocol type), the traffic rate received at the tunnel destination device is 8,040,592 bps:

```
4879 packets/second X 206 bytes/packet X 8 bits/byte = 8,040,592 bits/second
```

## Configuring Output Scheduling and Shaping at GRE Tunnel Physical and Logical Interfaces

### Step-by-Step Procedure

To configure the GRE tunnel device with scheduling and shaping at GRE tunnel physical and logical interfaces:

1. Define eight transmission queues.

```
[edit]
user@host# set class-of-service forwarding-classes queue 0 be
user@host# set class-of-service forwarding-classes queue 1 ef
user@host# set class-of-service forwarding-classes queue 2 af
user@host# set class-of-service forwarding-classes queue 3 nc
user@host# set class-of-service forwarding-classes queue 4 be1
user@host# set class-of-service forwarding-classes queue 5 ef1
user@host# set class-of-service forwarding-classes queue 6 af1
user@host# set class-of-service forwarding-classes queue 7 nc1
```



**NOTE:** To configure up to eight forwarding classes with one-to-one mapping to output queues for interfaces on M120, M320, MX Series, and T Series routers and EX Series switches, use the `queue` statement at the `[edit class-of-service forwarding-classes]` hierarchy level.

If you need to configure up to 16 forwarding classes with multiple forwarding classes mapped to single queues for those interface types, use the `class` statement instead.

2. Configure BA classifier `gr-inet` that, based on IPv4 precedence bits set in an incoming packet, sets the forwarding class, loss-priority value, and DSCP bits of the packet.

```
[edit]
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class be loss-
priority low code-points 000
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class ef loss-
```

```

priority low code-points 001
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class af loss-
priority low code-points 010
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class nc loss-
priority low code-points 011
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class be1 loss-
priority low code-points 100
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class ef1 loss-
priority low code-points 101
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class af1 loss-
priority low code-points 110
user@host# set class-of-service classifiers inet-precedence gr-inet forwarding-class nc1 loss-
priority low code-points 111

```

3. Apply BA classifier `gr-inet` to the GRE tunnel device input at logical interface `ge-1/1/0.0`.

```

[edit]
user@host# set class-of-service interfaces ge-1/1/0 unit 0 classifiers inet-precedence gr-inet

```

4. Define a scheduler for each forwarding class.

```

[edit]
user@host# set class-of-service schedulers be_sch transmit-rate percent 30
user@host# set class-of-service schedulers ef_sch transmit-rate percent 40
user@host# set class-of-service schedulers af_sch transmit-rate percent 25
user@host# set class-of-service schedulers nc_sch transmit-rate percent 5
user@host# set class-of-service schedulers be1_sch transmit-rate percent 60
user@host# set class-of-service schedulers be1_sch priority low
user@host# set class-of-service schedulers ef1_sch transmit-rate percent 40
user@host# set class-of-service schedulers ef1_sch priority medium-low
user@host# set class-of-service schedulers af1_sch transmit-rate percent 10
user@host# set class-of-service schedulers af1_sch priority strict-high
user@host# set class-of-service schedulers nc1_sch shaping-rate percent 10
user@host# set class-of-service schedulers nc1_sch priority high

```

5. Define a scheduler map for each of three GRE tunnels.

```

[edit]
user@host# set class-of-service scheduler-maps sch_map_1 forwarding-class be scheduler be_sch
user@host# set class-of-service scheduler-maps sch_map_1 forwarding-class ef scheduler ef_sch

```

```

user@host# set class-of-service scheduler-maps sch_map_1 forwarding-class af scheduler af_sch
user@host# set class-of-service scheduler-maps sch_map_1 forwarding-class nc scheduler nc_sch
user@host# set class-of-service scheduler-maps sch_map_2 forwarding-class be scheduler be1_sch
user@host# set class-of-service scheduler-maps sch_map_2 forwarding-class ef scheduler ef1_sch
user@host# set class-of-service scheduler-maps sch_map_3 forwarding-class af scheduler af_sch
user@host# set class-of-service scheduler-maps sch_map_3 forwarding-class nc scheduler nc_sch

```

6. Define traffic control profiles for three GRE tunnel interfaces.

```

[edit]
user@host# set class-of-service traffic-control-profiles gr-if1-tcp1 scheduler-map sch_map_1
user@host# set class-of-service traffic-control-profiles gr-if1-tcp1 shaping-rate 8m
user@host# set class-of-service traffic-control-profiles gr-if1-tcp1 guaranteed-rate 3m
user@host# set class-of-service traffic-control-profiles gr-if1-tcp2 scheduler-map sch_map_2
user@host# set class-of-service traffic-control-profiles gr-if1-tcp2 guaranteed-rate 2m
user@host# set class-of-service traffic-control-profiles gr-if1-tcp3 scheduler-map sch_map_3
user@host# set class-of-service traffic-control-profiles gr-if1-tcp3 guaranteed-rate 5m
user@host# set class-of-service traffic-control-profiles gr-if1-tcp shaping-rate 10m
user@host# set class-of-service traffic-control-profiles gr-if1-tcp-remain shaping-rate 7m
user@host# set class-of-service traffic-control-profiles gr-if1-tcp-remain guaranteed-rate 4m

```

7. Apply CoS scheduling and shaping to the output traffic at the physical interface and logical interfaces.

```

[edit]
user@host# set class-of-service interfaces gr-1/1/10 output-traffic-control-profile gr-ifd-tcp
user@host# set class-of-service interfaces gr-1/1/10 output-traffic-control-profile-remaining
gr-ifd-remain
user@host# set class-of-service interfaces gr-1/1/10 unit 1 output-traffic-control-profile gr-
if1-tcp1
user@host# set class-of-service interfaces gr-1/1/10 unit 2 output-traffic-control-profile gr-
if1-tcp2
user@host# set class-of-service interfaces gr-1/1/10 unit 2 output-traffic-control-profile gr-
if1-tcp3

```

8. If you are done configuring the device, commit the configuration.

```

[edit]
user@host# commit

```

## Results

From configuration mode, confirm your configuration by entering the `show class-of-service forwarding-classes`, `show class-of-service classifiers`, `show class-of-service interfaces ge-1/1/0`, `show class-of-service schedulers`, `show class-of-service scheduler-maps`, `show class-of-service traffic-control-profiles`, and `show class-of-service interfaces gr-1/1/10` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Confirm the configuration of output scheduling and shaping at the GRE tunnel physical and logical interfaces.

```
user@host# show class-of-service forwarding-classes
```

```
queue 0 be;
queue 1 ef;
queue 2 af;
queue 3 nc;
queue 4 be1;
queue 5 ef1;
queue 6 af1;
queue 7 nc1;
```

```
user@host# show class-of-service classifiers
```

```
inet-precedence gr-inet {
  forwarding-class be {
    loss-priority low code-points 000;
  }
  forwarding-class ef {
    loss-priority low code-points 001;
  }
  forwarding-class af {
    loss-priority low code-points 010;
  }
  forwarding-class nc {
    loss-priority low code-points 011;
  }
  forwarding-class be1 {
    loss-priority low code-points 100;
  }
  forwarding-class ef1 {
    loss-priority low code-points 101;
  }
  forwarding-class af1 {
```

```
        loss-priority low code-points 110;
    }
    forwarding-class nc1 {
        loss-priority low code-points 111;
    }
}

user@host# show class-of-service interfaces ge-1/1/0
unit 0 {
    classifiers {
        inet-precedence gr-inet;
    }
}

user@host# show class-of-service schedulers
be_sch {
    transmit-rate percent 30;
}
ef_sch {
    transmit-rate percent 40;
}
af_sch {
    transmit-rate percent 25;
}
nc_sch {
    transmit-rate percent 5;
}
be1_sch {
    transmit-rate percent 60;
    priority low;
}
ef1_sch {
    transmit-rate percent 40;
    priority medium-low;
}
af1_sch {
    transmit-rate percent 10;
    priority strict-high;
}
nc1_sch {
    shaping-rate percent 10;
    priority high;
}
```

```
user@host# show class-of-service scheduler-maps
sch_map_1 {
    forwarding-class be scheduler be_sch;
    forwarding-class ef scheduler ef_sch;
    forwarding-class af scheduler af_sch;
    forwarding-class nc scheduler nc_sch;
}
sch_map_2 {
    forwarding-class be scheduler be1_sch;
    forwarding-class ef scheduler ef1_sch;
}
sch_map_3 {
    forwarding-class af scheduler af_sch;
    forwarding-class nc scheduler nc_sch;
}

user@host# show class-of-service traffic-control-profiles
gr-ifl-tcp1 {
    scheduler-map sch_map_1;
    shaping-rate 8m;
    guaranteed-rate 3m;
}
gr-ifl-tcp2 {
    scheduler-map sch_map_2;
    guaranteed-rate 2m;
}
gr-ifl-tcp3 {
    scheduler-map sch_map_3;
    guaranteed-rate 5m;
}
gr-ifd-remain {
    shaping-rate 7m;
    guaranteed-rate 4m;
}
gr-ifd-tcp {
    shaping-rate 10m;
}

user@host# show class-of-service interfaces gr-1/1/10
gr-1/1/10 {
    output-traffic-control-profile gr-ifd-tcp;
    output-traffic-control-profile-remaining gr-ifd-remain;
}
```



```

unit 1 {
    output-traffic-control-profile gr-ifl-tcp1;
}
unit 2 {
    output-traffic-control-profile gr-ifl-tcp2;
}
unit 3 {
    output-traffic-control-profile gr-ifl-tcp3;
}
}

```

## Verification

### IN THIS SECTION

- [Verifying That Scheduling and Shaping Are Attached to the GRE Tunnel Interfaces | 41](#)
- [Verifying That Scheduling and Shaping Are Functioning at the GRE Tunnel Interfaces | 43](#)

Confirm that the configurations are working properly.

### Verifying That Scheduling and Shaping Are Attached to the GRE Tunnel Interfaces

#### Purpose

Verify the association of traffic control profiles with GRE tunnel interfaces.

#### Action

Verify the traffic control profile attached to the GRE tunnel physical interface by using the **show class-of-service interface gr-1/1/10 detail** operational mode command.

- user@host> **show class-of-service interface gr-1/1/10 detail**  
Physical interface: gr-1/1/10, Enabled, Physical link is Up  
Type: GRE, Link-level type: GRE, MTU: Unlimited, Speed: 1000mbps  
Device flags : Present Running  
Interface flags: Point-To-Point SNMP-Traps  
  
Physical interface: gr-1/1/10, Index: 220

```

Queues supported: 8, Queues in use: 8
  Output traffic control profile: gr-ifd-tcp, Index: 17721
  Output traffic control profile remaining: gr-ifd-remain, Index: 58414
  Congestion-notification: Disabled

Logical interface gr-1/1/10.1
  Flags: Point-To-Point SNMP-Traps 0x4000 IP-Header
10.70.1.3:10.70.1.1:47:df:64:0000000000000000 Encapsulation: GRE-NULL
  Gre keepalives configured: Off, Gre keepalives adjacency state: down
  inet 10.100.1.1/24
  Logical interface: gr-1/1/10.1, Index: 331
Object          Name                Type                Index
Traffic-control-profile gr-if1-tcp1        Output              17849
Classifier      ipprec-compatibility ip                  13

Logical interface gr-1/1/10.2
  Flags: Point-To-Point SNMP-Traps 0x4000 IP-Header
10.80.1.3:10.80.1.1:47:df:64:0000000000000000 Encapsulation: GRE-NULL
  Gre keepalives configured: Off, Gre keepalives adjacency state: down
  inet 10.200.1.1/24
  Logical interface: gr-1/1/10.2, Index: 332
Object          Name                Type                Index
Traffic-control-profile gr-if1-tcp2        Output              17856
Classifier      ipprec-compatibility ip                  13

Logical interface gr-1/1/10.3
  Flags: Point-To-Point SNMP-Traps 0x4000 IP-Header
10.90.1.3:10.90.1.1:47:df:64:0000000000000000 Encapsulation: GRE-NULL
  Gre keepalives configured: Off, Gre keepalives adjacency state: down
  inet 10.201.1.1/24
  Logical interface: gr-1/1/10.3, Index: 333
Object          Name                Type                Index
Traffic-control-profile gr-if1-tcp3        Output              17863
Classifier      ipprec-compatibility ip                  13

```

## Meaning

Ingress IPv4 traffic routed to GRE tunnels on the device is subject to CoS output scheduling and shaping.

## Verifying That Scheduling and Shaping Are Functioning at the GRE Tunnel Interfaces

### Purpose

Verify the traffic rate shaping at the GRE tunnel interfaces.

### Action

1. Pass traffic through the GRE tunnel at logical interfaces `gr-1/1/10.1`, `gr-1/1/10.2`, and `gr-1/1/10.3`.
2. To verify the rate shaping at each GRE tunnel source, use the `show interfaces queue operational mode` command.

The following example command output shows detailed CoS queue statistics for logical interface `gr-1/1/10.1` (the GRE tunnel from source IP address `10.70.1.1` to destination IP address `10.70.1.3`):

```

user@host> show interfaces queue gr-1/1/10.1
Logical interface gr-1/1/10.1 (Index 331) (SNMP ifIndex 4045)
Forwarding classes: 16 supported, 8 in use
Egress queues: 8 supported, 8 in use
Burst size: 0
Queue: 0, Forwarding classes: be
  Queued:
    Packets      :          59613061          51294 pps
    Bytes        :       12220677505       84125792 bps
  Transmitted:
    Packets      :          2230632           3039 pps
    Bytes        :       457279560        4985440 bps
  Tail-dropped packets :          4471146           2202 pps
  RED-dropped packets :          52911283          46053 pps
    Low         :          49602496          46053 pps
    Medium-low  :                   0              0 pps
    Medium-high :                   0              0 pps
    High        :          3308787           0 pps
  RED-dropped bytes   :       10846813015       75528000 bps
    Low             :       10168511680       75528000 bps
    Medium-low     :                   0              0 bps
    Medium-high    :                   0              0 bps
    High           :          678301335           0 bps
Queue: 1, Forwarding classes: ef
  Queued:
    Packets      :          15344874          51295 pps

```

Bytes	:	3145699170	84125760 bps
Transmitted:			
Packets	:	366115	1218 pps
Bytes	:	75053575	1997792 bps
Tail-dropped packets	:	364489	1132 pps
RED-dropped packets	:	14614270	48945 pps
Low	:	14614270	48945 pps
Medium-low	:	0	0 pps
Medium-high	:	0	0 pps
High	:	0	0 pps
RED-dropped bytes	:	2995925350	80270528 bps
Low	:	2995925350	80270528 bps
Medium-low	:	0	0 bps
Medium-high	:	0	0 bps
High	:	0	0 bps
...			



**NOTE:** This step shows command output for queue 0 (forwarding class be) and queue 1 (forwarding class ef) only.

## Meaning

Now that traffic shaping is attached to the GRE tunnel interfaces, the command output shows that traffic shaping specified for the tunnel at logical interface gr-1/1/10.1 (shaping-rate 8m and guaranteed-rate 3m) is honored.

- For queue 0, the GRE tunnel device transmits traffic at a rate of 3039 pps. The traffic rate received at the tunnel destination device is 5,008,272 bps:

$$3039 \text{ packets/second} \times 206 \text{ bytes/packet} \times 8 \text{ bits/byte} = 5,008,272 \text{ bits/second}$$

- For queue 0, the GRE tunnel device transmits traffic at a rate of 1218 pps. The traffic rate received at the tunnel destination device is 2,007,264 bps:

$$1218 \text{ packets/second} \times 206 \text{ bytes/packet} \times 8 \text{ bits/byte} = 2,007,264 \text{ bits/second}$$

Compare these statistics to the baseline measurements taken without traffic shaping, as described in ["Measuring GRE Tunnel Transmission Rates Without Shaping Applied" on page 33](#).

## RELATED DOCUMENTATION

*Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels*

*Configuring Traffic Control Profiles for Shared Scheduling and Shaping*

## Configuring Ingress Hierarchical CoS

You can configure ingress CoS parameters, including hierarchical schedulers, on devices with Enhanced Queuing DPCs (that is, line cards that have a queuing chip). In general, the supported configuration statements apply to per-unit schedulers or to hierarchical schedulers.



**NOTE:** Ingress CoS is not supported on line cards that do not contain a queuing chip.

To configure ingress CoS for per-unit schedulers, include the following statements at the [edit class-of-service interfaces *interface-name*] hierarchy level:



**NOTE:** The input-scheduler-map and input-traffic-control-profile statements are mutually exclusive at the same hierarchy level.

```
[edit class-of-service interfaces interface-name]
input-excess-bandwidth-share (proportional value | equal);
input-scheduler-map map-name;
input-shaping-rate rate;
input-traffic-control-profile profile-name shared-instance instance-name;
unit logical-unit-number;
    input-scheduler-map map-name;
    input-shaping-rate (percent percentage | rate);
    input-traffic-control-profile profile-name shared-instance instance-name;
}
```

To configure ingress CoS for hierarchical schedulers, include the interface-set *interface-set-name* statement at the [edit class-of-service interfaces] hierarchy level:

```
[edit class-of-service interfaces]
interface-set interface-set-name {
    input-excess-bandwidth-share (proportional value | equal);
    input-traffic-control-profile profile-name shared-instance instance-name;
    input-traffic-control-profile-remaining profile-name;
}
```

```

interface interface-name {
    input-excess-bandwidth-share (proportional value | equal);
    input-traffic-control-profile profile-name shared-instance instance-name;
    input-traffic-control-profile-remaining profile-name;
    unit logical-unit-number;
        input-traffic-control-profile profile-name shared-instance instance-name;
    }
}

```

For many platforms, CoS queuing and scheduling are enabled on the egress side but disabled on the ingress side by default. To enable ingress CoS, configure the traffic-manager statement with ingress-and-egress mode:

```

[edit chassis fpc slot-number pic pic-number]
traffic-manager mode ingress-and-egress;

```



**NOTE:** If you enable ingress CoS settings and inline services on the same FPC, the FPC moves to the offline state. This behavior is expected because null route detection is triggered that causes the FPC to move to the offline state.

Configured CoS features on the ingress are independent of CoS features on the egress, with the following exceptions:

- If you configure a per-unit or hierarchical scheduler at the [edit class-of-service interfaces] hierarchy level, the schedulers apply in both the ingress and egress directions.
- You cannot configure the same logical interface on an ingress and an egress interface set. A logical interface can only belong to one interface set.
- The DPC's frame buffer is shared between ingress and egress configurations.

The following behavior aggregate (BA) classification tables are supported on the ingress side:

- inet-precedence
- DSCP
- exp (MPLS)
- DSCP for IPv6
- IEEE 802.1p

## RELATED DOCUMENTATION

*Configuring Traffic Control Profiles for Shared Scheduling and Shaping*

*Enhanced Queuing DPC CoS Properties*

## Hierarchical Class of Service for Network Slicing

### IN THIS SECTION

- [Understanding Network Slicing | 47](#)
- [Workflow for Creating Slices | 49](#)
- [Hierarchical Class of Service \(CoS\) Queuing for Slice Per-hop-behavior | 50](#)

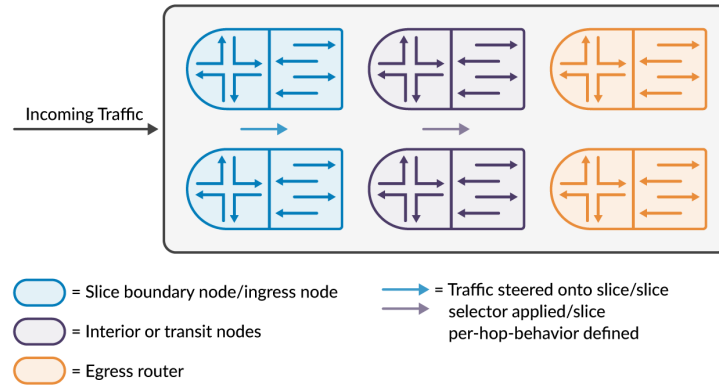
### Understanding Network Slicing

Network slicing is the partitioning of a physical network into multiple logical networks. Each logical network is called a slice. On virtue of being a logical network, a slice is a designated set of network resources, such as interfaces, firewall filters, policers, virtual output queues, schedulers, shapers, traffic control profiles etc. to carry traffic.

### Slice Domain

A set of connected physical nodes such as routers and switches (along with their links) that participate in network slicing is called a slice domain. The slice domain has ingress nodes, transit nodes, and egress nodes. Ingress and egress nodes are located at the borders of the slice domain. The ingress nodes receive traffic into the domain and may classify them before forwarding them to the transit nodes. The egress nodes forward traffic out of the slice domain, and before doing so, may classify the packets.

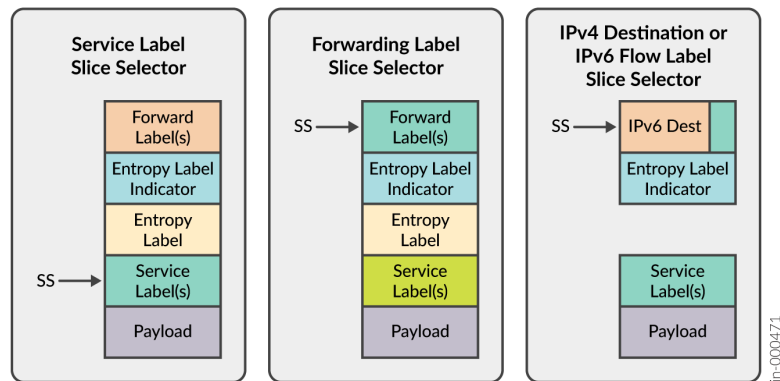
Figure 4: Slice Domain



### Slice Selector

By definition, a slice selector is information in the packet header. The information is used by the boundary nodes and/or transit nodes of a slice domain to classify and/or process packets. There are various options to encapsulate/identify/designate a slice selector in the packet. As an example, a Service Label in the packet header can be used as a slice selector. If defined, this label is allotted a position in the packet header and is checked at this position by firewall filters to determine/designate slices. Similarly, there are several other options as depicted in the following figure.

Figure 5: Slice Selector Types





## Workflow for Creating Slices

Slices as entities are created by specifying them under network slicing hierarchy under services. Then these slices are used to steer packets and to manage traffic destined to slices.

### Hierarchical class of service for slices

You define a traffic control profile for a slice under a physical or aggregated Ethernet interface. Note that you can define traffic control profiles for multiple slices under a physical or aggregated Ethernet interface. See *slice (CoS Interfaces)*.

See "[Hierarchical Class of Service \(CoS\) Queuing for Slice Per-hop-behavior](#)" on page 50 to understand how slices (as part of a hierarchy) are used to control traffic.

### Packet steering

Packet steering is the process of marking/matching packets to/from slices. Packets can be steered using firewall filters (firewall steering) and/or routing policy (route steering).

### Firewall steering

- A firewall filter can be used at the ingress node to mark matched packets as belonging to slices using the "slice" action. See *slice (firewall filter action)*.
- A firewall filter can also be used at the transit node to match slice packets using the "slice" match condition. See *slice (firewall filter match condition)*. The packet can then be marked to another slice if required by the firewall filter or a policer applied to this packet etc.
- Packets that are not marked/matched to to/from slices are processed as non-slice traffic.

### Route steering

- An export policy can be used at the ingress and/or transit node to mark matched routes as belonging to slices. See *slice (export routing policy action)*.
- The export policy can also attach a firewall filter to the route. The firewall filter is used to typically apply a policer to the packets matching the route on the ingress side. This firewall filter is not attached to any interface. Rather, it is part of the forwarding information of the route. See *filter (export routing policy action)*.
- The slice and/or firewall filter will be part of the route's next-hop forwarding information. See `show route extensive expanded-nh` to view slices and/or firewall filters attached to routes.
- Packets that do not match routes with slice information, are classified as non-slice traffic. Packets that match routes with no slice information, are also classified as non-slice traffic.

To summarize, the following are the configurations that are to be enabled before creating slices, can be used to create slices, or manage packets belonging to slices.

- Specify the slices under network slicing hierarchy under services- – Refer to *network-slicing*.
- Enable enhanced-ip mode – Refer *network-services*.
- Perform class of service configuration to enable a slice under an interface and also apply an output traffic control profile for the slice – See *slice (CoS Interfaces)*. See "[Hierarchical Class of Service \(CoS\) Queuing for Slice Per-hop-behavior](#)" on page 50 to understand how slices (as part of a hierarchy) are used to control traffic.
- Configure firewall filters to steer routes to slices and/or match routes from slices – See *slice (firewall filter match condition)* and *slice (firewall filter action)*.
- Use routing policy to steer routes to slices and/or attach firewall – See *slice (export routing policy action)* and *filter (export routing policy action)*.
- View slices and/or firewall filters attached to routes. See *show route extensive <route> expanded-nh*.
- View slices attached to the forwarding table. See *show class-of-service forwarding-table slice*.
- Show mapping of traffic control profiles to slices. See *show class-of-service forwarding-table slice mapping*.
- View traffic control profile(s) attached to a slice under an interface. See *show class-of-service slice <slice\_name> interface <interface\_name>*.
- View statistics for a slice. See *show cos help flow queue-stats*.

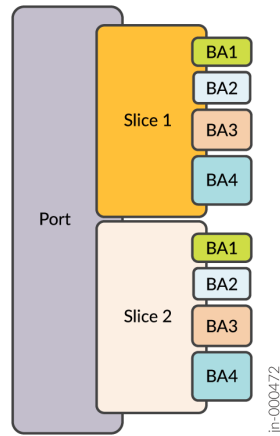
## Hierarchical Class of Service (CoS) Queuing for Slice Per-hop-behavior

In hierarchical CoS, packets are classified at various levels. It could be at the port level, followed by the logical unit level, and then at the queue level. This means that packets are passing through a hierarchy. At every stage of the hierarchy, packets are being classified, shaped, scheduled etc.

In the context of network slicing, a slice also becomes part of the hierarchy. Shapers, schedulers, and traffic control profiles can be applied to the slice. Just as queues are made available to logical units, queues are made available to slices as well.

As the following figure shows, the queues (labeled BA1, BA2, BA3, BA4) are made available to the slice. The queues map to forwarding classes (FCs). Based on behavioral aggregate classifiers, packets are classified into FCs, and subsequently into a corresponding queue (BA1 or BA2 or BA3 etc.).

Figure 6: Hierarchical Queuing



As "[Workflow for Creating Slices](#)" on page 49 describes, you configure network slicing using a combination of firewall filters and Class of Service (CoS) configuration. See *slice (CoS Interfaces)* to read on how to enable slice(s) under any interface using CoS configuration.

# Configuring Hierarchical Class of Service on MICs, MPCs, MLCs, and Aggregated Ethernet Interfaces

## IN THIS CHAPTER

- [Understanding Hierarchical Scheduling for MIC and MPC Interfaces | 52](#)
- [Configuring Ingress Hierarchical CoS on MIC and MPC Interfaces | 55](#)
- [Per-Unit Scheduling and Hierarchical Scheduling for MPC Interfaces | 57](#)
- [Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview | 61](#)
- [Jitter Reduction in Hierarchical CoS Queues | 64](#)
- [Example: Reducing Jitter in Hierarchical CoS Queues | 67](#)
- [Hierarchical Schedulers on Aggregated Ethernet Interfaces Overview | 76](#)
- [Configuring Hierarchical Schedulers on Aggregated Ethernet Interfaces | 77](#)
- [Example: Configuring Scheduling Modes on Aggregated Interfaces | 78](#)
- [Increasing Available Bandwidth on Rich-Queueing MPCs by Bypassing the Queuing Chip | 84](#)

## Understanding Hierarchical Scheduling for MIC and MPC Interfaces

### IN THIS SECTION

- [Scheduler Node Scaling for MIC and MPC Interfaces | 53](#)
- [Hierarchical Scheduling Priority Levels for MIC and MPC Interfaces | 53](#)
- [Guaranteed Bandwidth and Weight of an Interface Node on MIC and MPC Interfaces | 54](#)
- [Hierarchical Scheduling for MIC and MPC Interfaces in Oversubscribed PIR Mode | 54](#)

## Scheduler Node Scaling for MIC and MPC Interfaces

In per-unit scheduling, the logical interfaces share a common level 2 node (one per port). In hierarchical-scheduling, each logical interface has its own level 2 node. Thus, scaling is limited by the number of level 2 nodes.

To better control system resources in hierarchical-scheduling mode, you can limit the number of scheduler node levels to two. In this case, all logical interfaces and interface sets with CoS scheduling policy share a single level 2 node. Consequently, the maximum number of logical interfaces with CoS scheduling policies is increased (the interface sets must be at level 3).

To configure scheduler node scaling, include the `hierarchical-scheduler` statement and set the `maximum-hierarchy-levels` option to 2 at the `[edit interfaces xe-fpc/pic/port]` hierarchy level.

```
[edit interfaces]
xe-2/0/0 {
  hierarchical-scheduler {
    maximum-hierarchy-levels 2;
  }
}
```



**NOTE:** The `maximum-hierarchy-levels` option supports level 3 interface sets but not level 2 interface sets. If you configure level 2 interface sets with the `maximum-hierarchy-levels` option, you generate Packet Forwarding Engine errors.

## Hierarchical Scheduling Priority Levels for MIC and MPC Interfaces

The queuing model used by MIC and MPC interfaces supports three priority levels for guaranteed scheduling priority and two lower priority levels for excess scheduling priority. You can configure a queue with one guaranteed priority and one excess priority. For example, you can configure a queue for guaranteed low (GL) as the guaranteed priority and configure excess high (EH) as the excess priority.

You can associate a guaranteed level with only one excess level. You can associate an excess level with any number of guaranteed priority levels, including none.

Interface nodes maintain their guaranteed priority level (for example, guaranteed high, GH) as long as they do not exceed their guaranteed bandwidth. If the queue bandwidth exceeds the guaranteed rate, then the priority drops to the excess priority (for example, excess high, EH). Because excess level priorities are lower than their guaranteed counterparts, the bandwidth guarantees for each of the other levels can be maintained.

## Guaranteed Bandwidth and Weight of an Interface Node on MIC and MPC Interfaces

The queuing model used by MIC and MPC interfaces separates the concepts of *guaranteed bandwidth* and *weight* of an interface node, although the two terms are often used interchangeably. The guaranteed bandwidth for an interface node is the bandwidth the node can use, independent of what is happening at the other nodes of the scheduling hierarchy. The weight of an interface node, on the other hand, is a value that determines how *excess bandwidth* is used. The weight of a node comes into play when other nodes at the same hierarchical scheduling level use less than the sum of their guaranteed bandwidths

For some application traffic types (such as constant bit rate voice, where there is little concern about excess bandwidth), the guaranteed bandwidth dominates the node. For other types of application traffic (such as bursty data, where a well-defined bandwidth is not always possible), the concept of weight dominates the node.

## Hierarchical Scheduling for MIC and MPC Interfaces in Oversubscribed PIR Mode

In contrast to the Intelligent Queuing Enhanced (IQE) and Intelligent Queuing 2 Enhanced (IQ2E) PICs, the interfaces on MICs and MPCs set the guaranteed rate to zero in oversubscribed peak information rate (PIR) mode for the per-unit scheduler. Also, the configured rate is scaled down to fit the oversubscribed value. For example, if there are two logical interface units with a shaping rate of 1 Gbps each on a 1-Gbps port (which is, therefore, oversubscribed 2 to 1), then the guaranteed rate on each unit is scaled down to 500 Mbps (scaled down by 2).

With hierarchical schedulers in oversubscribed PIR mode, the guaranteed rate for every logical interface unit is set to zero. This means that the queue transmit rates are always oversubscribed.

Because in oversubscribed PIR mode the queue transmit rates are always oversubscribed, the following are true:

- If the queue transmit rate is set as a percentage, then the guaranteed rate of the queue is set to zero; but the excess rate (weight) of the queue is set correctly.
- If the queue transmit rate is set as an absolute value and if the queue has guaranteed high or medium priority, then traffic up to the queue's transmit rate is sent at that priority level. However, for guaranteed low traffic, that traffic is demoted to the excess low region. This means that best-effort traffic well within the queue's transmit rate gets a lower priority than out-of-profile excess high traffic. This differs from the IQE and IQ2E PICs.

### RELATED DOCUMENTATION

*Per-Unit Queuing and Hierarchical Queuing for MIC and MPC Interfaces*

*CoS Features and Limitations on MIC and MPC Interfaces*

*Jitter Reduction in Hierarchical CoS Queues*

*Scheduling and Shaping in Hierarchical CoS Queues for Traffic Routed to GRE Tunnels*

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces](#) | 160

## Configuring Ingress Hierarchical CoS on MIC and MPC Interfaces

Use [Feature Explorer](#) to confirm platform and release support for specific features.

You can configure ingress CoS parameters, including hierarchical schedulers, on supported interfaces. In general, the supported configuration statements apply to per-unit schedulers or to hierarchical schedulers.



**NOTE:** Junos OS does not support ingress queuing and ingress hierarchical CoS on AE interfaces. You can, however, configure standard CoS classification and rewrite rules on AE interfaces.

To configure ingress CoS for per-unit schedulers, include the following statements at the [edit class-of-service interfaces *interface-name*] hierarchy level:

```
[edit class-of-service interfaces interface-name]
input-excess-bandwidth-share (proportional value | equal);
input-scheduler-map map-name;
input-shaping-rate rate;
input-traffic-control-profile profile-name;
unit logical-unit-number;
    input-scheduler-map map-name;
    input-shaping-rate (percent percentage | rate);
    input-traffic-control-profile profile-name;
```

To configure ingress CoS for hierarchical schedulers, include the interface-set *interface-set-name* statement at the [edit class-of-service interfaces] hierarchy level:

```
[edit class-of-service interfaces]
interface-set interface-set-name {
    input-traffic-control-profile profile-name;
    input-traffic-control-profile-remaining profile-name;
    interface interface-name {
```

```

input-excess-bandwidth-share (proportional value | equal);
input-traffic-control-profile profile-name;
input-traffic-control-profile-remaining profile-name;
unit logical-unit-number;
}
}

```

By default, ingress CoS features are disabled interfaces. To enable ingress CoS on an interface, configure the traffic-manager statement with ingress-and-egress mode as shown in the following example:

```

chassis {
  fpc 7 {
    pic 0 {
      traffic-manager {
        mode ingress-and-egress;
      }
    }
  }
}

```

Configured CoS features on the ingress are independent of CoS features on the egress.



**NOTE:** HQoS MPC cards installed in MX240, MX480, MX960, MX2008, MX2010, and MX2020 routers have a hardware limitation with an ingress queuing CoS "ingress-and-egress" configuration.

Ingress queuing can be enabled for a maximum of 10 ports per MIC Slot, resulting in 20 ports per MPC2E-3D-NG HQoS and MPC3E-3D-NG HQoS line card with 10 ports per MIC slot. In the XM chip there are 16 loopback streams allocated per port group for PG0 and PG1, where PG0 is mapped to MIC slot 0 and PG1 is mapped to MIC slot 1. On enabling ingress queuing on a PIC slot, one loopback stream from the XM chip is allocated per interface from the respective port group. Because there are only 16 loopback streams, out of which 2 are used by default and 4 are used for tunnel interfaces and inline services, 10 streams are left for ingress CoS.

## Change History Table



Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
17.4R2	Starting with Junos 17.4R2 on supported devices, you can have precise control over which ports have ingress CoS enabled by configuring traffic-manager at the port level.

## RELATED DOCUMENTATION

| [mode \(Layer 2 Tunneling Protocol Shaping\)](#)

## Per-Unit Scheduling and Hierarchical Scheduling for MPC Interfaces

### IN THIS SECTION

- [Scheduling Models Supported for MPC Interfaces | 57](#)
- [Scheduler Node Levels for MPC Interfaces | 58](#)

### Scheduling Models Supported for MPC Interfaces

Interfaces hosted on Modular Port Concentrator (MPC) line cards in MX Series 5G Universal Routing Platforms support the following models of class-of-service (CoS) scheduling, depending on MPC type:

#### Limited Scale Per-Unit Scheduling MPCs

Per-unit scheduling features on a limited scale are supported for interfaces hosted on some MPCs that do not have a dedicated queuing chip.

On MPCs that support per-unit scheduling, the following scheduling capabilities are available:

- Four or eight egress queues per unit.
- Delay buffer capacities of 100 ms by default, and up to 200 ms maximum delay.
- Rate shaping of the ports and their queues.

- Guaranteed rate enforced at the queues.

The per-unit scheduling features also support pre-classification of incoming packets to protect high priority packets in the event of congestion. Such features include ingress DSCP rewrite and per-VLAN classification, ingress and egress policing, and rewrites.

### **Hierarchical Scheduling MPCs**

MPCs that provide a dedicated queuing chip support hierarchical scheduling.

Hierarchical scheduling MPCs support all per-unit scheduling functionality plus fine-grained queuing abilities over four or five levels of hierarchical scheduling:

- Hierarchical scheduling with ports, interface sets, and logical interfaces.
- Shaping—Committed Information Rate (CIR) and a peak information rate (PIR)—at all scheduling levels, including queues.
- Three normal- priority levels and two excess- priority levels configurable at all scheduling levels, including queues.
- Per-priority shaping of traffic at Level 1 or Level 2.
- Shaping for unconfigured customer VLANs (C-VLANs) and for service VLANs (S-VLANs).

### **Scheduler Node Levels for MPC Interfaces**

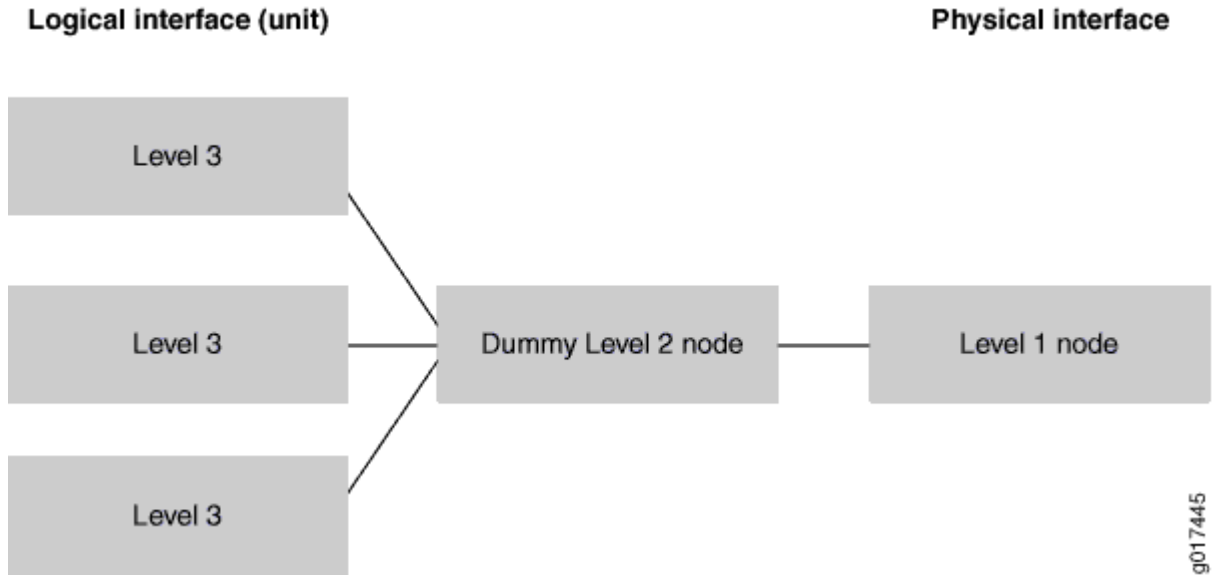
Interfaces hosted on MPCs support different scheduler node levels, depending on the MPC type:

#### **Scheduler Node Levels for Per-Unit Scheduling MPCs**

For an interface hosted on a per-unit scheduling MPC, each logical interface has its own dedicated level 3 node, and all logical interfaces share a common level 2 node (one per port).

[Figure 7 on page 59](#) illustrates scheduler node levels for an interface hosted on a per-unit queuing MPC.

Figure 7: Scheduler Node Levels for Per-Unit Scheduling MPCs



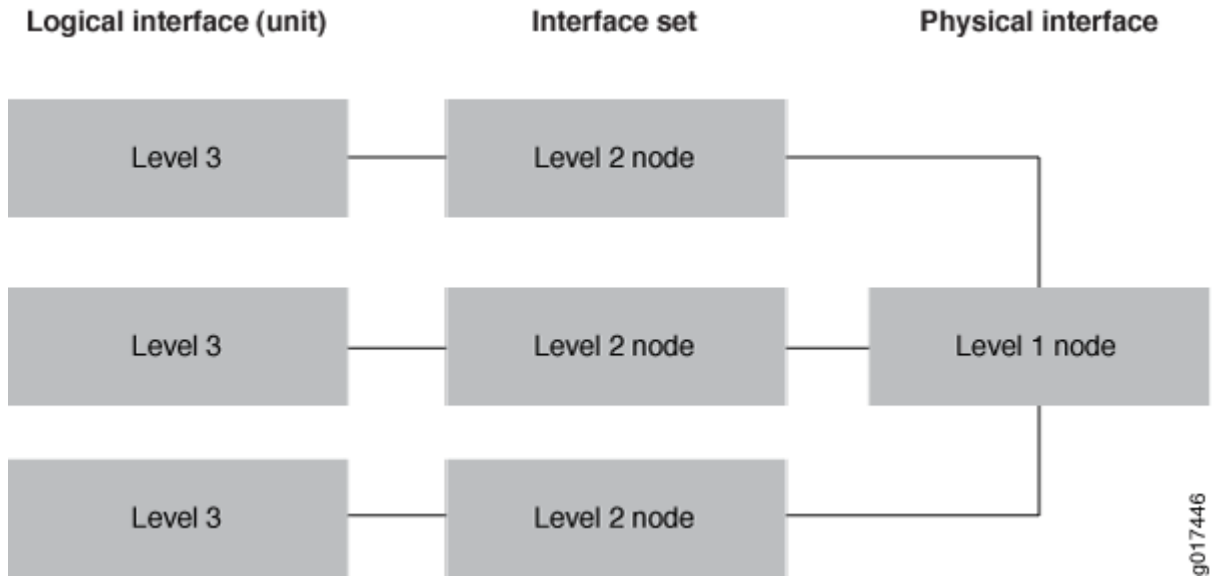
For interfaces hosted on per-unit scheduling MPCs, the level 2 node is always a dummy node.

### Scheduler Node Levels for Hierarchical Scheduling MPCs

The queuing model used by interfaces hosted on hierarchical scheduling MPCs supports up to five levels of scheduler nodes: the queue itself (level 5), session logical interface (ppp or dhcp) (level 4), customer VLAN (C-VLAN) (level 3), the interface set or service VLAN (S-VLAN) collection (level 2), and the physical interface or port (level 1).

[Figure 8 on page 60](#) illustrates the scheduler node levels for an interface hosted on a hierarchical scheduling MIC or MPC.

Figure 8: Scheduler Node Levels for Interfaces on Hierarchical Scheduling MPCs



The figure depicts scheduler nodes for an interface that does not include interface sets and for which traffic control profiles are applied to the logical interfaces only.

**i** **NOTE:** If an interface set has a CoS scheduling policy but none of its child logical interfaces has a CoS scheduling policy, then the interface set is considered to be a leaf node and has one level 2 and one level 3 node.

**RELATED DOCUMENTATION**

- [Understanding Hierarchical Scheduling for MIC and MPC Interfaces](#)
- [CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)
- [MX Series MPC Overview](#)
- [MPCs Supported by MX Series Routers](#)
- [MX Series MIC Overview](#)
- [MICs Supported by MX Series Routers](#)
- [MX5, MX10, MX40, and MX80 Modular Interface Card Description](#)

## Dedicated Queue Scaling for CoS Configurations on MIC and MPC Interfaces Overview

### IN THIS SECTION

- Queue Scaling for MPCs | 61
- Managing Remaining Queues | 63

Queueing Ethernet Modular Port Concentrators (MPCs) provide a set of dedicated queues for subscriber interfaces configured with hierarchical scheduling or per-unit scheduling.

The dedicated queues offered on these MPCs enable service providers to reduce costs through different scaling configurations. These queueing MPCs enable service providers to reduce the cost per subscriber by allowing many subscriber interfaces to be created with four or eight queues.

This topic describes the overall queue, scheduler node, and *logical interface* scaling for subscriber interfaces created on these MIC and MPC combinations.

### Queue Scaling for MPCs

Beginning with Junos OS Release 15.1, MPC2E-3D-NG-Q, MPC3E-3D-NG-Q, MPC5EQ-40G10G, and MPC5EQ-100G10G MPCs support up to five levels of hierarchical queuing. Beginning with Junos OS Release 16.1R1, MPC7 line cards also support five levels of hierarchical queuing. [Table 6 on page 61](#) lists the number of dedicated queues and nodes supported per MPC.

**Table 6: Dedicated Queues for MPCs**

MPC	Dedicated Queues	Level 4 Nodes	Level 3 Nodes	Level 2 Nodes	Level 1 Nodes (Ports)
MPC2E-3D-NG-Q	512,000	64,000	16,000	4000	384
MPC3E-3D-NG-Q					

**Table 6: Dedicated Queues for MPCs (Continued)**

MPC	Dedicated Queues	Level 4 Nodes	Level 3 Nodes	Level 2 Nodes	Level 1 Nodes (Ports)
MPC5EQ-40G10G	1 million	128,000	32,000	4000	384
MPC5EQ-100G10G					
MPC7	512,000	64,000	16,000	8,000	252
MPC10E on MX10K series platforms	256,000	32,000	8,000	4,000	128



**CAUTION:** The maximum scaling targets provided in [Table 6 on page 61](#) are based on system level design specifications. Actual realized subscriber or session scale is highly dependent upon the configuration and can be influenced by configuration variables including: the number of routes, the number of enabled services, the number of policy and firewall filters, policers, counters, statistics and access model type. Once you define a configuration, your Juniper account team can help characterize the expected system level scale or scale range for your live deployment.

MPCs vary in the number of Packet Forwarding Engines on board. MPC2E-3D-NG-Q and MPC3E-3D-NG-Q MPCs each have one Packet Forwarding Engine, allowing all 64,000 level 4 (subscriber) nodes to be allocated to a single MIC. MPC5EQ MPCs have two Packet Forwarding Engines, one for each possible MIC, each supporting 64,000 level 4 (subscriber) nodes. MPC7 MPCs also have two Packet Forwarding Engines, one for each possible MIC, each supporting 256,000 dedicated queues and 32,000 level 4 (subscriber) nodes.



**NOTE:** The nonqueuing MPCs MPC2E-3D-NG, MPC3E-3D-NG, MPC5E-40G10G, and MPC5E-100G10G provide up to eight queues per port in standard configuration. However, each of these MPCs can be configured to provide limited-scale hierarchical class of service (HCoS) and up to 32,000 queues.

## Managing Remaining Queues

In Junos OS releases earlier than Release 15.1R4, SNMP traps generate system log messages to notify you:

- When the number of available dedicated queues on the MPC drops below 10 percent. For example:

```
Mar 15 14:55:22.977 host cosd[1963]: COSD_OUT_OF_DEDICATED_QUEUES: Queue usage count for
interface xe-3/0/0 is at 90 percent
```

- When the maximum number of dedicated queues on the MPCs is reached. For example,

```
Mar 15 18:01:59.344 host cosd[3848]: COSD_OUT_OF_DEDICATED_QUEUES: Queue usage count for
interface xe-3/0/0 is at 100 percent.
```

When the maximum number of dedicated queues is allocated, the system does not provide subsequent subscriber interfaces with a dedicated set of queues. For per-unit scheduling configurations, there are no configurable queues remaining on the MPC.

For hierarchical scheduling configurations, remaining queues are available when the maximum number of dedicated queues is reached on the MPC. Traffic from these logical interfaces is considered unclassified and attached to a common set of queues that are shared by all subsequent logical interfaces. These common queues are the default port queues that are created for every port. You can configure a traffic-control profile and attach that to the interface to provide CoS parameters for the remaining queues. These subscriber interfaces remain with this traffic-control profile, even if dedicated queues become available.



**NOTE:** Starting in Junos OS Release 15.1R4, the `COSD_OUT_OF_DEDICATED_QUEUES` functionality is not available for QoS-enabled dynamic subscribers. Starting in Junos OS Release 17.4R1, CoS resource monitoring enables you to set a per-FPC queue threshold of up to 90 percent of resources bound to a scheduling hierarchy; subscriber logins are not allowed when the threshold is reached. However, this threshold applies to all queues, not dedicated queues alone. See [Resource Monitoring for Subscriber Management and Services Overview](#) for more information.

### Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
16.1R1	Beginning with Junos OS Release 16.1R1, MPC7 line cards also support five levels of hierarchical queuing.
15.1R1	Beginning with Junos OS Release 15.1, MPC2E-3D-NG-Q, MPC3E-3D-NG-Q, MPC5EQ-40G10G, and MPC5EQ-100G10G MPCs support up to five levels of hierarchical queuing.

## RELATED DOCUMENTATION

[Hierarchical Class of Service User Guide](#)

[Understanding Hierarchical Scheduling](#)

[Managing Dedicated and Remaining Queues for Static CoS Configurations on MIC and MPC Interfaces](#)

[Understanding Hierarchical Scheduling for MIC and MPC Interfaces](#)

## Jitter Reduction in Hierarchical CoS Queues

### IN THIS SECTION

- [Queue Jitter as a Function of the Maximum Number of Queues | 64](#)
- [Default Maximum Queues for Hierarchical Queuing MICs and MPCs | 65](#)
- [Shaping Rate Granularity as a Function of the Rate Wheel Update Period | 66](#)

### Queue Jitter as a Function of the Maximum Number of Queues

Each queuing chip on a Modular Interface Card (MIC) or Modular Port Concentrator (MPC) internally hosts a *rate wheel thread* that updates the *shaper credits* into the *shapers* available at each level of scheduling hierarchy. At each hierarchy level, the length of this update period determines two key characteristics of scheduling:

- The minimum buffer needed for the queue to pass packets without dropping.



- The degree of *jitter* encountered in the queue.

At each hierarchy level, the length of the rate wheel update period is dependent upon the number of entities enabled for that node level. Because traffic is queued at Level 5 (queues) and scheduled upwards to Level 1 (the port), the number of entities (queues) enabled at Level 5 determines the number of entities (logical interfaces, interface-sets, or ports) enabled at the other levels of the scheduling hierarchy. By extension, the number of queues enabled for a given scheduler node hierarchy determines the length of the update period at all hierarchy levels. Consequently, limiting the maximum number of queues supported by a hierarchical queuing MIC or MPC can reduce jitter in the queues. To configure the maximum number of queues allowed per hierarchical queuing MIC or MPC, include the `max-queues` statement at the `[edit chassis fpc slot-number]` hierarchy level.

## Default Maximum Queues for Hierarchical Queuing MICs and MPCs

The QX chip on a MIC or MPC consists of two symmetrical halves, and each half supports a maximum of 64 K queues (128 K queues per QX chip). The 2-port and 4-port 10-Gigabit Ethernet MICs with XFP and the MPC1\_Q line cards have one chipset and can support a maximum of 128 K queues, distributed across the two partitions of the single QX chip. The MPC2 Q and MPC2 EQ line cards have two chipsets and can support a maximum of 256 K queues, distributed across the four partitions of the two QX chips.

[Table 7 on page 65](#) lists the maximum number of queues supported by default and the corresponding rate wheel update period for each hierarchical queuing MIC or MPC.

**Table 7: Default Maximum Queues and Corresponding Rate Wheel Update Periods**

Router Model	Hierarchical Queuing MIC or MPC	Maximum Queues	Rate Wheel Update Period
MX5, MX10, MX40, and MX80 modular	<b>2-port 10-Gigabit Ethernet MIC with XFP</b> The chassis base board hosts one chipset-based Packet Forwarding Engine process that operates in standalone mode. The single QX chip is composed of two partitions that each support 64 K queues for egress ports.	128 K	1.6 ms
MX240, MX480, MX960, MX2010, and MX2020	<b>MPC1 Q</b> The MPC1 Q line card hosts one chipset-based Packet Forwarding Engine process that operates in fabric mode. The single QX chip is composed of two partitions that each support 64 K queues for egress ports.	128 K	1.6 ms

**Table 7: Default Maximum Queues and Corresponding Rate Wheel Update Periods (Continued)**

Router Model	Hierarchical Queuing MIC or MPC	Maximum Queues	Rate Wheel Update Period
	<b>MPC2 Q</b> The MPC2 Q line card hosts two chipset-based Packet Forwarding Engine processes that operate in fabric mode. The two QX chips are composed of four partitions that each support 64 K queues for egress ports.	256 K	1.6 ms
	<b>MPC2 EQ</b> The MPC2 EQ line card hosts two chipset-based Packet Forwarding Engine processes that operate in fabric mode. The two QX chips are composed of four partitions that each support 64 K queues for egress ports.	256 K	2.6 ms

You can configure hierarchical queuing MICs and MPCs to support a reduced maximum number of queues. Doing so reduces the rate wheel update period used by the QX chip, which in turn reduces jitter in the queues for the egress interfaces hosted on the line card.

### Shaping Rate Granularity as a Function of the Rate Wheel Update Period

Reducing the length of the QX chip rate wheel update period, in addition to reducing jitter in the hierarchical scheduling queues, also indirectly increases the shaping granularity.

For a given port line rate and scheduling hierarchy level, the shaping granularity is a function of the minimum shaper credit size and the rate wheel update period in effect as a result of the number of queues supported by the line card.

$$\text{shaping granularity} = \text{minimum shaper credit size} / \text{rate wheel update period}$$

[Table 8 on page 67](#) shows how shaping granularity is calculated for non-enhanced hierarchical queuing MIC and MPC line cards with default values for minimum shaper credit size and for rate wheel update period.

**Table 8: Default Shaping Granularities on Non-Enhanced Queuing MICs and MPCs**

Port Type	Hierarchy Level	Non-Enhanced Queuing MIC or MPC Defaults		Calculation of Shaping Granularity
		Minimum Credit	Update Period	
1 Gbps Queuing	Level 1 (port), Level 4 (queues)	4 bytes = 32 bits	13.33 ms = 0.01333 sec	32 bits / 0.01333 sec = <b>2.4 Kbps</b>
	Level 2, Level 3	16 bytes = 128 bits	1.66 ms = 0.00166 sec	128 bits / 0.01333 sec = <b>9.6 Kbps</b>
10 Gbps Queuing	Level 1 (port), Level 4 (queues)	16 bytes = 128 bits	13.33 ms = 0.01333 sec	128 bits / 0.01333 sec = <b>9.6 Kbps</b>
	Level 2, Level 3	64 bytes = 512 bits	1.66 ms = 0.00166 sec	512 bits / 0.01333 sec = <b>38.4 Kbps</b>

**RELATED DOCUMENTATION**

*Example: Reducing Jitter in Hierarchical CoS Queues*

*Per-Unit Queuing and Hierarchical Queuing for MIC and MPC Interfaces*

*Understanding Hierarchical Scheduling for MIC and MPC Interfaces*

**Example: Reducing Jitter in Hierarchical CoS Queues****IN THIS SECTION**

- Requirements | 68
- Overview | 68
- Configuration | 68

This example shows how to reduce jitter in the output queues for VLAN ports hosted on a hierarchical queuing MPC.

## Requirements

This example uses the following Juniper Networks hardware and Junos OS software:

- MX960 router in an IPv4 network and running Junos OS Release 13.2 or later.
- Available Gigabit Ethernet port hosted on FPC slot 2, PIC slot 0, port 0.
- Available Gigabit Ethernet port hosted on port 0 of a Gigabit Ethernet Modular Interface Card (MIC) in PIC slot 0 of an MPC2 Q Modular Port Concentrator (MPC) in FPC slot 5.

Before you begin configuring this example, make sure that the maximum number of queues allowed for the hierarchical queuing MPC in slot 5 has not yet been configured. When you enter the `show chassis fpc 5` command from configuration mode, the `max-queues` statement should not display.

## Overview

In this example you configure hierarchical scheduling on a VLAN port hosted on a hierarchical queuing MPC. To reduce jitter in the queues for all egress ports hosted on the MPC, reduce the maximum number of queues allowed for MPC.

## Configuration

### IN THIS SECTION

- [Verification | 74](#)

## CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

```
set interfaces xe-2/0/0 per-unit-scheduler
set interfaces xe-2/0/0 flexible-vlan-tagging
set interfaces xe-2/0/0 unit 0 vlan-id 1
set interfaces xe-2/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces xe-2/0/0 unit * classifiers ieee-802.1 ieee_jitter
```

```

set interfaces xe-5/0/0 per-unit-scheduler
set interfaces xe-5/0/0 flexible-vlan-tagging
set interfaces xe-5/0/0 unit 0 vlan-id 1
set interfaces xe-5/0/0 unit 0 family inet address 10.2.1.1/24
set class-of-service-interfaces xe-5/0/0 unit * output-traffic-control-profile tcp
set class-of-service forwarding-classes queue 0 be
set class-of-service forwarding-classes queue 1 ef
set class-of-service forwarding-classes queue 2 af
set class-of-service forwarding-classes queue 3 nc
set class-of-service schedulers be_sch priority low
set class-of-service schedulers ef_sch priority low
set class-of-service schedulers af_sch priority strict-high
set class-of-service schedulers nc_sch priority low
set class-of-service classifiers ieee_jitter forwarding-class be loss-priority low code-points
000
set class-of-service classifiers ieee_jitter forwarding-class ef loss-priority low code-points
001
set class-of-service classifiers ieee_jitter forwarding-class af loss-priority low code-points
010
set class-of-service classifiers ieee_jitter forwarding-class nc loss-priority low code-points
011
set class-of-service scheduler-maps smap_jitter forwarding-class be scheduler be_sch
set class-of-service scheduler-maps smap_jitter forwarding-class ef scheduler ef_sch
set class-of-service scheduler-maps smap_jitter forwarding-class af scheduler af_sch
set class-of-service scheduler-maps smap_jitter forwarding-class nc scheduler nc_sch
set class-of-service traffic-control-profiles tcp scheduler-map smap_jitter
set class-of-service traffic-control-profiles tcp shaping-rate 6g

```

## Baseline Configuration

### Step-by-Step Procedure

Configure hierarchical scheduling at xe-5.0.0.

1. To configure the VLAN 1 input and output at xe-2/0/0.0 and xe-5/0/0.0:

```

[edit]
user@host# set interfaces xe-2/0/0 per-unit-scheduler
user@host# set interfaces xe-2/0/0 flexible-vlan-tagging
user@host# set interfaces xe-2/0/0 unit 0 vlan-id 1
user@host# set interfaces xe-2/0/0 unit 0 family inet address 10.1.1.1/24

```

```

user@host# set interfaces xe-5/0/0 per-unit-scheduler
user@host# set interfaces xe-5/0/0 flexible-vlan-tagging
user@host# set interfaces xe-5/0/0 unit 0 vlan-id 1
user@host# set interfaces xe-5/0/0 unit 0 family inet address 10.2.1.1/24

```

2. Map each of four queues to a forwarding class.

```

[edit]
user@host# set class-of-service forwarding-classes queue 0 be
user@host# set class-of-service forwarding-classes queue 1 ef
user@host# set class-of-service forwarding-classes queue 2 af
user@host# set class-of-service forwarding-classes queue 3 nc

```

3. Assign a packet-scheduling priority value to each forwarding class.

```

[edit]
user@host# set class-of-service schedulers be_sch priority low
user@host# set class-of-service schedulers ef_sch priority low
user@host# set class-of-service schedulers af_sch priority strict-high
user@host# set class-of-service schedulers nc_sch priority low

```

4. Customize the default IEEE 802.1p classifier (BA classifier based on Layer 2 header) by defining different values for IEEE 802.1p code points.

```

[edit]
user@host# set class-of-service classifiers ieee_jitter forwarding-class be loss-priority low
code-points 000
user@host# set class-of-service classifiers ieee_jitter forwarding-class ef loss-priority low
code-points 001
user@host# set class-of-service classifiers ieee_jitter forwarding-class af loss-priority low
code-points 010
user@host# set class-of-service classifiers ieee_jitter forwarding-class nc loss-priority low
code-points 011

```

5. Apply the BA classifier to the input of the logical units on xe-2/0/0.

```
[edit]
user@host# set interfaces xe-2/0/0 unit * classifiers ieee-802.1 ieee_jitter
```

6. Configure the scheduler map smap\_jitter to map the forwarding classes to the schedulers.

```
[edit]
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class be scheduler
be_sch
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class ef scheduler
ef_sch
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class af scheduler
af_sch
user@host# set class-of-service scheduler-maps smap_jitter forwarding-class nc scheduler
nc_sch
```

7. Configure the traffic control profile tcp to combine the scheduler map smap\_jitter (that maps the forwarding classes to the schedulers for port-based scheduling) with a shaping rate (for hierarchical scheduling).

```
[edit]
user@host# set class-of-service traffic-control-profiles tcp scheduler-map smap_jitter
user@host# set class-of-service traffic-control-profiles tcp shaping-rate 6g
```

8. Apply the traffic control profile to the router output at xe-5/0/0.

```
[edit]
user@host# set class-of-service-interfaces xe-5/0/0 unit * output-traffic-control-profile tcp
```

9. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Results

Confirm your configuration by entering `show interfaces` and `show class-of-service` commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
xe-2/0/0 {
  per-unit-scheduler;
  flexible-vlan-tagging;
  unit 0 {
    vlan-id 1;
    family inet {
      address 10.1.1.1/24;
    }
  }
}
xe-5/0/0 {
  per-unit-scheduler;
  flexible-vlan-tagging;
  unit 0 {
    vlan-id 1;
    family inet {
      address 10.2.1.1/24;
    }
  }
}
```

```
[edit]
user@host# show class-of-service
classifiers {
  ieee-802.1 ieee_jitter {
    forwarding-class be {
      loss-priority low code-points 000;
    }
    forwarding-class ef {
      loss-priority low code-points 001;
    }
    forwarding-class af {
      loss-priority low code-points 010;
    }
  }
}
```



```
    }
    forwarding-class nc {
        loss-priority low code-points 011;
    }
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
traffic-control-profiles {
    tcp {
        scheduler-map smap_jitter;
        shaping-rate 6g;
    }
}
interfaces {
    xe-2/0/0 {
        unit * {
            classifiers {
                ieee-802.1 ieee_jitter;
            }
        }
    }
    xe-5/0/0 {
        unit * {
            output-traffic-control-profile tcp;
        }
    }
}
scheduler-maps {
    smap_jitter {
        forwarding-class be scheduler be_sch;
        forwarding-class ef scheduler ef_sch;
        forwarding-class af scheduler af_sch;
        forwarding-class nc scheduler nc_sch;
    }
}
schedulers {
    be_sch {
        priority low;
    }
}
```

```
}
ef_sch {
  priority low;
}
af_sch {
  priority strict-high;
}
nc_sch {
  priority low;
}
}
```

## Verification

### IN THIS SECTION

- [Measuring End-to-End Jitter to Establish the Baseline | 74](#)
- [Configuring Jitter Reduction | 75](#)
- [Measuring End-to-End Jitter to Verify Jitter Reduction | 75](#)

Confirm that the configuration is working properly

### *Measuring End-to-End Jitter to Establish the Baseline*

#### Purpose

Establish a baseline measurement by noting the amount of jitter that occurs when the hierarchical queuing line card hosting the egress port is configured with the default maximum number of queues.

#### Action

To measure jitter:

1. Pass traffic through the VLAN.
2. Measure the variation in packet delay for selected packets in the data flow.

## Configuring Jitter Reduction

### Purpose

Reduce jitter in the VLAN port output queues.

### Action

1. Configure a reduced maximum number of queues for egress ports on the hierarchical queuing MPC in slot 5, thereby reducing the jitter in the port queues.

```
[edit]  
user@host# set chassis fpc 5 max-queue 64k
```

2. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

## Measuring End-to-End Jitter to Verify Jitter Reduction

### Purpose

Measure the amount of jitter that occurs when the hierarchical queuing line card hosting the egress port is configured with a reduced maximum number of queues.

### Action

To measure jitter:

1. Pass traffic through the VLAN.
2. Measure the variation in packet delay for selected packets in the data flow.

## RELATED DOCUMENTATION

*Jitter Reduction in Hierarchical CoS Queues*  
*max-queues*

## Hierarchical Schedulers on Aggregated Ethernet Interfaces Overview

On MX Series routers, you can apply hierarchical schedulers on aggregated ethernet bundles using interface sets. This feature enables you to configure a group of virtual LANs (VLANs) and control their bandwidth. This feature is supported at egress only.

You can configure interface sets for aggregated Ethernet (AE) interfaces created under static configurations. You can configure class-of-service parameters on AE interfaces, in either link-protect or non-link-protect mode. You can configure these parameters at the AE physical interface level. The CoS configuration is fully replicated for all AE member links in link-protect mode. You can control the way these parameters are applied to member links in non-link-protect mode by configuring the AE interface to operate in scaled mode or replicate mode.

The link membership list and scheduler mode of the interface set is inherited from the underlying aggregated Ethernet interface over which the interface set is configured. When an aggregated Ethernet interface operates in link protection mode, or if scheduler mode is configured to replicate member links, the scheduling parameters of the interface set are copied to each of the member links.

If the scheduler mode of the aggregated Ethernet interface is set to scale member links, the scheduling parameters are scaled based on the number of active member links (scaling factor is  $1/A$  where A is the number of active links in the bundle) and applied to each of the AE interface member links.

To configure an interface set, include the `interface-set` statement at the `[edit class-of-service interfaces]` hierarchy level.

To apply scheduling and queuing parameters to the interface set, include the `output-traffic-control-profile profile-name` statement at the `[edit class-of-service interfaces interface-name interface-set interface-set-name]` hierarchy level.

To apply an output traffic scheduling and shaping profile for the remaining traffic to the *logical interface* or interface set, include the `output-traffic-control-profile-remaining profile-name` statement at the `[edit class-of-service interfaces interface-name]` hierarchy level or the `[edit class-of-service interfaces interface-name interface-set interface-set-name]` hierarchy level.

### RELATED DOCUMENTATION

[\*Configuring Hierarchical Schedulers on Aggregated Ethernet Interfaces\*](#)

[\*output-traffic-control-profile-remaining\*](#)

[\*Controlling Remaining Traffic\*](#)

## Configuring Hierarchical Schedulers on Aggregated Ethernet Interfaces

The following example shows the creation of an interface set for aggregated Ethernet interfaces in a static Ethernet configuration.

To configure interface sets for aggregated Ethernet (AE) interfaces created under static configurations:

1. Create the AE interfaces.

```
[edit]
user@host# show chassis | display set
set chassis aggregated-devices ethernet device-count 10
```

2. Configure the AE physical interfaces and member links.

```
user@host# show interfaces | display set

set interfaces ge-5/2/0 gigether-options 802.3ad ae0
set interfaces ge-5/2/1 gigether-options 802.3ad ae0
set interfaces ae0 hierarchical-scheduler maximum-hierarchy-levels 2
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 unit 0 vlan-id 100
set interfaces ae0 unit 1 vlan-id 101
set interfaces ae0 unit 2 vlan-id 102
set interfaces ae0 unit 3 vlan-id 103
set interfaces ae0 unit 4 vlan-id 104
```

3. Configure the interface set.

```
set interfaces interface-set ifset1-ae0 interface ae0 unit 0
set interfaces interface-set ifset1-ae0 interface ae0 unit 1
```

4. Configure class-of-service parameters for the interface sets.

```
set class-of-service interfaces interface-set ifset1-ae0 output-traffic-control-profile tcp
```



**NOTE:** You also need to configure the parameters of the traffic control profile. For more information, see the Related Documentation section on this page.

## 5. Configure scheduler mode.

```
set class-of-service interfaces ae0 member-link-scheduler scale
```

### RELATED DOCUMENTATION

*[Configuring Traffic Control Profiles for Shared Scheduling and Shaping](#)*

*[Hierarchical Schedulers on Aggregated Ethernet Interfaces Overview](#)*

*[Example: Configuring Shared Resources on Ethernet IQ2 Interfaces](#)*

*[Configuring Traffic Control Profiles for Shared Scheduling and Shaping](#)*

## Example: Configuring Scheduling Modes on Aggregated Interfaces

You can configure class-of-service parameters, such as queuing or shaping parameters on aggregated interfaces, in either link-protect or non-link-protect mode. You can configure these parameters for per-unit schedulers, hierarchical schedulers, or shaping at the physical and logical interface level. You can control the way these parameters are applied by configuring the aggregated interface to operate in scale or replicate mode.

You can apply these parameters on the following routers:

- MX Series router interfaces on EQ DPCs
- MX Series router interfaces on MICs or MPCs through Junos OS Release 10.2 (non-link-protect mode only)
- M120 or M320 routers
- T Series router interfaces on IQ2 PICs
- PTX Series Packet Transport Routers

You can configure the applied parameters for aggregated interfaces operating in non-link-protected mode. In link-protected mode, only one link in the bundle is active at a time (the other link is a backup link) so schedulers cannot be scaled or replicated. In non-link-protected mode, all the links in the bundle are active and send traffic; however, there is no backup link. If a link fails or is added to the bundle in non-link-protected mode, the links' traffic is redistributed among the active links.

To set the scheduling mode for aggregated interfaces, include the `scale` or `replicate` option of the `member-link-scheduler` statement at the `[edit class-of-service interfaces ae $n$ ]` hierarchy level, where  $n$  is the configured number of the interface:

```
[edit class-of-service interfaces ae $n$ ]
member-link-scheduler (replicate | scale);
```

By default, if you do not include the `member-link-scheduler` statement, scheduler parameters are applied to the member links in the `scale` mode (also called “equal division mode”).

The aggregated Ethernet interfaces are otherwise configured as usual. For more information on configuring aggregated Ethernet interfaces, see the [Junos OS Network Interfaces Library for Routing Devices](#).

The following examples set `scale` mode on the `ae0` interface and `replicate` mode on the `ae1` interface.

```
[edit class-of-service]
interfaces ae0 {
    member-link-scheduler scale;
}

[edit class-of-service]
interfaces ae1 {
    member-link-scheduler replicate;
}
```



**NOTE:** The `member-link-scheduler` statement only appears for aggregated interfaces. You configure this statement for aggregated interfaces in non-link-protected mode. For more information about link protection modes, see the *Network Interfaces Configuration Guide*.

Aggregated interfaces support both hierarchical and per-unit schedulers.



**NOTE:** The `traffic-control-profiles` statement is not supported for PTX Series Packet Transport Routers.

When interface parameters are using the `scale` option of the `member-link-scheduler` statement, the following parameters under the `[edit class-of-service traffic-control-profiles traffic-control-profile-name]` configuration are scaled on egress when hierarchical schedulers are configured:

- `shaping-rate` (PIR)
- `guaranteed-rate` (CIR)
- `delay-buffer-rate`

When interface parameters are using the `scale` option of the `member-link-scheduler` statement, the following parameters under the `[edit class-of-service schedulers scheduler-name]` configuration are scaled on egress when per-unit schedulers are configured:

- `transmit-rate`
- `buffer-size`



**NOTE:** You cannot apply a hierarchical scheduler at the interface set level for an `ae` interface. (Interface sets cannot be configured under an `ae` interface.)

The following configuration parameters are not supported on `ae` interfaces in non-link-protection mode:

- Input scheduler maps
- Input traffic control profiles
- Input shaping rates

The following configuration conventions are also not supported:

- Scaling of the `input-traffic-control-profile-remaining` statement.
- The `scheduler-map-chassis` statement and the `derived` option for the `ae` interface. Chassis scheduler maps should be applied under the physical interfaces.
- Dynamic and demux interfaces are not supported as part of the `ae` bundle.

Depending on whether the `scale` or `replicate` option is configured, the `member-link-scheduler` statement operates in either scaled mode (also called “equal division mode”) or replicated mode, respectively.

In scaled mode, a VLAN can have multiple flows that can be sent over multiple member links of the `ae` interface. Likewise, a member link can receive traffic from any VLAN in the `ae` bundle. In scaled mode, the physical interface bandwidth is divided equally among all member links of the `ae` bundle.

In scaled mode, the following scheduler parameter values are divided equally among the member links:

- When the parameters are configured using traffic control profiles, then the parameters scaled are the shaping rate, guaranteed rate, and delay buffer rate.



- When the parameters are configured using scheduler maps, then the parameters scaled are the transmit rate and buffer size. Shaping rate is also scaled if you configure it in bits per second (bps). Shaping rate is not scaled if you configure it as a percentage of the available interface bandwidth.

For example, consider an ae bundle between routers R1 and R2 consisting of three links. These are ge-0/0/1, ge-0/0/2 and ge-0/0/3 (ae0) on R1; and ge-1/0/0, ge-1/0/1, and ge-1/0/2 (ae2) on R2. Two logical interfaces (units) are also configured on the ae0 bundle on R1: ae0.0 and ae0.1.

On ae0, traffic control profiles on R1 are configured as follows:

- ae0 (the physical interface level) has a PIR of 450 Mbps.
- ae0.0 (VLAN 100 at the logical interface level) has a PIR of 150 Mbps and a CIR of 90 Mbps.
- ae0.1 (VLAN 200 at the logical interface level) has a PIR of 90 Mbps and a CIR of 60 Mbps.

In scaled mode, the ae0 PIR is first divided among the member physical interfaces. Because there are three members, each receives  $450 / 3 = 150$  Mbps as a derived value. So the scaled PIR for the members interfaces is 150 Mbps each.

However, there are also two logical interfaces (ae0.0 and ae0.1) and VLANs (100 and 200) on ae0. Traffic can leave on any of the three physical interfaces (ge-0/0/1, ge-0/0/2, or ge-0/0/3) in the bundle. Therefore, two derived logical interfaces are added to the member links to represent the two VLANs.

There are now six logical interfaces on the physical interfaces of the links making up the ae bundle, one set for VLAN 100 and the other for VLAN 200:

- ge-0/0/1.0 and ge-0/0/1.1
- ge-0/0/2.0 and ge-0/0/2.1
- ge-0/0/3.0 and ge-0/0/3.1

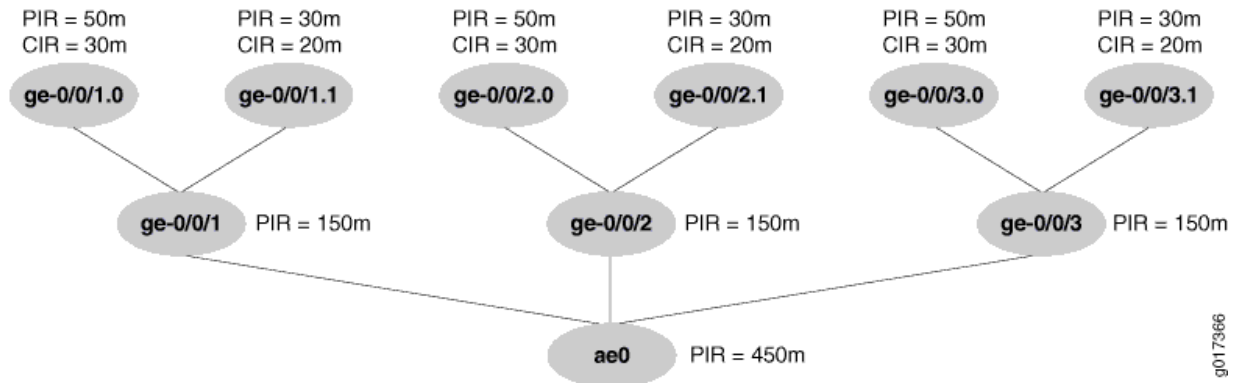
The traffic control profile parameters configured on ae0.0 are divided across all the underlying logical interfaces (the unit 0s). In the same way, the traffic control profile parameters configured on ae0.1 are divided across all the underlying logical interfaces (the unit 1s).

Therefore, the derived values of the scaled parameters on the interfaces are:

- For ge-0/0/1.0 and ge-0/0/2.0 and ge-0/0/3.0, each CIR =  $90 / 3 = 30$  Mbps, and each PIR =  $150 / 3 = 50$  Mbps.
- For ge-0/0/1.1 and ge-0/0/2.1 and ge-0/0/3.1, each CIR =  $60 / 3 = 20$  Mbps, and each PIR =  $90 / 3 = 30$  Mbps.

The scaled values are shown in [Figure 9 on page 82](#).

Figure 9: Scaled Mode for Aggregated Ethernet Interfaces



In scaled mode, when a new member link is added to the bundle, or an existing member link is either removed or fails, then the scaling factor (based on the number of active links) is recomputed and the new scheduler or traffic control profile parameters are reassigned. Only the PIR, CIR, and buffer parameters are recomputed: all other parameters are simply copied at each level.



**NOTE:** In `show class-of-service scheduler-map` commands, values derived in scaled mode instead of explicitly configured are flagged with `&sf*n` suffix, where `n` indicates the value of the scaling factor.

The following sample shows the output for the scheduler map named `smap-all-abs` with and without a scaling factor:

```
user@host> show class-of-service scheduler-map
Scheduler map: smap-all-abs, Index: 65452

Scheduler: q0_sch_abs, Forwarding class: be, Index: 6775
Transmit rate: 40000000 bps, Rate Limit: none, Buffer size: remainder,
Priority: low
  Excess Priority: unspecified
Drop profiles:
  Loss priority  Protocol  Index  Name
  Low           any       1      <default-drop-profile>
  Medium low    any       1      <default-drop-profile>
  Medium high   any       1      <default-drop-profile>
  High          any       1      <default-drop-profile>

user@host> show class-of-service scheduler-map
Scheduler map: smap-all-abs, Index: 65452
```

```
Scheduler: q0_sch_abs&***sf**3, Forwarding class: be, Index: 2128
Transmit rate: 13333333 bps, Rate Limit: none, Buffer size: remainder,
Priority: low
  Excess Priority: unspecified
  Drop profiles:
    Loss priority  Protocol  Index  Name
    Low           any      1      <default-drop-profile>
    Medium low    any      1      <default-drop-profile>
    Medium high   any      1      <default-drop-profile>
    High          any      1      <default-drop-profile>
```



**NOTE:** There can be multiple scheduler maps created with different scaling factors, depending on when the child interfaces come up. For example, if there are only two active children on a parent interface, a new scheduler map with a scaling factor of 2 is created. The scheduler map name is `smap-all-abs&***sf**2`.

In replicated mode, in contrast to scaled mode, the configured scheduler parameters are simply replicated, not divided, among all member links of the ae bundle.

In replicated mode, the following scheduler parameter values are replicated among the member links and logical interfaces:

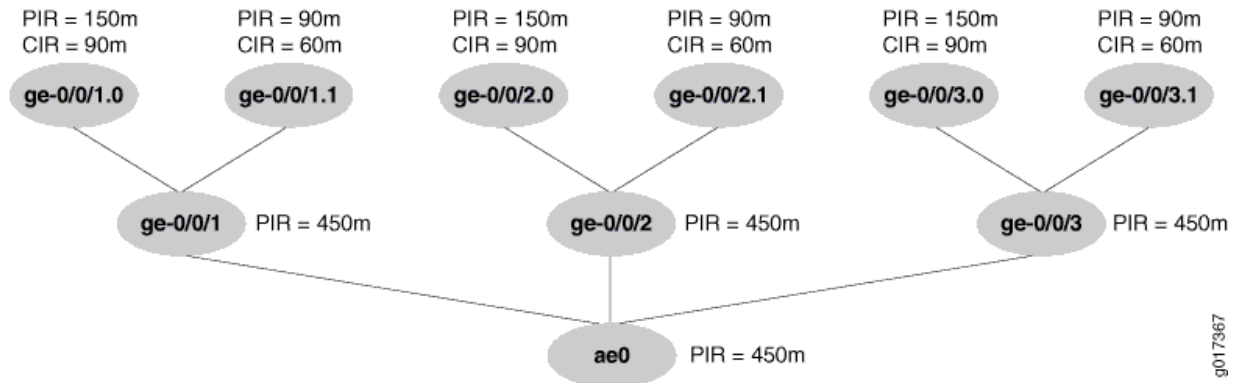
- When the parameters are configured using traffic control profiles, then the parameters replicated are the shaping rate, guaranteed rate, and delay buffer rate.
- When the parameters are configured using scheduler maps, then the parameters replicated are the transmit rate and buffer size.

If the scheduler parameters in the example configuration between routers R1 and R2 are applied with the `member-link-scheduler replicate` statement and option, the following parameters are applied:

- The `ae0` PIR is copied among the member physical interfaces. Each receives 450 Mbps as a PIR.
- For each logical interface unit `.0`, the configured PIR and CIR for `ae0.0` is replicated (copied). Each logical interface unit `.0` receives a PIR of 150 Mbps and a CIR of 90 Mbps.
- For each logical interface unit `.1`, the configured PIR and CIR for `ae0.1` is replicated (copied). Each logical interface unit `.1` receives a PIR of 90 Mbps and a CIR of 60 Mbps.

The replicated values are shown in [Figure 10 on page 84](#).

Figure 10: Replicated Mode for Aggregated Ethernet Interfaces



In replicated mode, when a new member link is added to the bundle, or an existing member link is either removed or fails, the values are either copied or deleted from the required levels.

## RELATED DOCUMENTATION

*How Schedulers Define Output Queue Properties*

*Default Schedulers Overview*

*Configuring Schedulers*

## Increasing Available Bandwidth on Rich-Queuing MPCs by Bypassing the Queuing Chip

Queuing MPCs contain a queuing chip that enables rich-queuing features such as hierarchical and per-vlan queuing. By default, all traffic passing through an interface on one of these MPCs also passes through the queuing chip, which decreases the available bandwidth of the interface. If you do not require hierarchical or per-vlan queuing on a particular interface of a queuing MPC, you can bypass the queuing chip to increase the available bandwidth.

To bypass the queuing chip on a queuing MPC, you must be running Junos OS Release 14.2 or later.

Starting with Junos OS 18.2R1, you can enable this option on vMX routers to save a vCPU when scheduling is not needed on an interface.

To bypass the queuing chip on an interface on a queuing MPC:

1. Ensure that neither `per-unit-scheduler` nor `hierarchical-scheduler` is configured on the interface.



**NOTE:** It is not possible to bypass the queuing chip on an interface if per-unit or hierarchical scheduling is configured on that interface.

2. Ensure that *flexi-queuing-mode* is enabled.
3. Enable `bypass-queuing-chip` on the interface.

For example:

```
[edit interfaces]
user@router# set interface-name bypass-queuing-chip
```

4. Commit your changes.

```
[edit interfaces]
user@router# show
interface-name {
    bypass-queuing-chip;
}
```

5. Verify your changes.

```
user@router> show interfaces interface-name
Physical interface: interface-name, Enabled, Physical link is Up
  Interface index: 147, SNMP ifIndex: 524
  Link-level type: Ethernet, MTU: 1514, MRU: 1522, LAN-PHY mode, Speed: 1000mbps,
  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online
  Pad to minimum frame size: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 4 maximum usable queues
  Schedulers     : 0, Queuing Chip Bypassed
  Current address: 00:21:59:0f:35:31, Hardware address: 00:21:59:0f:35:31
  Last flapped   : 2014-04-29 14:10:18 PDT (02:27:46 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  Interface transmit statistics: Disabled
```

### Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
18.2R1	Starting with Junos OS 18.2R1, you can enable this option on vMX routers to save a vCPU when scheduling is not needed on an interface.
14.2	To bypass the queuing chip on a queuing MPC, you must be running Junos OS Release 14.2 or later.

### RELATED DOCUMENTATION

| [\*bypass-queuing-chip\*](#)

# 2

PART

## Hierarchical CoS for Subscriber Management

---

[Hierarchical Class of Service for Subscriber Management | 88](#)

[Applying CoS to Groups of Subscriber Interfaces | 121](#)

[Configuring Hierarchical Scheduling for MPLS Pseudowire Interfaces | 153](#)

[Configuring Hierarchical Scheduling for L2TP | 171](#)

[Preventing Bandwidth Contention on Subscriber Interfaces | 182](#)

[Configuring Targeted Distribution of Subscribers on Aggregated Ethernet Interfaces | 208](#)

[Applying CoS Using Parameters Received from RADIUS | 235](#)

---

# Hierarchical Class of Service for Subscriber Management

## IN THIS CHAPTER

- [Hierarchical Class of Service for Subscriber Management Overview | 88](#)
- [Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)
- [Hardware Requirements for Dynamic Hierarchical CoS | 98](#)
- [Configuring Static Hierarchical Scheduling in a Dynamic Profile | 100](#)
- [Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links | 101](#)
- [Configuring Hierarchical CoS on a Static PPPoE Subscriber Interface | 103](#)
- [Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access | 104](#)

## Hierarchical Class of Service for Subscriber Management Overview

The hierarchical class-of-service (HCoS) architecture as supported on fine-grained queuing MPCs is a powerful feature designed to provide a flexible and scalable CoS solution in broadband network gateway (BNG) subscriber access applications where triple-play or business class offerings are enabled through IP CoS.

Hierarchical CoS enables you to apply traffic scheduling and queuing parameters (which can include a delay-buffer bandwidth) and packet transmission scheduling parameters (which can include buffer management parameters) to an individual subscriber interface rather than to all interfaces configured on the port. HCoS enables you to dynamically modify queues when subscribers require services.

The logical interface set construct in a five-level scheduler hierarchy is the key feature that enables HCoS. The interface set feature allows you to group subscribers into aggregate classes with specific guaranteed and peak rates that map to service classes. Service classes ultimately map to how much you can charge for the differentiated service levels.

HCoS can be applied dynamically through the use of dynamic traffic profiles and RADIUS vendor-specific attributes (VSAs).



Dynamic traffic profiles are used to dynamically apply CoS to individual subscribers or groups of subscribers. This enables you, as a service provider, to deploy a BRAS solution without having to manually provision each customer. In a dynamic traffic profile, variables are used to represent the values for things like shaping rate and drop priority.

Dynamic traffic profiles are used in conjunction with dynamic profiles. Dynamic profiles allow you to dynamically provision IP service definitions by creating a template configuration and having the specific variable values assigned in real time when the subscriber authenticates to the network.



**NOTE:** For a complete list of the Junos OS system variables, see: [Junos OS Predefined Variables](#).

To learn more about how to use HCoS in conjunction with dynamic traffic control profiles for subscriber management, read the [Day One: Dynamic Subscriber Management](#) book. Note that you will need to have a login and password to access this document.

In addition, the following topics are very helpful:

- ["Hierarchical Class of Service Overview" on page 2](#)
- *Subscriber Access Network Overview*
- *CoS for Subscriber Access Overview*
- *Subscriber Management Overview*
- *Class of Service and Subscriber Management Overview*

Before applying dynamic HCoS on your network, you should learn about HCoS, define your needs, plan how you want to implement HCoS, and test the operation in a simulated environment.

## RELATED DOCUMENTATION

[Hierarchical Class of Service Overview | 2](#)

[Hierarchical Class of Service Network Scenarios | 6](#)

## Understanding Hierarchical CoS for Subscriber Interfaces

### IN THIS SECTION

- [Two-Level Hierarchical Scheduling | 91](#)
- [Three-Level Hierarchical Scheduling | 92](#)
- [Four-Level Hierarchical Scheduling | 96](#)

Hierarchical CoS enables you to apply traffic scheduling and queuing parameters and packet transmission scheduling parameters to an individual subscriber interface rather than to all interfaces configured on a port. Hierarchical CoS enables you to dynamically modify queues when subscribers require services.

Hierarchical CoS is supported on MX Series routers with either Enhanced Queuing DPCs or queuing MPCs/MICs installed. Beginning with Junos OS Release 16.1, five levels of hierarchy are supported on MPC5E 3D Q line cards.

Interfaces support up to a five-level CoS scheduling hierarchy that, when fully configured, generally consists of the physical interface (level 1), an interface set or underlying interface (level 2), one or more underlying logical interfaces (level 3), one or more session or customer VLANs (level 4), and one or more queues (level 5). Although all CoS scheduling hierarchies are five-level, level 1 is always the physical interface and level 5 is always the queue. Hierarchical scheduling configurations consist of the type of interfaces you configure—for example, a *logical interface* or an interface set—and where those interfaces reside in the scheduling hierarchy—level 2, level 3, or level 4. Because many hierarchical scheduling configurations are possible, we use the terms *two-level hierarchical scheduling*, *three-level hierarchical scheduling*, *four-level hierarchical scheduling* in this topic.

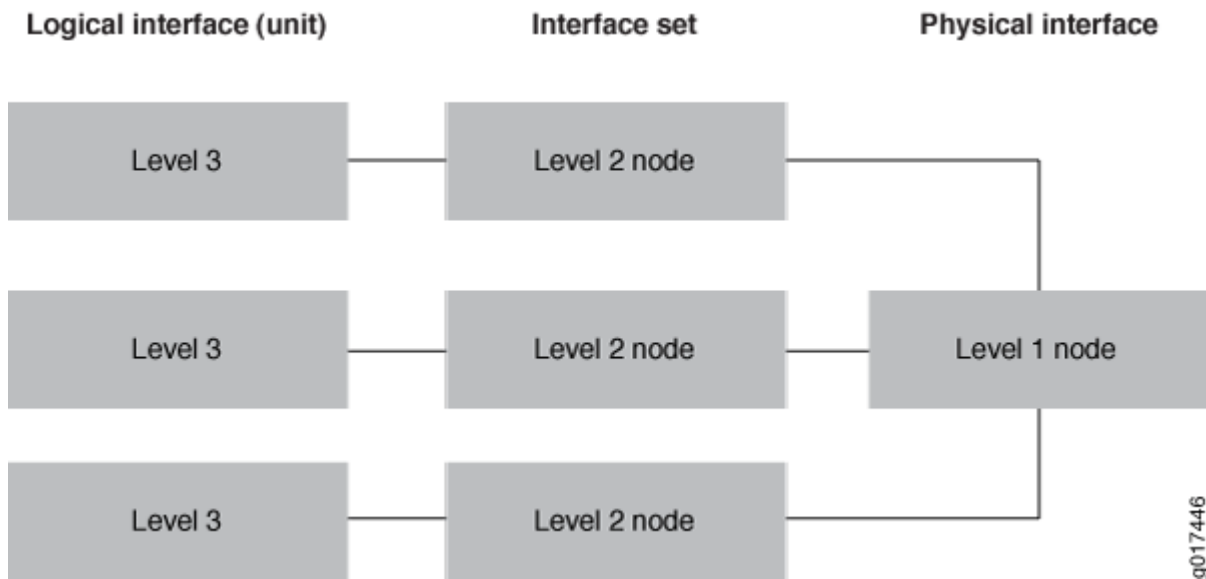


**NOTE:** Starting with Junos OS 18.4R1, you can apply dynamic and static logical interfaces in the same dynamic interface set on all MPCs that support 4 and 5-level hierarchical CoS. You can also apply dynamic interface sets in dynamic interface sets. Starting with Junos OS 19.3R1, you can apply an input traffic control profile (TCP) to a dynamic logical interface set in 4-level hierarchical scheduling or to two dynamic logical interface sets in 5-level hierarchical scheduling. Thus, Junos CoS enables you to dynamically assign a static input TCP with shaping-rate to a dynamic interface-set to enforce a customer's SLA. If no such SLA enforcement is needed, you can configure a static TCP that is designated as the default input TCP assigned to any dynamic interface-set that does not already have an explicitly assigned input TCP.

## Two-Level Hierarchical Scheduling

Two-level hierarchical scheduling limits the number of hierarchical levels in the scheduling hierarchy to two as shown in [Figure 11 on page 91](#). In this configuration, interface sets are not configured and only the logical interfaces have traffic control profiles (TCPs). Configuring two levels of hierarchy on MPCs that support more levels preserves resources and allows the system to scale higher.

**Figure 11: Two-Level Hierarchical Scheduling**



In a two-level scheduling hierarchy, all logical interfaces and interface sets share a single node; no hierarchical relationship is formed.

You control two-level hierarchical scheduling by setting the `maximum-hierarchy-levels` option under the `[edit interfaces interface-name hierarchical-scheduler]` hierarchy to 2:

- If the `maximum-hierarchy-levels` option is not set, then interface sets can be at either level 2 or level 3, depending on whether the member logical interfaces within the interface set have a traffic control profile.
- If any member logical interface has a traffic-control profile, then the interface set is always a level 2 CoS scheduler node.
- If no member logical interface has a traffic-control profile, the interface set is always a level 3 CoS scheduler node.
- If the `maximum-hierarchy-levels` option is set, then the interface set can only be at level 3; it cannot be at level 2. In this case, if you configure a level 2 interface set, you generate Packet Forwarding Engine errors.

Table 9 on page 92 summarizes the interface hierarchy and the CoS scheduler node levels for two-level hierarchical scheduling.

**Table 9: Two-Level Hierarchical Scheduling—Interface Hierarchy Versus Scheduling Nodes**

Level 1	Level 2	Level 3
Physical interface	Logical interface	One or more queues
Physical interface	Interface set	One or more queues

To configure two-level hierarchical scheduling, include the `hierarchical-scheduler` statement at the [edit interfaces *interface-name*] hierarchy level and set the `maximum-hierarchy-levels` option to 2.

```
[edit interfaces]
interface-name {
  hierarchical-scheduler {
    maximum-hierarchy-levels 2;
  }
}
```



**CAUTION:** MPC3E, 32x10GE MPC4E, and 2x100GE + 8x10GE MPC4E MPCs support only two levels of scheduling hierarchy. When enabling hierarchical scheduling on these cards, you must explicitly set `maximum-hierarchy-levels` to 2.

### Three-Level Hierarchical Scheduling

Three-level hierarchical scheduling is supported only on MX Series routers running MPC/MIC interfaces. Three-level hierarchical scheduling supports up to eight *CoS* queues. You can configure many different three-level scheduling hierarchies, depending on the location of the interface set or the use of underlying interfaces. In all variations, the physical interface is a level 1 CoS scheduler node and the queues reside at the highest level. Configuring three levels of hierarchy on MPCs that support more levels preserves resources and allows the system to scale higher.



**NOTE:** Three-level hierarchical scheduling is supported only on subscriber interfaces and interface sets running over aggregated Ethernet interfaces on MPC/MIC interfaces in MX Series routers.

When you use three-level hierarchical scheduling, interface sets can reside at either level 3 or level 4. You can also configure an underlying logical interface at level 3 and a logical interface at level 4. [Table 10 on page 93](#) summarizes the most common cases of the interface hierarchy and the CoS scheduler node levels for three-level hierarchical scheduling.

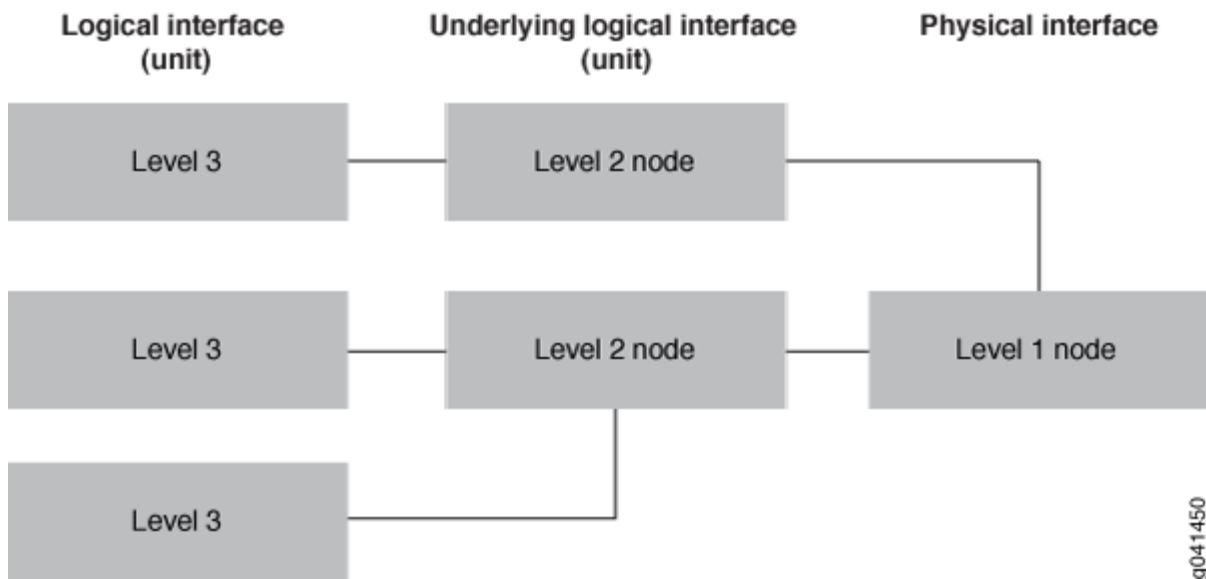
**Table 10: Three-Level Hierarchical Scheduling—Interface Hierarchy Versus CoS Scheduling Node Levels**

Level 1	Level 2	Level 3	Level 4
Physical interface	Interface set	Logical interface	One or more queues
Physical interface	Logical interface	Interface set	One or more queues
Physical interface	Underlying logical interface	Logical interface	One or more queues

In three-level hierarchical scheduling, the CoS scheduler nodes at level 1, level 2, and level 3 form a hierarchical relationship.

With a three-level hierarchical scheduling, logical interfaces can reside at level 2, or they can reside at level 3 if the logical interface at level 2 is an underlying logical interface. This is shown in [Figure 12 on page 93](#).

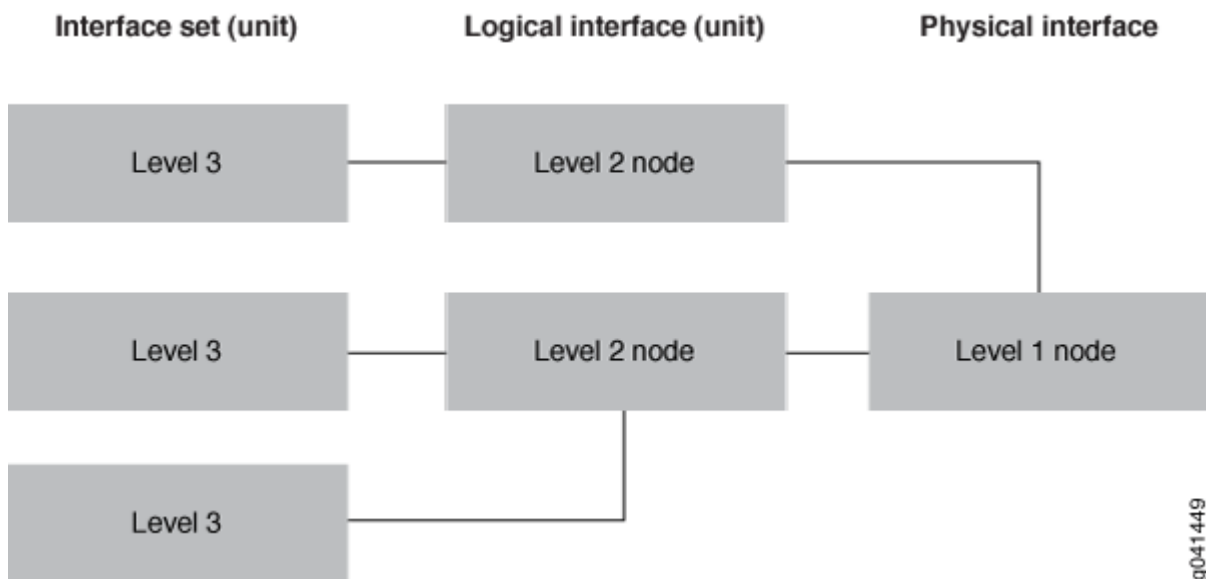
**Figure 12: Three-Level Hierarchical Scheduling—Logical Interfaces at Level 3 with Underlying Logical Interfaces at Level 2**



9041450

Another possible configuration for three-level hierarchical scheduling is shown in [Figure 13 on page 94](#). In this configuration, the logical interfaces are located at level 2 and the interface sets are located at level 3.

**Figure 13: Three-Level Hierarchical Scheduling—Logical Interfaces at Level 2 with Interface Sets at Level 3**



To configure three-level hierarchical scheduling, include the `implicit-hierarchy` option at the `[edit interfaces interface-name hierarchical-scheduler]` hierarchy level and optionally set the `maximum-hierarchy-levels` option to 3. (The default value for `maximum-hierarchy-levels` is 3.)

```
[edit interfaces]
interface-name {
  hierarchical-scheduler {
    implicit-hierarchy;
    maximum-hierarchy-levels 3;
  }
}
```

## Interface Hierarchy Versus CoS Hierarchy

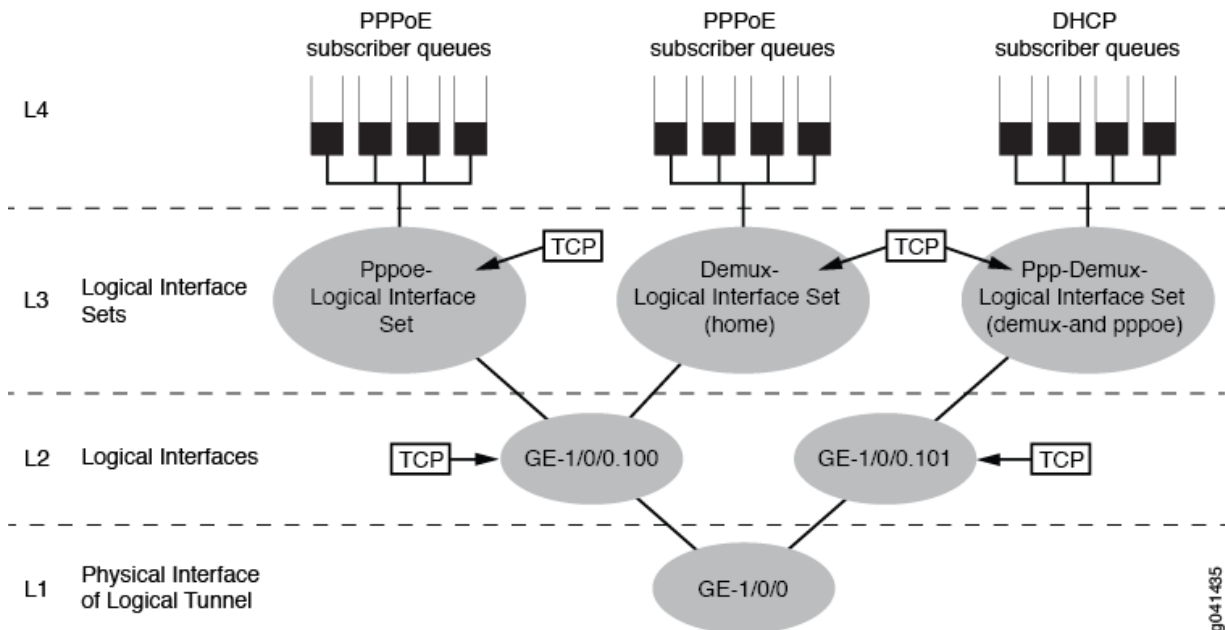
An interface hierarchy and a CoS scheduling hierarchy are distinctly different. Interface hierarchy refers to the relationship between the various interfaces—for example, the relationship between logical interfaces and an interface set, the relationship between a logical interface and an underlying logical interface, or the relationship between the physical interface and the logical interface. CoS scheduling

hierarchy refers to the hierarchical relationship between the CoS scheduler nodes. In two-level hierarchical scheduling, no hierarchy is formed between the CoS scheduler nodes—the logical interface and interface set share a single level 2 scheduler node. However, when you use the `implicit-hierarchy` option for three-level hierarchical scheduling, the CoS scheduler nodes form a scheduling hierarchy.

Figure 14 on page 95 and Figure 15 on page 96 provide two scenarios for this discussion. Figure 14 on page 95 shows an interface hierarchy where a Gigabit Ethernet interface (`ge-1/0/0`) is the physical interface. Two logical interfaces (`ge-1/0/0.100` and `ge-1/0/0.101`) are configured on the physical interface:

- Logical interface `ge-1/0/0.100` is a member of a PPPoE interface set and a Demux interface set.
- Logical interface `ge-1/0/0.101` is a member of a demux interface set.

Figure 14: Logical Interfaces at Level 2 and Interface Sets at Level 3



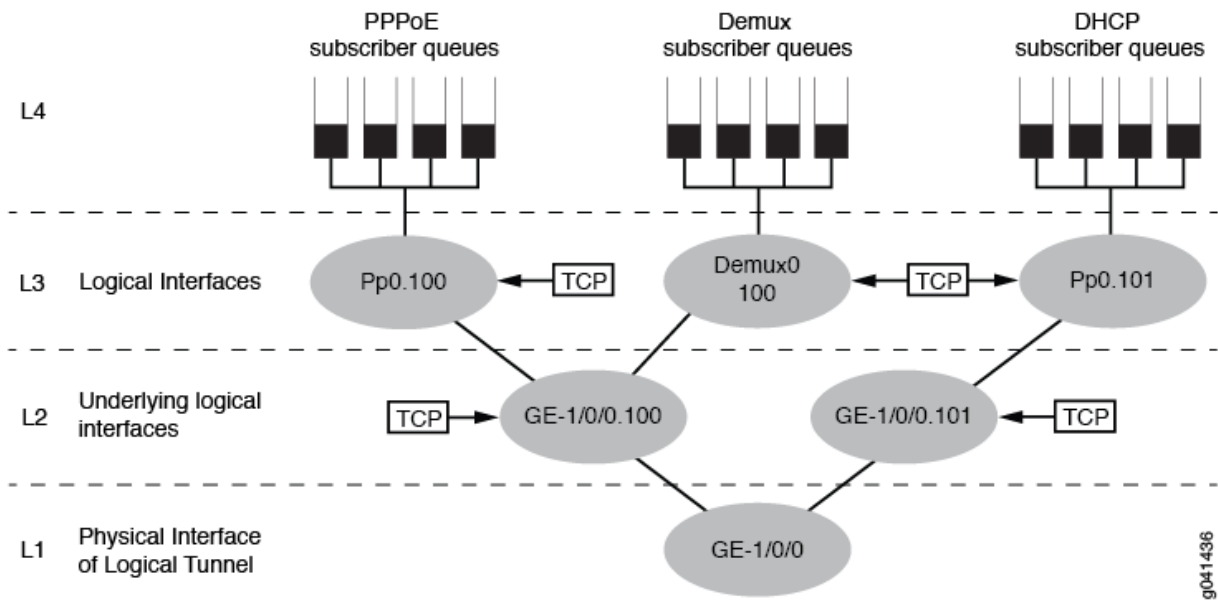
Each interface set has a dedicated queue. The CoS scheduler nodes at level 1 (physical interface), level 2 (underlying logical interfaces), and level 3 (interface sets) form a scheduling hierarchy.

To configure this scenario, you must include the `implicit-hierarchy` option under the `hierarchical-scheduler` statement on physical interface `ge-1/0/0` and configure and apply traffic-control profiles on each interface set and underlying logical interface.

Figure 15 on page 96 shows an interface hierarchy where Gigabit Ethernet interface `ge-1/0/0` is the physical interface. Three logical interfaces are configured:

- Two logical interfaces (Pp0.100 and Demux0.100) reside on the underlying logical interface ge-1/0/0.100.
- A third logical interface (Pp0.101) resides on the underlying logical interface ge-1/0/0.101.

**Figure 15: Logical Interfaces at Level 3 and Underlying Logical Interfaces at Level 2**



Each logical interface has a dedicated queue. The CoS scheduler nodes at level 1 (physical interface), level 2 (underlying logical interfaces), and level 3 (logical interfaces) form a scheduling hierarchy.

To configure this scenario, you must include the `implicit-hierarchy` option under the `hierarchical-scheduler` statement on physical interface GE-1/0/0 and configure and apply traffic-control profiles on each logical interface and underlying logical interface.

You can configure many different three-level scheduling hierarchies; [Figure 14 on page 95](#) and [Figure 15 on page 96](#) present just two possible scenarios. [Table 10 on page 93](#) summarizes the possible interface locations and CoS scheduler nodes.

## Four-Level Hierarchical Scheduling

Beginning with Junos OS Release 16.1, four-level hierarchical scheduling is supported on MX Series routers running NG-MPC2E, NG-MPC3E, MPC5, and MPC7 line cards. Four-level hierarchical scheduling supports up to eight *class of service* queues. In four-level scheduling hierarchies, the physical interface is a level 1 CoS scheduler node and the queues reside at level 5.





**NOTE:** Four-level hierarchical scheduling is not supported agent circuit identifier (ACI) or aggregated Ethernet (AE) interfaces.

When you use four-level hierarchical scheduling, interface sets reside at levels 2 and 3 and logical interfaces reside at levels 3 and 4. [Table 11 on page 97](#) summarizes the most common case of the interface hierarchy and the CoS scheduler node levels for four-level hierarchical scheduling. Starting with Junos OS 18.4R1, you can apply dynamic and static logical interfaces in the same dynamic interface set on all MPCs that support 4 and 5-level hierarchical CoS. You can also apply dynamic interface sets in dynamic interface sets.

**Table 11: Four-Level Hierarchical Scheduling—Interface Hierarchy Versus CoS Scheduling Node Levels**

Level 1	Level 2	Level 3	Level 4	Level 5
Physical interface	Interface set	Customer VLAN (C-VLAN)	Session Logical Interface (ppp or dhcp)	One or more queues

In four-level hierarchical scheduling, the CoS scheduler nodes at level 1, level 2, level 3, and level 4 form a hierarchical relationship.

To configure four-level hierarchical scheduling, include the `implicit-hierarchy` option at the `[edit interfaces interface-name hierarchical-scheduler]` hierarchy level and set the `maximum-hierarchy-levels` option to 4.

```
[edit interfaces]
  interface-name {
    hierarchical-scheduler {
      implicit-hierarchy;
      maximum-hierarchy-levels 4;
    }
  }
```

## Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
19.3R1	Starting with Junos OS 19.3R1, you can apply an input traffic control profile (TCP) to a dynamic logical interface set in 4-level hierarchical scheduling or to two dynamic logical interface sets in 5-level hierarchical scheduling. Thus, Junos CoS enables you to dynamically assign a static input TCP with shaping-rate to a dynamic interface-set to enforce a customer's SLA. If no such SLA enforcement is needed, you can configure a static TCP that is designated as the default input TCP assigned to any dynamic interface-set that does not already have an explicitly assigned input TCP.
18.4R1	Starting with Junos OS 18.4R1, you can apply dynamic and static logical interfaces in the same dynamic interface set on all MPCs that support 4 and 5-level hierarchical CoS. You can also apply dynamic interface sets in dynamic interface sets.
16.1	Beginning with Junos OS Release 16.1, four-level hierarchical scheduling is supported on MX Series routers running NG-MPC2E, NG-MPC3E, MPC5, and MPC7 line cards.

## RELATED DOCUMENTATION

[Hardware Requirements for Dynamic Hierarchical CoS | 98](#)

*Configuring Hierarchical Schedulers for CoS*

[Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links | 101](#)

[Configuring Hierarchical CoS on a Static PPPoE Subscriber Interface | 103](#)

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)

*hierarchical-scheduler (Subscriber Interfaces on MX Series Routers)*

## Hardware Requirements for Dynamic Hierarchical CoS

[Table 12 on page 99](#) lists the hardware requirements based on subscriber interface type for hierarchical scheduling in dynamic CoS configurations.

**Table 12: Hardware Required for Dynamic Hierarchical CoS Configurations**

Dynamic CoS Configuration	Subscriber Interface Type	EQ DPCs on MX Series Routers	MPC Q/MIC Modules on MX Series Routers
<b>Hierarchical CoS</b>	Static and dynamic VLANs	Yes	Yes
	Static and dynamic VLANs over aggregated Ethernet	Yes	Yes
	Static or dynamic IP demux interfaces	Yes	Yes
	Static or dynamic IP demux interfaces over aggregated Ethernet	Yes	Yes
	Static or dynamic VLAN demux interfaces	No	Yes
	Static or dynamic VLAN demux interfaces over aggregated Ethernet	No	Yes
	Static PPPoE interfaces	No	Yes
	Dynamic PPPoE interfaces	No	Yes
	Static or dynamic PPPoE interfaces over aggregated Ethernet	No	Yes
	L2TP LAC tunnel over PPP	No	Yes

**Table 12: Hardware Required for Dynamic Hierarchical CoS Configurations (Continued)**

Dynamic CoS Configuration	Subscriber Interface Type	EQ DPCs on MX Series Routers	MPC Q/MIC Modules on MX Series Routers
	L2TP LNS inline service over PPP	No	Yes

**RELATED DOCUMENTATION**

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

**Configuring Static Hierarchical Scheduling in a Dynamic Profile**

You configure static scheduling and queuing in a dynamic profile for subscriber access. To configure CoS in a dynamic profile for subscriber access using static scheduling and queuing parameters:

**1. Configure the static CoS parameters in the [edit class-of-service] hierarchy.**

- a. Enable the hierarchical scheduler for the interface.

See "[Understanding Hierarchical CoS for Subscriber Interfaces](#)" on page 90.

- b. Configure the scheduler map and schedulers.

When you configure static scheduling and queuing in a dynamic profile, you reference the scheduler map in the dynamic profile.

See *Configuring Schedulers*.

- c. Configure the drop profiles.

See *RED Drop Profiles for Congestion Management*.

- d. Configure the forwarding classes.

See *Configuring a Custom Forwarding Class for Each Queue*.

- e. Configure the rewrite-rules and classifier definitions.

See *Configuring Rewrite Rules* and *Configuring Behavior Aggregate Classifiers*.

See *The Junos OS CoS Components Used to Manage Congestion and Control Service Levels* for information about configuring the remaining CoS parameters.

2. Configure a static or dynamic subscriber interface that can be referenced in the dynamic profile.
3. Configure CoS parameters in a dynamic profile.
  - a. Configure the dynamic profile.  
See *Configuring a Basic Dynamic Profile*.
  - b. Configure traffic shaping and scheduling parameters in the dynamic profile using a traffic-control profile. Reference the scheduler map you configured in the static [edit class-of-service] hierarchy.  
See *Configuring Static Traffic Shaping and Scheduling Parameters in a Dynamic Profile*.
  - c. Apply CoS parameters to a subscriber interface by referencing an interface in the dynamic profile.  
See *Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile*.
4. To configure default values for subscribers on login, and enable subscribers to replace other CoS parameters when replacing services, configure variables in the dynamic profile.  
See *Configuring Static Default Values for Traffic Scheduling and Shaping*.

## RELATED DOCUMENTATION

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

*CoS for Subscriber Access Overview*

[Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access | 104](#)

## Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links

You can enable hierarchical CoS on a subscriber interface with an underlying aggregated Ethernet interface.

Before you begin, configure the subscriber interface with aggregated Ethernet.

- To configure a VLAN interface over aggregated Ethernet with link protection, see *Configuring a Static or Dynamic VLAN Subscriber Interface over Aggregated Ethernet* and *Configuring Link Protection for Aggregated Ethernet Interfaces*.
- To configure a demux subscriber interface:

For static and dynamic IP demux interfaces, see *Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet*.

For static and dynamic VLAN demux interfaces, see *Configuring a Static or Dynamic VLAN Demux Subscriber Interface over Aggregated Ethernet*.



**BEST PRACTICE:** Link protection is not required for IP or demux subscriber interfaces. We recommend that you enable targeted distribution on the demux interface to provide accurate hierarchical scheduling for these links. See "[Providing Accurate Scheduling for a Demux Subscriber Interface of Aggregated Ethernet Links](#)" on page 212.



**BEST PRACTICE:** While subscribers are active on aggregated Ethernet physical interfaces with targeted distribution, we recommend that you do not change any attribute of the physical interfaces, such as MTU. Instead, perform the following steps:

1. Log out all the subscribers.
2. Disable the interface.
3. Make the desired attribute changes.
4. Reenable the interface.

If you do not follow these steps, the commit check for your configuration fails, starting in Junos OS Release 19.2. In earlier releases, changing the attribute values while subscribers are active brings down the physical interface and all subscribers using that interface.

To avoid service interruptions, we recommend that you make the changes during a maintenance window.

To configure hierarchical CoS on the link aggregation (LAG) bundle:

1. Specify that you want to access the LAG bundle.

```
user@host# edit interfaces aex
```

2. Configure the link aggregation (LAG) bundle with hierarchical scheduler mode.

```
[edit interfaces aex]
user@host# set hierarchical-scheduler
```

You can then attach static or dynamic traffic shaping and scheduling parameters at the aggregated Ethernet logical interface or its underlying physical interface. See:

- *Configuring Traffic Scheduling and Shaping for Subscriber Access*
- *Configuring Schedulers in a Dynamic Profile for Subscriber Access*
- *Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile*

## RELATED DOCUMENTATION

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

*Verifying the Scheduling and Shaping Configuration for Subscriber Access*

*CoS for Subscriber Access Overview*

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

## Configuring Hierarchical CoS on a Static PPPoE Subscriber Interface

Before you begin:

- Configure the static PPPoE subscriber interface.

See *Point-to-Point Protocol over Ethernet (PPPoE)*.

You can configure hierarchical CoS on a static PPPoE subscriber interface.

To configure hierarchical CoS on a static PPPoE subscriber interface:

1. Specify the PPPoE interface that you want to configure.

```
user@host# edit interfaces pppoe-interface-name
```

2. Configure the hierarchical scheduler for the interface.

```
[edit interfaces interface-name]  
user@host# set hierarchical-scheduler
```

3. (Optional) Group the PPPoE interfaces in an interface set.

```
[edit]  
user@host# edit interfaces interface-set interface-set-name
```

You can now configure static traffic and scheduling parameters for each traffic-control profile, and attach each traffic-control profile to the PPPoE interface or the PPPoE interface set. For more information, see *Using the CLI to Modify Traffic-Control Profiles That Are Currently Applied to Subscribers*.

## RELATED DOCUMENTATION

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

*CoS for PPPoE Subscriber Interfaces Overview*

*Verifying the Scheduling and Shaping Configuration for Subscriber Access*

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

## Example: Maintaining a Constant Traffic Flow by Configuring a Static VLAN Interface with a Dynamic Profile for Subscriber Access

### IN THIS SECTION

- [Requirements | 104](#)
- [Overview | 105](#)
- [Configuration | 106](#)
- [Verification | 119](#)

This example shows how to configure a static VLAN interface with a dynamic profile using static schedulers and CoS parameters for subscriber access to maintain a constant traffic flow. The CoS parameters configure a best-effort data service for subscribers.

### Requirements

Before you begin, be sure that your environment meets the following requirements:

- The interface is hosted on an MX Series router.
- For hierarchical scheduling configurations, hierarchical scheduling is enabled in the static CLI for the interface referenced in the dynamic profile. If not, the dynamic profile fails.
- Only one traffic-control-profile is configured under a dynamic profile.



- The output-traffic-control-profile that binds the traffic-control profile to the interface is defined within the same dynamic profile as the interface.

## Overview

### IN THIS SECTION

- [Topology | 105](#)

In a dynamic profile, you can configure VLAN subscriber interfaces over the following statically created *logical interface* types:

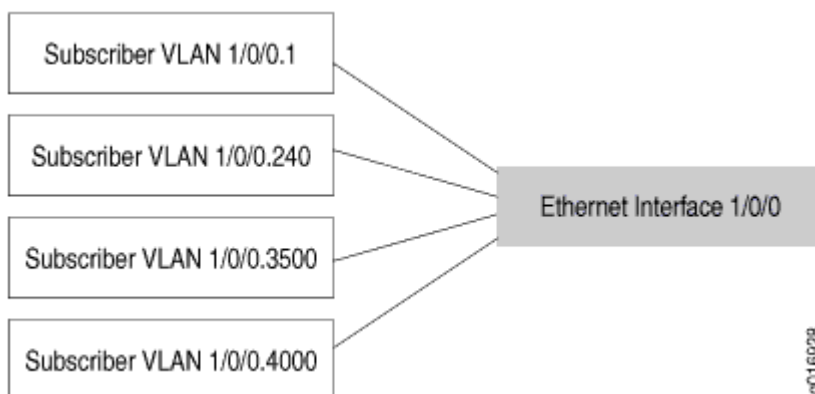
- GE—Gigabit Ethernet
- XE—10-Gigabit Ethernet
- AE—Aggregated Ethernet

## Topology

We recommend that you configure each subscriber on a statically created VLAN.

[Figure 16 on page 105](#) shows an example of subscriber interfaces on an individual VLAN.

**Figure 16: VLAN Subscriber Interfaces**



You can further separate VLANs on subscriber interfaces by configuring a VLAN interface as the underlying interface for a set of IP demux interfaces.

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 106](#)
- [Configuring a Subscriber Interface with a Static VLAN | 107](#)
- [Associating the Dynamic Profile with a Statically Created Interface | 109](#)
- [Configuring the Firewall Filter | 111](#)
- [Configuring Static Schedulers in a Dynamic Profile | 113](#)
- [Associating the Scheduler with a Scheduler Map | 115](#)
- [Configuring and Applying Static Traffic Shaping and Scheduling Parameters in a Dynamic Profile | 116](#)

To configure a static VLAN interface with a dynamic profile for subscriber access, perform these tasks:

### CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the [edit] hierarchy level.

```
set interfaces ge-2/2/0
set interfaces ge-2/2/0 hierarchical-scheduler
set interfaces ge-2/2/0 vlan-tagging
set interfaces ge-2/2/0 vlan-tagging unit 100 vlan-id 100
set interfaces ge-2/2/0 vlan-tagging unit 100 vlan-id 100 family inet
set interfaces ge-2/2/0 vlan-tagging unit 100 vlan-id 100 family inet unnumbered-address lo0.0
preferred-source-address 10.0.0.1
set dynamic-profiles data-service
set dynamic-profiles data-service interfaces $junos-interface-ifd-name
set dynamic-profiles data-service interfaces $junos-interface-ifd-name unit $junos-underlying-
interface-unit
set dynamic-profiles data-service interfaces $junos-interface-ifd-name unit $junos-underlying-
interface-unit family inet
set dynamic-profiles data-service firewall family inet filter filter EF_limit_G=768K
set dynamic-profiles data-service firewall family inet filter filter EF_limit_G=768K term EF
set dynamic-profiles data-service firewall family inet filter filter EF_limit_G=768K term default
set dynamic-profiles data-service firewall family inet filter filter EF_limit_G=768K term EF
from forwarding-class EF
```

```

set dynamic-profiles data-service firewall family inet filter filter EF_limit_G=768K term EF
then policer POL_EF_G=768K
set dynamic-profiles data-service firewall family inet filter filter EF_limit_G=768K term
default then accept
set dynamic-profiles data-service class-of-service schedulers be-scheduler
set dynamic-profiles data-service class-of-service schedulers be-scheduler buffer-size remainder
set dynamic-profiles data-service class-of-service schedulers be-scheduler drop-profile-map loss-
priority any protocol any
set dynamic-profiles data-service class-of-service schedulers be-scheduler drop-profile-map loss-
priority any protocol any drop-profile drop3
set dynamic-profiles data-service class-of-service schedulers be-scheduler priority low
user@host# set dynamic-profiles data-service class-of-service schedulers be-scheduler transmit-
rate percent 40
set dynamic-profiles data-service class-of-service schedulers be-scheduler excess-rate percent 90
set dynamic-profiles data-service class-of-service schedulers be-scheduler excess-priority high
set dynamic-profiles data-service class-of-service scheduler-maps data-service-map
set dynamic-profiles data-service class-of-service scheduler-maps data-service-map forwarding-
class best-effort
set dynamic-profiles data-service class-of-service scheduler-maps data-service-map forwarding-
class best-effort scheduler be-scheduler
set dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-service
set dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-service
scheduler-map data-service-map
set dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-service
shaping-rate 50k
set dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-service
guaranteed-rate 10k
set dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-service
delay-buffer-rate 10k
set dynamic-profiles data-service class-of-service interfaces $junos-interface-ifd-name unit
$junos-underlying-interface-unit output-traffic-control-profile tcp-data-service

```

## Configuring a Subscriber Interface with a Static VLAN

### Step-by-Step Procedure

After you configure a static VLAN interface, you can reference it in a dynamic profile.

1. Configure the static VLAN interface.

```
[edit]
user@host# set interfaces ge-2/2/0
```

2. Enable hierarchical scheduling for the interface.

```
[edit interfaces ge-2/2/0]
user@host# set hierarchical-scheduler
```

3. Enable VLAN tagging.

```
[edit interfaces ge-2/2/0]
user@host# set vlan-tagging
```

4. Configure the unit and assign a VLAN ID.

```
[edit interfaces ge-2/2/0 vlan-tagging]
user@host# set unit 100 vlan-id 100
```

5. Define the family address type (inet for IPv4) for the VLAN interface.

```
[edit interfaces ge-2/2/0 vlan-tagging unit 100 vlan-id 100]
user@host# set family inet
```

6. Enable the physical interface to borrow an IP address from the loopback interface by setting an unnumbered interface address. Configure a secondary IP address on the loopback interface, lo0.0, and configure it as the preferred source address.

```
[edit interfaces ge-2/2/0 vlan-tagging unit 100 vlan-id 100 family inet]
user@host# set unnumbered-address lo0.0 preferred-source-address 10.0.0.1
```

## Results

Confirm the configuration of the static VLAN interface by entering the `show interfaces configuration` command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show interfaces
interfaces {
  ge-2/2/0 {
    hierarchical-scheduler;
    vlan-tagging;
    unit 100 {
      vlan-id 100;
      family inet {
        unnumbered-address lo0.0 preferred-source-address 10.0.0.1;
      }
    }
  }
}
```

## Associating the Dynamic Profile with a Statically Created Interface

### Step-by-Step Procedure

A dynamic profile is a set of characteristics, defined in a type of template, that you can use to provide dynamic subscriber access and services for broadband applications. When configuring the interface at the `[dynamic-profiles profile-name interfaces]` hierarchy level for a dynamic profile, you use variables to specify the interface name and the logical unit value. When a DHCP subscriber sends a DHCP request to the interface, the dynamic profile replaces the interface name variable and logical unit name variable with the actual interface name and logical unit number of the interface that received the DHCP request.



**NOTE:** Configuration of the interface name variable and logical interface name variable at the `[edit dynamic-profiles profile-name interfaces]` hierarchy level is required for a dynamic profile to function.

1. Create the new dynamic profile for data services for subscribers.

```
[edit]
user@host# set dynamic-profiles data-service
```

2. Define the *interface-name* variable statement with the internal **\$junos-interface-ifd-name** variable used by the router to match the interface name of the receiving interface.

```
[edit dynamic-profiles data-service]
user@host# set interfaces $junos-interface-ifd-name
```

3. Define the unit statement with the internal variable.

- When referencing an existing interface, specify the **\$junos-underlying-interface-unit** variable used by the router to match the unit value of the receiving interface.
- When creating dynamic interfaces, specify the **\$junos-interface-unit** variable used by the router to generate a unit value for the interface.

```
[edit dynamic-profiles data-service interfaces $junos-interface-ifd-name]
user@host# set unit $junos-underlying-interface-unit
```

or

```
[edit dynamic-profiles data-service interfaces $junos-interface-ifd-name]
user@host# set unit $junos-interface-unit
```

4. Define the family address type (inet for IPv4) for the **\$junos-interface-unit** variable.

```
[edit dynamic-profiles data-service interfaces $junos-interface-ifd-name unit $junos-
underlying-interface-unit]
user@host# set family inet
```

## Results

Confirm the configuration of the dynamic profile by entering the `show dynamic-profiles configuration` command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles
dynamic-profiles {
  data-service {
    interfaces {
      $junos-interface-ifd-name {
        unit $junos-underlying-interface-unit {
          family inet;
        }
      }
    }
  }
}
```

## Configuring the Firewall Filter

### Step-by-Step Procedure

To configure a static VLAN interface with a dynamic profile for subscriber access, you can configure a firewall filter to provide enhanced security by blocking packets based on various match criteria, such as subjecting traffic to a policer for rate limiting, assigning the traffic to a class-of-service (CoS) forwarding class for later queuing and packet rewrite operations, or directing traffic to a specific routing instance.

1. Configure the family address type (inet for IPv4) for the firewall filter and specify the filter name.

We recommend that you name the filter something that indicates the filter's purpose. In this example, we use the bandwidth limit settings.

```
[edit dynamic-profiles data-service]
user@host# set firewall family inet filter EF_limit_G=768K
```

- Specify the term names for the filter. Make each term name unique and represent what its function is. The first term matches traffic that has been classified into the Expedited Forwarding (EF) class, and the second term matches all non-EF traffic.

```
[edit dynamic-profiles data-service firewall family inet filter EF_limit_G=768K]
user@host# set term EF
user@host# set term default
```

- In each firewall filter term, specify the conditions used to match components of a packet. Configure the first term to match all traffic classified as EF class.

```
[edit dynamic-profiles data-service firewall family inet filter EF_limit_G=768K term EF]
user@host# set from forwarding-class EF
```

- Specify the actions to take when the packet matches the condition in the first term. Send the EF traffic to the policer named **POL\_EF\_G=768K**.

```
[edit dynamic-profiles data-service firewall family inet filter EF_limit_G=768K term EF]
user@host# set then policer POL_EF_G=768K
```

- Specify the action to take when the packet matches the condition in the second term. All non-EF packet traffic is accepted.

```
[edit dynamic-profiles data-service firewall family inet filter EF_limit_G=768K term default]
user@host# set then accept
```

## Results

Confirm the configuration by entering the `show dynamic-profiles data-service firewall` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles data-service firewall
family inet {
  filter EF_limit_G=768K {
    term EF {
      from {
```



```

        forwarding-class EF;
    }
    then policer POL_EF_G=768K;
}
term default {
    then accept;
}
}
}
}

```

## Configuring Static Schedulers in a Dynamic Profile

### Step-by-Step Procedure

You can configure static scheduling and queuing parameters in a dynamic profile for subscriber access. Schedulers are part of the basic class-of-service (CoS) infrastructure. You must define at least one scheduler per forwarding class. Schedulers indicate a forwarding class's priority, transmit weight, and buffer size, as well as various shaping and rate control mechanisms.

1. Specify the best-effort scheduler for which you want to configure parameters.

```

[edit dynamic-profiles data-service class-of-service]
user@host# set schedulers be-scheduler

```



**NOTE:** Set `schedulers` to the name of the scheduler to be configured or to the Junos OS predefined variable (`$junos-cos-scheduler`) used for dynamic subscriber interfaces. The predefined variable is replaced with the scheduler name obtained from the RADIUS server when a subscriber authenticates over the interface to which the dynamic profile is attached.

2. (Optional) Configure the buffer size to use the remaining buffer available.

This parameter allows you to specify an explicit buffer size, either as a percent of interface speed or as a function of time (specified in microseconds).

```

[edit dynamic-profiles data-service class-of-service schedulers be-scheduler]
user@host# set buffer-size remainder

```

3. (Optional) Configure the drop-profile map to associate one or more drop profiles with a queue.

The default random early detection (RED) drop profile is used when no explicit drop profile mapping is specified. Specify a packet-loss priority (PLP) level of any, and for the specified scheduler to accept any protocol type.

```
[edit dynamic-profiles data-service class-of-service schedulers be-scheduler]
user@host# set drop-profile-map loss-priority any protocol any
```

4. (Optional) Configure the drop profile to map a fill level (fullness of a queue) to a drop probability (probability that a packet is dropped).

```
[edit dynamic-profiles data-service class-of-service schedulers be-scheduler drop-profile-map
loss-priority any protocol any]
user@host# set drop-profile drop3
```

You enable RED by applying a drop profile to a scheduler.

5. (Optional) Configure the queue's scheduler priority to a specific level (low) for guaranteed rate traffic.

```
[edit dynamic-profiles data-service class-of-service schedulers be-scheduler]
user@host# set priority low
```

6. (Optional) Configure the queue's transmit weight [in bits per second (bps)] or as a percentage of transmission capacity.

```
[edit dynamic-profiles data-service class-of-service schedulers be-scheduler]
user@host# set transmit-rate percent 40
```

The transmit rate guarantees the rate for the queue, assuming no priority-based starvation occurs. When you do not specify a transmit weight, or when the transmit rate is reached, the queue can only send excess-rate traffic because that queue's priority is demoted to the excess region. A percentage of zero (0) drops all packets in the queue.

7. (Optional) Configure the queue's weight as either a percentage, or a proportion, for any unused bandwidth traffic to share.

```
[edit dynamic-profiles data-service class-of-service schedulers be-scheduler]
user@host# set excess-rate percent 90
```

Behavior varies based on interface mode, explicit configuration, and whether any other queues have explicit weight configured. By default, excess bandwidth between the guaranteed and shaped rate is shared equally among queues.

8. (Optional) Configure the priority of how excess bandwidth traffic is sent on a scheduler in a dynamic profile.

```
[edit dynamic-profiles data-service class-of-service schedulers be-scheduler]
user@host# set excess-priority high
```

To prevent the queue from sending any excess rate traffic, set to none.

## Results

Confirm the configuration of the scheduler with static values in the dynamic profile by entering the `show dynamic-profiles data-service class-of-service configuration` command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles data-service class-of-service
class-of-service {
  schedulers {
    be-scheduler {
      buffer-size remainder;
      drop-profile-map loss-priority any protocol any drop-profile drop3;
      priority low;
      transmit-rate percent 40;
      excess-rate percent 90;
      excess-priority high;
    }
  }
}
```

## Associating the Scheduler with a Scheduler Map

### Step-by-Step Procedure

After you define your schedulers, you must link them to a set of queues on a logical interface using a scheduler map. Applying a scheduler map to an interface places the related set of schedulers and drop profiles into effect.

1. Configure the scheduler map name.

```
[edit dynamic-profiles data-service class-of-service]
user@host# set scheduler-maps data-service-map
```

2. Configure a forwarding class to associate a scheduler with a scheduler map.

```
[edit dynamic-profiles data-service class-of-service scheduler-maps data-service-map]
user@host# set forwarding-class best-effort
```

3. Associate the scheduler you previously defined (**be-scheduler**) with the scheduler map.

```
[edit dynamic-profiles data-service class-of-service scheduler-maps data-service-map
forwarding-class best-effort]
user@host# set scheduler be-scheduler
```

## Results

Confirm the configuration of the scheduler map by entering the `show dynamic-profiles data-service class-of-service scheduler-maps` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles data-service class-of-service scheduler-maps
scheduler-maps {
  data-service-map {
    forwarding-class best-effort scheduler be-scheduler;
  }
}
```

## Configuring and Applying Static Traffic Shaping and Scheduling Parameters in a Dynamic Profile

### Step-by-Step Procedure

Configure static traffic shaping and scheduling parameters in a traffic-control profile. A traffic-control profile is a generic class-of-service (CoS) container that you can apply at all points of a CoS hierarchy to affect the committed information rate (CIR), peak information rate (PIR), and excess bandwidth handling.

You can specify the traffic-control profile at the port, logical interface, or logical interface-set level. The traffic-control profile also references the scheduler map.

1. Create the traffic-control profile and assign it a name.

```
[edit dynamic-profiles data-service class-of-service]
user@host# edit traffic-control-profiles tcp-data-service
```

2. Apply the static scheduler map, **data-service-map**, that you previously configured.

```
[edit dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-
service]
user@host# set scheduler-map data-service-map
```

3. Configure the shaping rate [in bits per second (bps)] to use for the scheduler in the dynamic profile.

```
[edit dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-
service]
user@host# set shaping-rate 50k
```

The shaping rate places a maximum limit on a queue's transmit capacity. By default, the shaping rate is equal to the interface speed/shaping rate enabling the queue to send at the full rate of the interface.

4. Configure the guaranteed rate [in bits per second (bps)] to use for the scheduler in the dynamic profile.

```
[edit dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-
service]
user@host# set guaranteed-rate 10k
```

The guaranteed rate is the minimum bandwidth the queue can receive; if excess physical interface bandwidth is available for use, the logical interface can receive more than the guaranteed rate provisioned for the interface, depending on how you choose to manage excess bandwidth and the interface's mode of PIR compared to CIR/PIR.

5. Configure the delay-buffer rate [in bits per second (bps)] based on the delay-buffer calculation.

```
[edit dynamic-profiles data-service class-of-service traffic-control-profiles tcp-data-
service]
user@host# set delay-buffer-rate 10k
```

The delay buffer rate setting at one level of the hierarchy becomes the reference bandwidth used at the next higher level, and the sum of the reference bandwidth cannot exceed the value used at a lower level. If you do not include this statement, the delay-buffer rate is based on the guaranteed rate if one is configured, or on the shaping rate if no guaranteed rate is configured.

6. After you configure the traffic shaping and scheduling CoS parameters in a dynamic profile, you apply them to an interface. The output traffic-control profile enables you to provide traffic scheduling to the interface.

Configure the interface name and logical interface using a variable, and apply the output traffic-control profile to the interface. Specify the previously defined traffic-control profile, **tcp-data-service**.

```
[edit dynamic-profiles data-service class-of-service]
user@host# set interfaces $junos-interface-ifs-name unit $junos-underlying-interface-unit
output-traffic-control-profile tcp-data-service
```

## Results

Confirm the configuration and application of the static traffic shaping and scheduling parameters by entering the `show dynamic-profiles` configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show dynamic-profiles
dynamic-profiles {
  data-service {
    class-of-service {
      interfaces {
        $junos-interface-ifs-name {
          unit $junos-underlying-interface-unit {
            output-traffic-control-profile tcp-data-service;
          }
        }
      }
    }
  }
  traffic-control-profiles {
```

```
        tcp-data-service {
            scheduler-map data-service-map;
            shaping-rate 50k;
            guaranteed-rate 10k;
            delay-buffer-rate 10k;
        }
    }
}
}
```

## Verification

### IN THIS SECTION

- [Verifying Traffic Shaping and Scheduling Profiles for Subscriber Access | 119](#)
- [Verifying the Mapping of Schedulers for Subscriber Access | 120](#)

Confirm that the configuration is working properly.

### Verifying Traffic Shaping and Scheduling Profiles for Subscriber Access

#### Purpose

View the class-of-service (CoS) configurations that are referenced in a dynamic profile for subscriber access.

#### Action

```
user@host> show class-of-service traffic-control-profile
Traffic control profile: tcp-data-service, Index: 57625
  Shaping rate: 50000
  Scheduler map: data-service-map
  Delay Buffer rate: 10000
  Guaranteed rate: 10000
```

## Meaning

The Shaping rate, Delay Buffer rate, and Guaranteed rate fields indicate rates of 50,000 bps, 10,000 bps, and 10,000 bps, respectively, for the traffic-control profile.

## Verifying the Mapping of Schedulers for Subscriber Access

### Purpose

Display the mapping of schedulers to forwarding classes and a summary of scheduler parameters for each entry.

### Action

```
user@host> show class-of-service scheduler-map
Scheduler map: data-service-map, Index: 84

Scheduler: be-scheduler, Index: 8721, Forwarding class: best-effort
  Transmit rate: 40 percent, Rate Limit: none, Maximum buffer delay: 39 ms,
  Priority: low
  Drop profiles:
    Loss priority  Protocol  Index  Name
    Any           Any      8724   drop3
```

## Meaning

The Scheduler map field indicates the parameters are for the best-effort scheduler. The Transmit rate field shows 40 percent; the Rate Limit field indicates no limit; and the Drop profiles fields are for drop3.

## RELATED DOCUMENTATION

[CoS for Subscriber Access Overview](#)

[Guidelines for Configuring Dynamic CoS for Subscriber Access](#)

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)



# Applying CoS to Groups of Subscriber Interfaces

## IN THIS CHAPTER

- [CoS for Interface Sets of Subscribers Overview | 121](#)
- [Configuring an Interface Set of Subscribers in a Dynamic Profile | 124](#)
- [Example: Configuring a Dynamic Interface Set of VLAN Subscribers | 125](#)
- [Example: Configuring a Dynamic Service VLAN Interface Set of Subscribers in a Dynamic Profile | 146](#)

## CoS for Interface Sets of Subscribers Overview

### IN THIS SECTION

- [Guidelines for Configuring Dynamic Interface Sets in a Subscriber Access Network | 121](#)

Interface sets enable service providers to group logical interfaces or other interface sets so they can apply CoS parameters to all of the traffic in the group.

Interface sets are beneficial for various scenarios in a subscriber access network. For example, you can use an interface set to configure a local loop with a small number of subscribers. Interface sets are also useful for grouping a large number of subscribers into a particular service class or for defining traffic engineering aggregates for DSLAMs.

### Guidelines for Configuring Dynamic Interface Sets in a Subscriber Access Network

When configuring interface sets for subscriber access, keep the following guidelines in mind:

- You can configure interface sets of VLAN demux, PPPoE, or demux interfaces over aggregated Ethernet interfaces.

- An interface can only belong to one interface set. If you try to add the same interface to different interface sets, the commit operation fails.
- You configure the interface set and the traffic scheduling and shaping parameters in a dynamic profile. However, you must apply the traffic-control profile to the interface set in the static [edit class-of-service] hierarchy.



**NOTE:** This rule applies to all interface sets except ACI sets.

- The `$junos-interface-set-name` predefined variable is available only for RADIUS Accept messages; change of authorization (CoA) requests are not supported.
- The `$junos-aggregation-interface-set-name` is the L2 interface-set representing a logical intermediate node (DPU-C or PON tree) in the access network.
- The `$junos-phy-ifd-underlying-intf-set-name` represents a default, topology-based interface-set (based on the physical interface name with a post-pend of “-underlying”) to conserve L2 CoS nodes.
- The `$junos-svlan-interface-set-name` predefined variable locally generates an interface set name for use by dual-tagged VLAN interfaces based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical\_interface\_name - outer\_VLAN\_tag*. For example, an aggregated Ethernet interface “ae0,” with a dual-tagged VLAN interface that has an outer tag of “111,” results in a `$junos-svlan-interface-set-name` dynamic variable of “ae0-111”. Similarly, a non-aggregated Ethernet interface of ge-1/1/0, with the same dual-tagged VLAN interface that has an outer tag of “111,” results in a `$junos-svlan-interface-set-name` dynamic variable of “ge-1/1/0-111”.
- The `$junos-phy-ifd-interface-set-name` predefined variable locally generates an interface set name associated with the underlying physical interface in a dynamic profile. This predefined variable enables you to group all the subscribers on a specific physical interface so that you can apply services to the entire group of subscribers.

Another use case for this predefined variable is to conserve CoS resources in a mixed business and residential topology by collecting the residential subscribers into an interface set associated with the physical interface, so that a level 2 node is used for the interface set rather than for each residential interface. Otherwise, because the business and residential subscribers share the same interface and business subscribers require three levels of CoS, then three levels are configured for each residential subscriber. That results in an unnecessary level 2 node being consumed for each residential connection, wasting CoS resources.

- The `$junos-tagged-vlan-interface-set-name` predefined variable locally generates an interface set name used for grouping logical interfaces stacked over logical stacked VLAN demux interfaces for either a 1:1 (dual-tagged; individual client) VLAN or N:1 (single tagged; service) VLAN. The format of the generated variable differs with VLAN type as follows:

- Dual-tagged (client) VLAN—*physical\_interface\_name - outer\_VLAN\_tag - inner\_VLAN\_tag*. For example, an aggregated Ethernet interface “ae0,” with a dual-tagged VLAN interface that has an outer tag of “111” and an inner tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ae0-200-111”. Similarly, a non-aggregated Ethernet interface of ge-1/1/0, with the same dual-tagged VLAN interface that has an outer tag of “111” and an inner tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ge-1/1/0-200-111”.
- Single tagged (service) VLAN—*physical\_interface\_name - VLAN\_tag*. For example, an aggregated Ethernet interface “ae0,” with an N:1 VLAN using the single tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ae0-200”. Similarly, a non-aggregated Ethernet interface of ge-1/1/0, with the same N:1 VLAN using the single tag of “200,” results in a `$junos-tagged-vlan-interface-set-name` dynamic variable of “ge-1/1/0-200”.
- All dynamic demux, dual-tagged VLAN logical interfaces with the same outer VLAN tag and physical interface are assigned to the same interface set and all CoS values provisioned with the dynamic profile are applied to the interfaces that are part of the set.
- The interface set name must be explicitly referenced in the CoS configuration as part of the static configuration outside of the dynamic profile. The CoS configuration is static and the interface set name must be statically referenced.



**NOTE:** This rule applies to all interface sets except ACI sets.

- RADIUS can return an *access-accept* message under certain conditions. A configured RADIUS VSA for the interface set name takes precedence over the locally generated variable on the router. This means that if the interface-set-name VSA is configured on RADIUS, the router continues to use this variable instead of the locally generated value from the dynamic variable.
- Sets of aggregated Ethernet interfaces are supported on MPC/MIC interfaces on MX Series routers only.
- The supported interface stacks for aggregated Ethernet in an interface set include VLAN demux interfaces, IP demux interfaces, and PPPoE logical interfaces over VLAN demux interfaces.
- The link membership list and scheduler mode of the interface set are inherited from the underlying aggregated Ethernet interface over which the interface set is configured.
- When an aggregated Ethernet interface operates in link protection mode, or if the scheduler mode is configured to replicate member links, the scheduling parameters of the interface set are copied to each of the member links.
- If the scheduler mode of the aggregated Ethernet interface is set to scale member links, the scheduling parameters are scaled based on the number of active member links and applied to each of the aggregated interface member links.

## RELATED DOCUMENTATION

*Configuring an Interface Set of Subscribers in a Dynamic Profile*

*Example: Configuring a Dynamic Service VLAN Interface Set of Subscribers in a Dynamic Profile*

## Configuring an Interface Set of Subscribers in a Dynamic Profile

Interface sets enable you to provide hierarchical scheduling to a group of subscriber interfaces.

Before you begin, configure the subscriber interfaces that you intend to include in the interface set.

To configure an interface set of subscriber interfaces:

1. Configure the interface set in the dynamic profile.

```
[edit dynamic-profiles profile-name interfaces]
user@host# edit interface-set interface-set-name
```

Replacing the *interface-set-name* variable with the \$junos-interface-set-name, \$junos-svlan-interface-set-name, or \$junos-tagged-vlan-interface-set-name predefined variable. The interface set is created dynamically when the subscriber logs in.

2. Include the interfaces within the dynamic interface-set.

```
[edit dynamic-profiles profile-name interfaces interface-set $junos-interface-set-name]
user@host# set interface interface-name unit logical-unit-number
```

3. Apply traffic shaping and queuing parameters to the interface set.



**TIP:** You must configure the interface set in the static [edit class-of-service] hierarchy, not in the [edit dynamic-profiles] hierarchy.

```
[edit class-of-service interfaces]
user@host# edit interface-set interface-set-name
[edit class-of-service interfaces interface-set interface-set-name]
user@host# set output-traffic-control-profile profile-name
```

## RELATED DOCUMENTATION

*CoS for Interface Sets of Subscribers Overview*

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

*CoS for Interface Sets of Subscribers Overview*

*Example: Configuring a Dynamic Interface Set of VLAN Subscribers*

*CoS for Aggregated Ethernet Subscriber Interfaces Overview*

## Example: Configuring a Dynamic Interface Set of VLAN Subscribers

### IN THIS SECTION

- [Requirements | 125](#)
- [Overview | 125](#)
- [Configuring the Dynamic VLANs | 126](#)
- [Configuring Dynamic Traffic Scheduling and Shaping | 129](#)
- [Configuring the Interface Set in the Dynamic Profile | 134](#)
- [Configuring DHCP Access | 137](#)
- [Configuring RADIUS Authentication | 139](#)
- [Verification | 145](#)

### Requirements

This example uses the following software and hardware components:

- MX Series Router with MPCs

### Overview

In this example, the network administrator groups dynamic VLAN interfaces in an interface set. The interface set is configured in a dynamic profile, and enables hierarchical scheduling for the VLAN interfaces for a multiplay service.

DHCP is used as the access method, and RADIUS is used as the authentication method for the interfaces associated with the interface set.

## Configuring the Dynamic VLANs

### IN THIS SECTION

- [CLI Quick Configuration | 126](#)
- [Configuring the Dynamic Profile for the Autoconfigured VLANs | 126](#)
- [Configuring the VLAN Interfaces | 127](#)
- [Configuring the Loopback Interface | 128](#)

### CLI Quick Configuration

To quickly configure the dynamic VLANs, copy the following commands and paste them into the router terminal window:

```
[edit]
edit dynamic-profiles vlan-prof
edit interfaces $junos-interface-ifd-name unit $junos-interface-unit
set vlan-id $junos-vlan-id
set demux-source inet
set family inet unnumbered-address lo0.0 preferred-source-address 203.0.113.32
top
edit interfaces ge-1/0/0
set hierarchical-scheduler
set vlan-tagging
edit auto-configure vlan-ranges dynamic-profile vlan-prof
set ranges any
set accept inet
top
set interfaces lo0 unit 0 family inet address 203.0.113.32/32
```

### Configuring the Dynamic Profile for the Autoconfigured VLANs

#### Step-by-Step Procedure

In this section, you create a dynamic profile for the VLAN IDs to be automatically assigned when subscribers log in.

To configure the dynamic profile for the VLANs:

1. Configure the dynamic profile.

```
[edit]
user@host#edit dynamic-profile vlan-prof
```

2. Configure the interfaces.

```
[edit dynamic-profiles vlan-prof]
user@host#edit interfaces $junos-interface-ifd-name unit $junos-interface-unit
```

3. Add the VLAN ID variable.

```
[edit dynamic-profiles vlan-prof interfaces $junos-interface-ifd-name unit $junos-interface-unit]
user@host#set vlan-id $junos-vlan-id
```

4. Configure the demux source as IPv4.

```
[edit dynamic-profiles vlan-prof interfaces $junos-interface-ifd-name unit $junos-interface-unit]
user@host#set demux-source inet
```

5. Configure the family.

```
[edit dynamic-profiles vlan-prof interfaces $junos-interface-ifd-name unit $junos-interface-unit]
user@host#set family inet unnumbered-address lo0.0 preferred-source-address 203.0.113.32
```

## Configuring the VLAN Interfaces

### Step-by-Step Procedure

To configure the VLAN interfaces:

1. Create the VLAN interface.

```
[edit]
user@host# edit interfaces ge-1/0/0
```

2. Enable hierarchical scheduling.

```
[edit interfaces ge-1/0/0]
user@host# set hierarchical-scheduler
```

3. Configure VLAN tagging.

```
[edit interfaces ge-1/0/0]
user@host# set vlan-tagging
```

4. Configure auto-configuration for the dynamic profile.

```
[edit interfaces ge-1/0/0]
user@host# edit auto-configure vlan-ranges dynamic-profile vlan-prof
```

5. Configure any VLAN ID range.

```
[edit interfaces ge-1/0/0 auto-configure vlan-ranges dynamic-profile vlan-prof]
user@host# set ranges any
```

6. Specify IPv4 traffic for the VLAN.

```
[edit interfaces ge-1/0/0 auto-configure vlan-ranges dynamic-profile vlan-prof]
user@host# set accept inet
```

## Configuring the Loopback Interface

### Step-by-Step Procedure

To configure the loopback interface:



1. Create the loopback interface.

```
[edit]
user@host# edit interfaces lo0
```

2. Configure the unit and the family.

```
[edit interfaces lo0]
user@host# set unit 0 family inet address 203.0.113.32/32
```

## Configuring Dynamic Traffic Scheduling and Shaping

### IN THIS SECTION

- [CLI Quick Configuration | 129](#)
- [Configuring the Schedulers in the Dynamic Profile | 131](#)
- [Configuring the Scheduler Map in the Dynamic Profile | 133](#)
- [Configuring the Traffic-Control Profile in the Dynamic Profile | 133](#)

### CLI Quick Configuration

To quickly configure the traffic scheduling and shaping parameters, copy the following commands and paste them into the router terminal window:

```
[edit]
edit dynamic-profiles multiplay class-of-service schedulers be_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit ef_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit af_sch
```

```
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit nc_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit voice_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit video_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit game_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up
edit data_sch
set transmit-rate percent 12
set buffer-size percent 12
set priority low
up 2
edit scheduler-maps all_smap
set forwarding-class be scheduler be_sch
set forwarding-class ef scheduler ef_sch
set forwarding-class af scheduler af_sch
set forwarding-class nc scheduler nc_sch
set forwarding-class voice scheduler voice_sch
set forwarding-class video scheduler video_sch
set forwarding-class game scheduler game_sch
set forwarding-class data scheduler data_sch
up 2
edit traffic-control-profiles multiplay
set scheduler-map all_smap
```

```
set shaping-rate 100m
set guaranteed-rate 20m
```

## Configuring the Schedulers in the Dynamic Profile

### Step-by-Step Procedure

In this section, you create a dynamic profile for the multiplay service and configure scheduling and shaping.

To configure the schedulers:

1. Create the multiplay dynamic profile.

```
[edit]
user@host# edit dynamic-profiles multiplay class-of-service schedulers
```

2. Configure the best effort scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit be_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

3. Configure the expedited forwarding scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit ef_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

4. Configure the assured forwarding scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit af_sch
user@host# set transmit-rate percent 12
```

```
user@host# set buffer-size percent 12
user@host# set priority low
```

5. Configure the network control scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit nc_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

6. Configure the voice scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit voice_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

7. Configure the video scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit video_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

8. Configure the gaming scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit game_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

## 9. Configure the data scheduler.

```
[edit dynamic-profiles multiplay class-of-service schedulers]
user@host# edit data_sch
user@host# set transmit-rate percent 12
user@host# set buffer-size percent 12
user@host# set priority low
```

## Configuring the Scheduler Map in the Dynamic Profile

### Step-by-Step Procedure

To configure the scheduler map:

1. Configure the scheduler map for all of the services.

```
[edit dynamic-profiles multiplay class-of-service]
user@host# edit scheduler-maps all_smap
```

2. Configure the forwarding classes for each service in the scheduler map.

```
[edit dynamic-profiles multiplay class-of-service scheduler-maps all_smap]
user@host# set forwarding-class be scheduler be_sch
user@host# set forwarding-class ef scheduler ef_sch
user@host# set forwarding-class af scheduler af_sch
user@host# set forwarding-class nc scheduler nc_sch
user@host# set forwarding-class voice scheduler voice_sch
user@host# set forwarding-class video scheduler video_sch
user@host# set forwarding-class game scheduler game_sch
user@host# set forwarding-class data scheduler data_sch
```

## Configuring the Traffic-Control Profile in the Dynamic Profile

### Step-by-Step Procedure

To configure the traffic-control profile the interface set:

1. Configure the traffic-control profile.

```
[edit dynamic-profiles multiplay class-of-service]
user@host# edit traffic control-profiles multiplay
```

2. Configure the scheduler map.

```
[edit dynamic-profiles multiplay class-of-service traffic control-profiles multiplay]
user@host# set scheduler-map all_smap
```

3. Configure the shaping rate.

```
[edit dynamic-profiles multiplay class-of-service traffic control-profiles multiplay]
user@host# set shaping-rate 100m
```

4. Configure the guaranteed rate.

```
[edit dynamic-profiles multiplay class-of-service traffic control-profiles multiplay]
user@host# set guaranteed-rate 20m
```

## Configuring the Interface Set in the Dynamic Profile

### IN THIS SECTION

- [CLI Quick Configuration | 135](#)
- [Configuring the Interfaces for the Interface Set | 135](#)
- [Configuring the Interface Set | 136](#)
- [Applying the Traffic-Control Profile to the Interface Set | 136](#)

## CLI Quick Configuration

To quickly configure the interface set, copy the following commands and paste them into the router terminal window:

```
[edit]
edit dynamic-profiles multiplay
edit interfaces interface-set $junos-interface-set-name
set interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
top
edit class-of-service interfaces interface-set
set output-traffic-control-profile multiplay
```

## Configuring the Interfaces for the Interface Set

### Step-by-Step Procedure

To configure the interface variable for the interface set:

1. Configure the dynamic profile for the interface set.

```
[edit]
user@host#edit dynamic-profiles multiplay
```

2. Configure the interface using the Junos OS predefined variable.

```
[edit dynamic-profiles multiplay]
user@host#edit interfaces $junos-interface-ifd-name unit $junos-underlying-interface-unit
```

3. Configure the family.

```
[edit dynamic-profiles multiplay interfaces $junos-interface-set-name unit $junos-underlying-interface-unit]
user@host#set family inet unnumbered-address lo0.0 preferred-source-address 203.0.113.32
```

## Configuring the Interface Set

### Step-by-Step Procedure

To configure the interface set:

1. Configure the interface set using the Junos OS predefined variable.

```
[edit dynamic-profiles multiplay]
user@host#edit interfaces interface-set $junos-interface-set-name
```

2. Add the dynamic VLAN interfaces to the interface set.

```
[edit dynamic-profiles multiplay interfaces $junos-interface-set-name]
user@host#set interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
```

## Applying the Traffic-Control Profile to the Interface Set

### Step-by-Step Procedure

You apply the traffic-control profile outside of the dynamic profile in the [edit class-of-service] hierarchy.

To apply the traffic-control profile:

1. Specify the interface set to which you want to apply the traffic-control profile.

```
[edit class-of-service]
user@host#edit interfaces interface-set dynamic-set
```

2. Attach the output traffic-control profile defined in the dynamic profile to the interface set.

```
[edit class-of-service interfaces]
user@host#set output-traffic-control-profile multiplay
```



## Configuring DHCP Access

### IN THIS SECTION

- [CLI Quick Configuration | 137](#)
- [Configuring the DHCP Local Server | 137](#)
- [Configuring Address Assignment Pools | 138](#)

### CLI Quick Configuration

To quickly configure DHCP access, copy the following commands and paste them into the router terminal window:

```
[edit]
edit system services dhcp-local-server authentication
set password $ABC123
set username-include user-prefix multiplay
up 1
set dynamic-profile dhcp-vlan-prof aggregate-clients replace
set group vlans interface ge-1/0/0
top
edit access address-assignment pool v4 family inet
set network 203.0.113.0/16
set range limited low 203.0.113.10
set range limited high 203.0.113.250
set dhcp-attributes maximum-lease-time 84600
```

### Configuring the DHCP Local Server

#### Step-by-Step Procedure

To configure DHCP access:

1. Configure the DHCP local server.

```
[edit system]
user@host# edit services dhcp-local-server authentication
```

2. Set the password.

```
[edit system services dhcp-local-server authentication]
user@host# set password $ABC123
```

3. Specify that you want to include optional information in the username.

```
[edit system services dhcp-local-server authentication]
user@host# set username-include user-prefix multiplay
```

4. Attach the dynamic profile with the interface set.

```
[edit system services dhcp-local-server]
user@host# set dynamic-profile dhcp-vlan-prof aggregate-clients replace
```

5. Configure a group for the VLAN interface.

```
[edit system services dhcp-local-server]
user@host# set group vlans interface ge-1/0/0
```

## Configuring Address Assignment Pools

### Step-by-Step Procedure

To configure address assignment pools:

1. Configure the pool of IPv4 addresses.

```
[edit access]
user@host#edit address-assignment pool v4 family inet
```

2. Configure the family of interfaces in the pool.

```
[edit access address-assignment pool v4]
user@host#set network 203.0.113.0/16
```

3. Configure the upper and lower bounds of the address range.

```
[edit access address-assignment pool v4]
user@host#set range limited low 203.0.113.10
user@host#set range limited high 203.0.113.250
```

4. Configure the maximum length of time in seconds for which a subscriber can request and hold a lease.

```
[edit access address-assignment pool v4]
user@host#set dhcp-attributes maximum-lease-time 84600
```

## Configuring RADIUS Authentication

### IN THIS SECTION

- [CLI Quick Configuration | 139](#)
- [Configuring RADIUS Access | 140](#)
- [Results | 141](#)

### CLI Quick Configuration

To quickly configure RADIUS authentication, copy the following commands and paste them into the router terminal window:

```
[edit]
edit access radius-server 192.51.100.108
set secret $ABC123ABC123ABC123
set timeout 5
set retry 5
up 2
edit profile acc-prof
set authentication-order radius
set radius authentication-server 192.51.100.108
```

## Configuring RADIUS Access

### Step-by-Step Procedure

To configure RADIUS access:

1. Configure the RADIUS server.

```
[edit access]
user@host#edit radius-server 192.51.100.108
```

2. Configure the required secret (password) that the local router or switch passes to the RADIUS client.

```
[edit access radius-server 192.51.100.108]
user@host# set secret $ABC123ABC123ABC123
```

3. Configure the length of time that the local router or switch waits to receive a response from a RADIUS server.

```
[edit access radius-server 192.51.100.108]
user@host# set timeout 5
```

4. Configure the number of times that the router or switch attempts to contact a RADIUS accounting server.

```
[edit access radius-server 192.51.100.108]
user@host# set retry 5
```

5. Configure the access profile.

```
[edit access]
user@host#edit profile acc-prof
```

6. Configure the authentication order.

```
[edit access profile acc-prof ]
user@host# set authentication-order radius
```

## 7. Configure the authentication server.

```
[edit access profile acc-prof]
user@host#set radius authentication-server 192.51.100.108
```

## Results

```
dynamic-profiles {
  vlan-prof {
    interfaces {
      "$junos-interface-ifd-name" {
        unit "$junos-interface-unit" {
          vlan-id "$junos-vlan-id";
          demux-source inet;
          family inet {
            unnumbered-address lo0.0 preferred-source-address 203.0.113.32;
          }
        }
      }
    }
  }
}
multiplay {
  class-of-service {
    traffic-control-profiles {
      multiplay {
        scheduler-map all_smap;
        shaping-rate 100m;
        guaranteed-rate 20m;
      }
    }
    interfaces {
      interface-set "$junos-interface-set-name" {
        interface "$junos-interface-ifd-name" {
          unit "$junos-underlying-interface-unit";
        }
      }
      "$junos-interface-ifd-name" {
        unit "$junos-interface-unit" {
          output-traffic-control-profile multiplay;
        }
      }
    }
  }
}
```

```
}
scheduler-maps {
    all_smap {
        forwarding-class be scheduler be_sch;
        forwarding-class ef scheduler ef_sch;
        forwarding-class af scheduler af_sch;
        forwarding-class nc scheduler nc_sch;
        forwarding-class voice scheduler voice_sch;
        forwarding-class video scheduler video_sch;
        forwarding-class game scheduler game_sch;
        forwarding-class data scheduler data_sch;
    }
}
schedulers {
    be_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
    ef_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
    af_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
    nc_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
    voice_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
    video_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
}
```

```
    }
    game_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
    data_sch {
        transmit-rate percent 12;
        buffer-size percent 12;
        priority low;
    }
}
}
}
}
access {
    radius-server {
        192.51.100.108 {
            secret "$ABC123ABC123ABC123"; ## SECRET-DATA
            timeout 5;
            retry 5;
        }
    }
    profile acc-prof {
        authentication-order radius;
        radius {
            authentication-server 192.51.100.108;
        }
    }
    address-assignment {
        pool v4 {
            family inet {
                network 203.0.113.0/16;
                range limited {
                    low 203.0.113.10;
                    high 203.0.113.250;
                }
                dhcp-attributes {
                    maximum-lease-time 84600;
                }
            }
        }
    }
}
}
```

```

class-of-service {
  interfaces {
    interface-set dynamic-set {
      output-traffic-control-profile multiplay;
    }
  }
}
interfaces {
  interface-set "$junos-interface-set-name" {
    interface "$junos-interface-ifd-name" {
      unit "$junos-underlying-interface-unit";
    }
  }
  "$junos-interface-ifd-name" {
    unit "$junos-underlying-interface-unit" {
      family inet {
        unnumbered-address lo0.0 preferred-source-address 203.0.113.32;
      }
    }
  }
}
}
}
interfaces {
  ge-1/0/0 {
    hierarchical-scheduler;
    vlan-tagging;
    auto-configure {
      vlan-ranges {
        dynamic-profile vlan-prof {
          accept inet;
          ranges {
            any;
          }
        }
      }
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 203.0.113.32/32;
    }
  }
}
}

```



```
    }
  }
}
system {
  services {
    dhcp-local-server {
      authentication {
        password $ABC123;
        username-include {
          user-prefix multiplay;
        }
      }
      dynamic-profile multiplay aggregate-clients replace;
      group vlans {
        interface ge-1/0/0.0;
      }
    }
  }
}
```

## Verification

### IN THIS SECTION

- [Verifying the Interfaces that are Included in the Interface Set | 145](#)
- [Verifying the Traffic Scheduling and Shaping Parameters for the Interface Set | 146](#)

To confirm that the configuration is correct, perform these tasks:

### Verifying the Interfaces that are Included in the Interface Set

#### Purpose

Verify the interfaces included in the interface set.

## Action

```
user@host> show interfaces interface-set dynamic-set terse
```

### Verifying the Traffic Scheduling and Shaping Parameters for the Interface Set

## Purpose

Verify that the traffic scheduling and shaping parameters are applied properly to an interface included in the interface set.

## Action

```
user@host> show class-of-service interface
```

## RELATED DOCUMENTATION

[Understanding Hierarchical CoS for Subscriber Interfaces](#)

*Configuring an Interface Set of Subscribers in a Dynamic Profile*

## Example: Configuring a Dynamic Service VLAN Interface Set of Subscribers in a Dynamic Profile

### IN THIS SECTION

- [Requirements | 147](#)
- [Overview | 147](#)
- [Configuration | 148](#)
- [Verification | 151](#)

Interface sets enable you to provide hierarchical scheduling to a group of subscriber interfaces. In this example, by using the `$junos-svlan-interface-set-name` internal dynamic variable when specifying the interface set name, you can locally generate an interface set name for use by SVLAN interfaces based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical\_interface\_name - outer\_VLAN\_tag*.

## Requirements

Before you begin, configure the subscriber interfaces that you intend to include in the interface set. You can find general configuration instructions for the supported dynamic interface configuration in *DHCP Subscriber Interface Overview* and in the following:

- For dynamic VLAN interfaces, see *Configuring a Static or Dynamic VLAN Subscriber Interface over Aggregated Ethernet*.
- For dynamic IP demux interfaces, see *Configuring Dynamic Subscriber Interfaces Using IP Demux Interfaces in Dynamic Profiles* and *Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet*.
- For dynamic VLAN demux interfaces, see *Configuring Dynamic Subscriber Interfaces Using VLAN Demux Interfaces in Dynamic Profiles*.

## Overview

Interface sets enable you to provide hierarchical scheduling to a group of subscriber interfaces. By using the `$junos-svlan-interface-set-name` internal dynamic variable when specifying the interface set name, you can locally generate an interface set name for use by SVLAN interfaces based on the outer tag of the dual-tagged VLAN. The format of the generated variable is *physical\_interface\_name - outer\_VLAN\_tag*.

This example includes the following statements:

- `interface-set`—Configures the name of the scheduler for dynamic CoS. In this example, you use the `$junos-svlan-interface-set-name` variable to obtain the locally generated interface set name for use by SVLAN interfaces based on the outer tag of the dual-tagged VLAN.
- `output-traffic-control-profile`—Applies an output traffic scheduling and shaping profile to the interface set.
- `output-traffic-control-profile-remaining`—Applies an output traffic scheduling and shaping profile for remaining traffic to the interface set.

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 148](#)
- [Procedure | 148](#)
- [Results | 150](#)

### CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
[edit]
set dynamic-profiles profile-dhcp-ipdemux interfaces interface-set $junos-svlan-interface-set-
name interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
set dynamic-profiles profile-dhcp-ipdemux interfaces $junos-interface-ifd-name unit $junos-
underlying-interface-unit
set class-of-service traffic-control-profiles tcp1 scheduler-map schedMap
set class-of-service traffic-control-profiles tcp1 shaping-rate 50m
set class-of-service traffic-control-profiles tcp1 guaranteed-rate 200k
set class-of-service traffic-control-profiles tcp3 scheduler-map ss1q0q1
set class-of-service traffic-control-profiles tcp3 shaping-rate 20m
set class-of-service traffic-control-profiles tcp3 guaranteed-rate 5m
set class-of-service interfaces interface-set ae0-111 output-traffic-control-profile tcp1
set class-of-service interfaces interface-set ae0-111 output-traffic-control-profile-remaining
tcp3
```

### Procedure

#### Step-by-Step Procedure

To configure an SVLAN interface set of subscriber interfaces:

1. Access the dynamic profile you want to modify for interface sets.

```
[edit]
user@host# edit dynamic-profiles profile-dhcp-ipdemux
```

2. Access the dynamic profile interface configuration.

```
[edit dynamic-profiles profile-dhcp-ipdemux]
user@host# edit interfaces
```

3. Configure the SVLAN interface set in the dynamic profile.

The interface set is created dynamically when the subscriber logs in.

```
[edit dynamic-profiles profile-dhcp-ipdemux interfaces]
user@host# edit interface-set $junos-svlan-interface-set-name
```

4. Include dynamic IP demux interface creation within the dynamic interface set.

```
[edit dynamic-profiles profile-dhcp-ipdemux interfaces interface-set $junos-svlan-interface-set-name]
user@host# set interface $junos-interface-ifd-name unit $junos-underlying-interface-unit
```

5. Access the SVLAN interface set name that you expect \$junos-svlan-interface-set-name to generate. For example, to specify the expected interface set name for aggregated Ethernet interface ae0 and outer VLAN tag 111, include **ae0-111** for the *interface-set-name* variable.

```
[edit class-of-service interfaces]
user@host# edit interface-set ae0-111
```

6. Apply traffic shaping and queuing parameters to the SVLAN interface set.



**TIP:** You must configure the interface set in the static [edit class-of-service] hierarchy, not in the [edit dynamic-profiles] hierarchy.

```
[edit class-of-service interfaces interface-set ae0-111]
user@host# set output-traffic-control-profile tcp1
```

## 7. Apply traffic shaping and queuing parameters to any remaining traffic on the SVLAN interface set.

```
[edit class-of-service interfaces interface-set ae0-111]
user@host# set output-traffic-control-profile-remaining tcp3
```

## Results

From configuration mode, confirm your configuration by entering the `show dynamic-profiles` command and the `show class-of-service` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show dynamic-profiles
dynamic-profiles {
  profile-dhcp-ipdemux {
    interfaces {
      interface-set "$junos-svlan-interface-set-name" {
        interface "$junos-interface-ifd-name" {
          unit "$junos-underlying-interface-unit";
        }
      }
      "$junos-interface-ifd-name" {
        unit "$junos-underlying-interface-unit";
      }
    }
  }
}
```

```
user@host# show class-of-service
class-of-service {
  traffic-control-profiles {
    tcp1 {
```

```
    scheduler-map schedMap;
    shaping-rate 50m;
    guaranteed-rate 200k;
  }
  tcp3 {
    inactive: scheduler-map ss1q0q1;
    shaping-rate 20m;
    guaranteed-rate 5m;
  }
}
interfaces {
  interface-set ae0-111 {
    output-traffic-control-profile tcp1;
    output-traffic-control-profile-remaining tcp3;
  }
}
}
```

## Verification

### IN THIS SECTION

- [Verifying the Interfaces that are Included in the Interface Set | 151](#)
- [Displaying Information for Active Subscribers | 152](#)

To confirm that the configuration is correct, perform these tasks:

### Verifying the Interfaces that are Included in the Interface Set

#### Purpose

Verify the interfaces that are included in the interface set.

#### Action

```
user@host> show class-of-service interface-set
```

## Displaying Information for Active Subscribers

### Purpose

Display information for active subscribers.

### Action

```
user@host> show subscribers detail
```

## RELATED DOCUMENTATION

---

*[Dynamic Profiles Overview](#)*

---

*[Configuring a Basic Dynamic Profile](#)*

---

*[Configuring Hierarchical Schedulers for CoS](#)*

---

*[Configuring Remaining Common Queues on MIC and MPC Interfaces](#)*



# Configuring Hierarchical Scheduling for MPLS Pseudowire Interfaces

## IN THIS CHAPTER

- [Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | 153](#)
- [CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | 154](#)
- [CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 156](#)
- [Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces | 158](#)
- [CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)
- [Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Transport Logical Interface\) | 165](#)
- [Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Pseudowire Interface Set\) | 167](#)

## Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview

Junos OS supports two aspects of CoS for MPLS pseudowire subscriber interfaces. You can apply CoS *rewrite rules* and behavior aggregate (BA) classifiers to MPLS pseudowire subscriber interfaces. In addition, CoS performs egress hierarchical shaping towards the subscriber on MPLS pseudowire subscriber interfaces.

Hierarchical CoS enables you to apply traffic scheduling and queuing parameters and packet transmission scheduling parameters to an individual subscriber interface rather than to all interfaces configured on the port. Hierarchical CoS is supported on MX Series routers with either EQ DPCs or MPC/MICs installed.

On Juniper Networks MX Series routers, MPC/MIC and EQ DPC interfaces support a four-level CoS scheduling hierarchy that, when fully configured, consists of the physical interface (level 1), the interface set or the underlying interface (level 2), one or more logical interfaces (level 3), and one or more queues (level 4). Although all CoS scheduling hierarchies are four-level, level 1 is always the physical interface and level 4 is always the queue. Hierarchical scheduling configurations consist of the type of interfaces you configure; for example, a *logical interface* or an interface set and where those interfaces reside in

the scheduling hierarchy, either level 2 or level 3. Because many hierarchical scheduling configurations are possible, we use the terms *two-level hierarchical scheduling* and *three-level hierarchical scheduling* in this discussion.

## RELATED DOCUMENTATION

*Pseudowire Subscriber Logical Interfaces Overview*

*Configuring a Pseudowire Subscriber Logical Interface*

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

[CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 156](#)

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)

[CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | 154](#)

*hierarchical-scheduler*

## CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces

CoS supports two-level and three-level hierarchies for MPLS pseudowire subscriber interfaces.

To configure two-level scheduling, include the `maximum-hierarchy-levels 2` option under the `[edit interfaces interface-name hierarchical-scheduler]` statement on the physical interface of the logical tunnel anchor point.

To configure three-level hierarchical scheduling, include the `implicit-hierarchy` option under the `[edit interfaces interface-name hierarchical-scheduler]` statement on the physical interface of the logical tunnel anchor point. Use the following guidelines for configuring the `implicit-hierarchy` option:

- If an output traffic-control profile is configured on the pseudowire transport interface and on a pseudowire service interface, the two interfaces form a scheduling hierarchy. The pseudowire transport interface resides in a level 2 scheduler node and the pseudowire service interface resides in a level 3 scheduler node.
- If an output traffic-control profile is configured on the pseudowire services interface but not on a pseudowire transport interface, the pseudowire services interface resides in a level 3 scheduler node.
- If an output traffic-control profile is only configured on the pseudowire transport interface and not on the pseudowire services interface, the pseudowire transport interface resides in a level 3 scheduler node and all pseudowire traffic uses this node.

If the `implicit-hierarchy` option is not set on the logical tunnel anchor point, logical interfaces behave normally with the `hierarchical-scheduler` mode configured with or without the `hierarchical-scheduler`

maximum-hierarchy-levels option under the [edit interfaces *interface-name* hierarchical-scheduler] statement. In this case, when you apply a traffic-control profile to the pseudowire transport and service logical interfaces, they both reside in level 3 scheduler nodes and do not form a scheduling hierarchy, which might not be the desirable behavior. In business edge, where only the pseudowire logical interfaces need to be shaped, applying the traffic-control profile at just the transport *logical interface* may be sufficient.

When configuring the logical tunnel physical interface for the maximum hierarchy level, all pseudowire logical interfaces operating on the physical interface use the same hierarchy model. If you want to mix two-level and three-level scheduling hierarchies, you can group the pseudowires together by hierarchy levels and share the same logical tunnel anchor point or you can use three-level scheduling for all pseudowires over the anchor point.

To specify *rewrite rules* and classifiers on pseudowire interfaces, reference the pseudowire device under the [edit class-of-service interfaces] hierarchy level and specify the rewrite rules and classifiers for the pseudowire interfaces.

To control all pseudowire traffic using the same logical tunnel interface, apply CoS policies at the physical interface for the anchor logical tunnel.



**NOTE:** Starting with Junos OS Release 17.3R1, stateful anchor point redundancy support is provided for pseudowire subscriber logical interface by the underlying redundant logical tunnel interface (rlt) in active-backup mode. This redundancy protects the access and the core facing link against anchor PFE (Packet Forwarding Engine) failure. Starting in Junos OS Release 18.1R2, full hierarchical CoS support is provided for stateful anchor point redundancy of pseudowire subscriber logical interfaces. Both transport and services logical interfaces created for the pseudowire subscriber logical interface are stacked on the underlying redundant logical tunnel control logical interface. This logical interface stacking model is used for both redundant and non-redundant pseudowire subscriber logical interfaces. Hierarchical CoS is supported and configured the same on both redundant and non-redundant pseudowire subscriber logical interfaces.

### Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
18.1R2	Starting in Junos OS Release 18.1R2, full hierarchical CoS support is provided for stateful anchor point redundancy of pseudowire subscriber logical interfaces.

## RELATED DOCUMENTATION

[Pseudowire Subscriber Logical Interfaces Overview](#)

[Configuring a Pseudowire Subscriber Logical Interface](#)

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

[Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | 153](#)

[Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces | 158](#)

[Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Transport Logical Interface\) | 165](#)

[Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Pseudowire Interface Set\) | 167](#)

[Anchor Redundancy Pseudowire Subscriber Logical Interfaces Overview](#)

[hierarchical-scheduler](#)

## CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces

Two-level hierarchical scheduling limits the number of hierarchical levels in the scheduling hierarchy to two. In a two-level scheduling hierarchy, all logical interfaces and interface sets share a single level 2 node. [Table 13 on page 156](#) summarizes the interface hierarchy and the CoS scheduler node levels for two-level hierarchical scheduling.

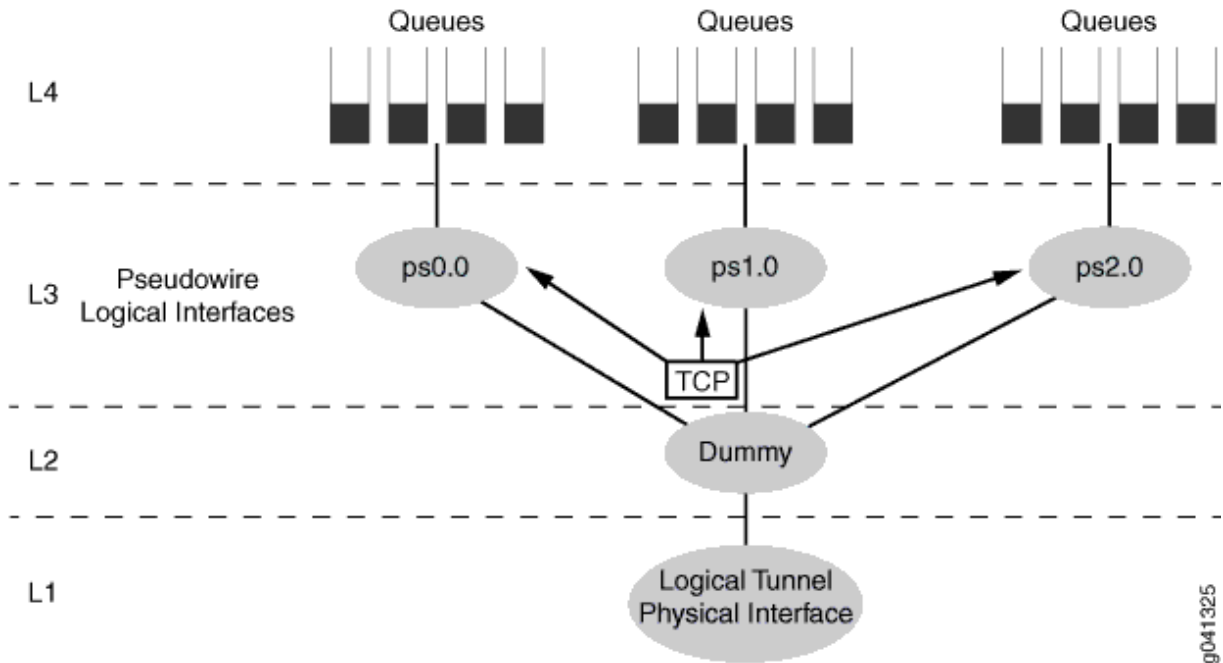
**Table 13: Two-Level Hierarchical Scheduling–Interface Hierarchy Versus Scheduling Nodes**

Level 1	Level 2	Level 3	Level 4
Physical interface	–	Pseudowire transport <i>logical interface</i>	One or more queues
Physical interface	–	Interface set	One or more queues
Physical interface	–	Pseudowire service logical interface	One or more queues

You use the two-level hierarchical scheduling when you have many pseudowires but you do not require shaping specific to the subscriber logical interface. For example, when your configuration is one subscriber per pseudowire interface.

Figure 17 on page 157 shows a two-level hierarchical scheduling configuration for the MPLS pseudowires. In this configuration, level 1 is the physical interface used for the logical tunnel anchor node. All of the pseudowire transport interfaces share a single level 2 node. The level 3 nodes are the pseudowire transport logical interfaces (ps0.0, ps1.0, and ps2.0). In this configuration, interface sets are not configured and only the logical interfaces have traffic control profiles.

Figure 17: MPLS Pseudowire Subscriber Interface Two-Level Scheduler Configuration



Two-level hierarchical scheduling has up to eight *class of service* queues. For this configuration, include the `maximum-hierarchy-levels 2` option under the `[edit interfaces interface-name hierarchical-scheduler]` hierarchy level at the physical interface for the anchor logical tunnel.



**NOTE:** You cannot configure shaping policies on both the pseudowire logical interfaces and the subscriber logical interfaces over the same pseudowire. If a traffic-control profile is configured on a pseudowire logical interface, and CoS policies are configured on the subscriber logical interface over another pseudowire, all of the logical interfaces are at level 3 and act as peers.

## RELATED DOCUMENTATION

[Pseudowire Subscriber Logical Interfaces Overview](#)

[Configuring a Pseudowire Subscriber Logical Interface](#)

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

[Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | 153](#)

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)

[CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | 154](#)

[Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces | 158](#)

*hierarchical-scheduler*

## Configuring CoS Two-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces

Before configuring CoS parameters for MPLS pseudowire subscriber interfaces, you must first complete these tasks:

1. Configure the pseudowire logical interfaces. See *Configuring a Pseudowire Subscriber Logical Interface*.
2. Configure the pseudowire device count. See *Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router*.
3. Configure the pseudowire device including the logical tunnel anchor point. See *Configuring a Pseudowire Subscriber Logical Interface Device*.
4. Configure the pseudowire transport logical interface. See *Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface*.
5. Configure the pseudowire signaling (either Layer 2 circuit signaling or Layer 2 VPN signaling). See *Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces* or *Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces*.
6. Configure the pseudowire logical interfaces. See *Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface*.

To configure CoS policies on MPLS pseudowire subscriber interfaces using two-level scheduling:

1. Configure the hierarchical scheduler for the physical interface used for the logical tunnel (anchor point). For two-level scheduling the hierarchical scheduler must be set to `maximum-scheduler levels 2`.

```
[edit]
user@host#edit interfaces ps-anchor-device-name
user@host#set hierarchical-scheduler maximum-hierarchy-levels 2
```

2. Specify the traffic-control profile to use on the pseudowire logical interface.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#set output-traffic-control-profile profile-name
```

3. Configure the rewrite rule.

The available rewrite rule types for pseudowire interfaces are `dscp` and `inet-precedence`.

```
[edit class-of-service]
user@host#edit rewrite-rules (dscp | inet-precedence) rewrite-name
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-point (alias | bits)
```

4. Configure the classifier.

The available classifier types for pseudowire interfaces are `dscp` and `inet-precedence`.

```
[edit class-of-service]
user@host#edit classifiers (dscp | inet-precedence) classifier-name
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-points [aliases] [bit-patterns]
```

5. Apply the rewrite rule and classifier to the pseudowire interface.

For the `interface_name` parameter, specify the pseudowire device name.

```
[edit class-of-service interfaces interface_name unit logical-unit-number]
user@host#set rewrite-rule (dscp | inet-precedence) (rewrite-name | default) protocol
protocol-types
user@host#set classifiers (dscp | inet-precedence) (classifier-name | default)
```

## RELATED DOCUMENTATION

*CoS on Ethernet Pseudowires in Universal Edge Networks Overview*

*Mapping CoS Component Inputs to Outputs*

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

[Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | 153](#)

[CoS Two-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 156](#)

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)

[CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | 154](#)

*hierarchical-scheduler*

## CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces

### IN THIS SECTION

- [Three-Level Scheduling Hierarchy: Pseudowire Logical Interfaces over a Transport Logical Interface | 161](#)
- [Three-Level Scheduling Hierarchy : Pseudowire Service Logical Interfaces over a Pseudowire Service Interface Set | 162](#)
- [Three-Level Scheduling Hierarchy Combined Deployment Scenario | 163](#)

In three-level hierarchical scheduling, the CoS scheduler nodes at level 1, level 2, and level 3 form a scheduling hierarchy. You can configure many different three-level scheduling hierarchies, depending on the location of the interface set and the use of underlying interfaces. In all variations, the physical interface on which the logical tunnel resides is a level 1 CoS scheduler node and the queues reside at level 4. Three-level scheduling hierarchies can have up to eight *class of service* queues. [Table 14 on page 161](#) summarizes the most common three-level hierarchical scheduling configurations and shows the interface hierarchy and CoS scheduler nodes.



**Table 14: Three-Level Hierarchical Scheduling–Interface Hierarchy Versus CoS Scheduling Node Levels**

Level 1	Level 2	Level 3	Level 4
Physical interface	Pseudowire interface set	Pseudowire service logical interfaces	One or more queues
Physical interface	Pseudowire transport <i>logical interface</i>	Pseudowire interface set	One or more queues
Physical interface	Pseudowire transport logical interface	Pseudowire service logical interfaces	One or more queues

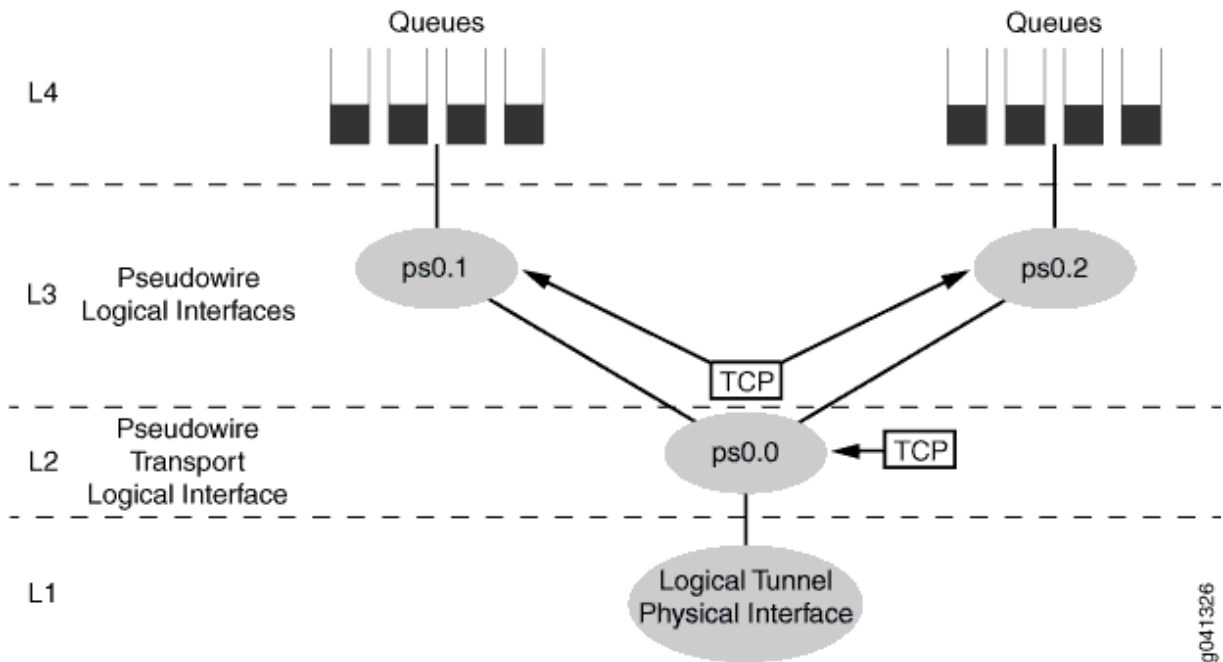
### Three-Level Scheduling Hierarchy: Pseudowire Logical Interfaces over a Transport Logical Interface

Figure 18 on page 162 shows an MPLS pseudowire three-level scheduling hierarchy that includes two pseudowire service logical interfaces over a pseudowire transport logical interface. This variation uses the following scheduler nodes:

- Level 4—Forwarding class-based queues
- Level 3—Pseudowire service logical interfaces (ps0.1 and ps0.2) for subscriber sessions
- Level 2—Pseudowire transport logical interface (ps0.0)
- Level 1—Common/shared physical interface of the logical tunnel anchor point

You apply the traffic-control profiles at the pseudowire transport logical interfaces (level 2) and the pseudowire service logical interfaces (level 3).

Figure 18: Three-Level Scheduling Hierarchy Case 1: Pseudowire Service Logical Interfaces over a Transport Logical Interface



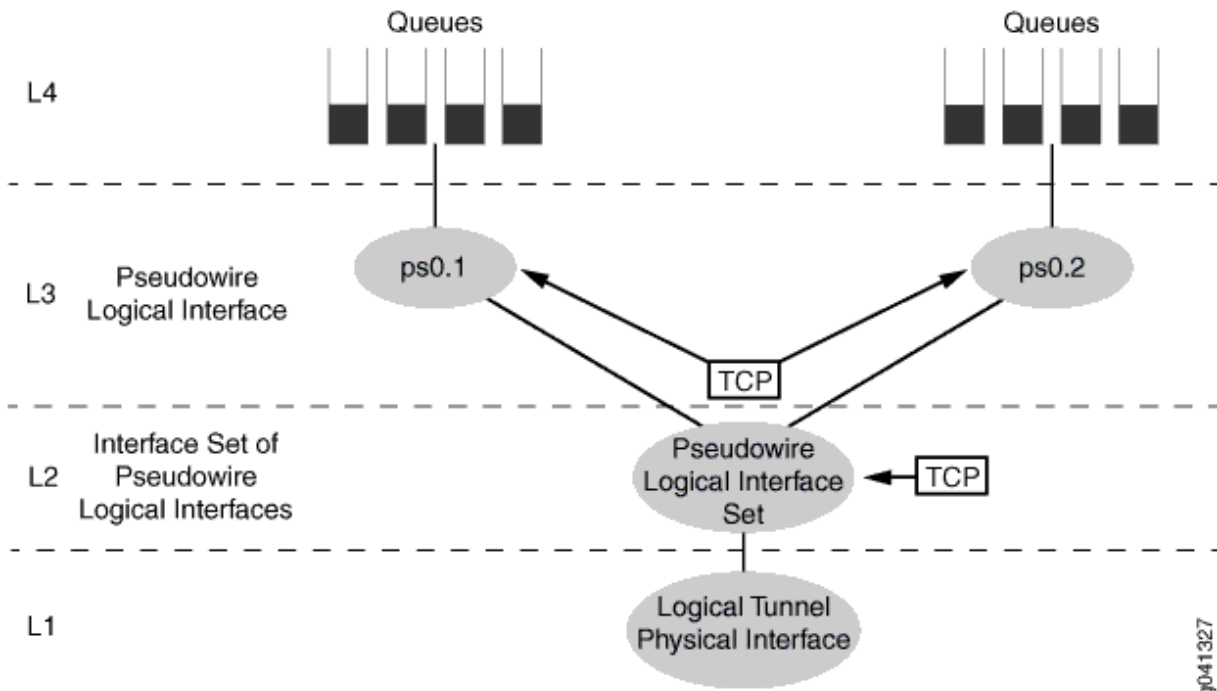
### Three-Level Scheduling Hierarchy : Pseudowire Service Logical Interfaces over a Pseudowire Service Interface Set

Figure 19 on page 163 shows another variation of MPLS pseudowire three-level hierarchical scheduling that includes two pseudowire service logical interfaces over a pseudowire service interface set. This variation uses the following CoS scheduler nodes:

- Level 4—Forwarding class-based queues
- Level 3—Pseudowire service logical interfaces (ps0.1 and ps0.2)
- Level 2—Pseudowire service interface set
- Level 1—Common/shared physical interface of the logical tunnel anchor point

You apply the traffic-control profile at the pseudowire service interfaces (level 3) and at the interface set (level 2). This variation is most useful for subscriber edge deployments.

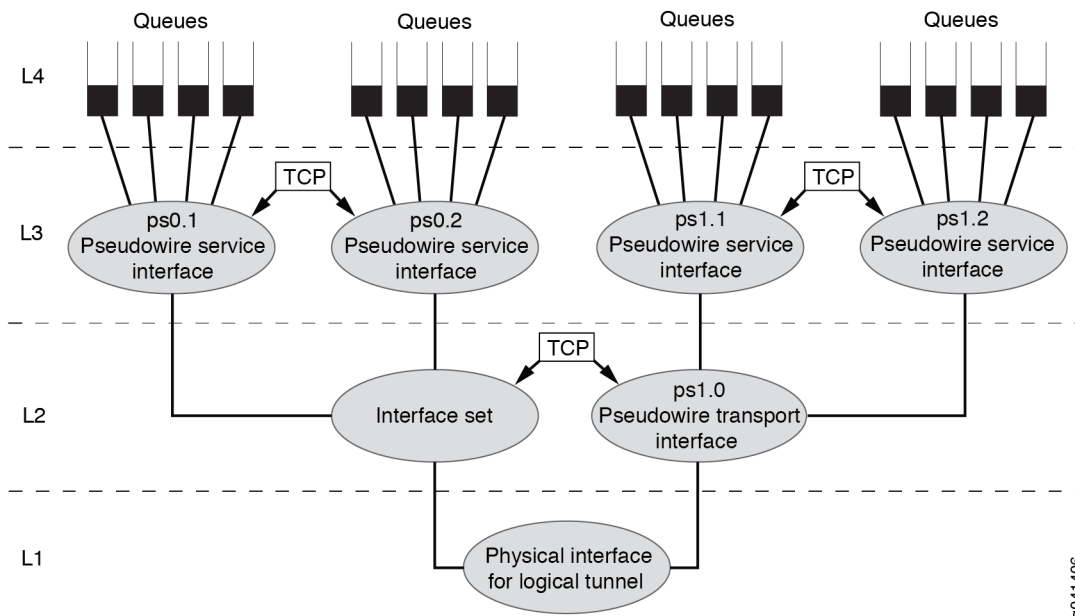
Figure 19: Three-Level Scheduling Hierarchy Case 2: Pseudowire Service Logical Interfaces over a Pseudowire Service Interface Set



### Three-Level Scheduling Hierarchy Combined Deployment Scenario

Figure 20 on page 164 shows a deployment scenario that combines the three-level hierarchical scheduling scenarios in Figure 18 on page 162 and Figure 19 on page 163.

**Figure 20: Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces—Deployment Scenario**



This variation uses the following CoS scheduler nodes:

- Level 4—Forwarding class-based queues
- Level 3—Pseudowire service logical interfaces (ps0.1, ps0.2, ps1.1, and ps1.2)
- Level 2—Service interface set for pseudowire service interfaces (ps0.1 and ps0.2) and transport logical interface (ps1.0) for the pseudowire service logical interfaces (ps1.1 and ps1.2)
- Level 1—Common/shared physical interface of the logical tunnel anchor point

You apply the traffic-control profiles to the interfaces at both level 2 and level 3, as well as the interface set at level 2.

## RELATED DOCUMENTATION

*Pseudowire Subscriber Logical Interfaces Overview*

*Configuring a Pseudowire Subscriber Logical Interface*

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

[Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | 153](#)

[CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | 154](#)

[Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Transport Logical Interface\) | 165](#)

[Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Pseudowire Interface Set\) | 167](#)

*hierarchical-scheduler*

## Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Transport Logical Interface)

Before configuring CoS three-level scheduling on pseudowire logical interfaces over a transport logical interface, you must first complete these tasks:

1. Configure the pseudowire logical interfaces. See *Configuring a Pseudowire Subscriber Logical Interface*.
2. Configure the pseudowire device count. See *Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router*.
3. Configure the pseudowire device including the logical tunnel anchor point. See *Configuring a Pseudowire Subscriber Logical Interface Device*.
4. Configure the pseudowire transport logical interface. See *Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface*.
5. Configure the pseudowire signaling (either Layer 2 circuit signaling or Layer 2 VPN signaling). See *Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces* or *Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces*.
6. Configure the pseudowire logical interfaces. See *Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface*.

Three-level scheduling on pseudowire logical interfaces over a transport logical interface requires you to apply the traffic-control profiles at both the pseudowire logical interface and the pseudowire transport logical interface. To configure CoS policies on three-level scheduling on pseudowire logical interfaces over a transport logical interface:

1. Configure the hierarchical scheduler for the physical interface used for the logical tunnel (anchor point). For three-level scheduling the hierarchical scheduler must be set to `implicit-hierarchy`.

```
[edit]
user@host#edit interfaces ps-anchor-device-name
user@host#set hierarchical-scheduler implicit-hierarchy
```

2. Specify the traffic-control profile to use on the pseudowire logical interface.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#set output-traffic-control-profile profile-name
```

3. Specify the traffic-control profile to use on the pseudowire transport logical interface.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#set output-traffic-control-profile profile-name
```

4. Configure the rewrite rule.

The available rewrite rule types for pseudowire interfaces are `dscp` and `inet-precedence`.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#edit rewrite-rules (dscp | inet-precedence) rewrite-name
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-point (alias | bits)
```

5. Configure the classifier.

The available classifier types for pseudowire interfaces are `dscp` and `inet-precedence`.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#edit classifiers (dscp | inet-precedence) classifier-name
```

```
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-points [aliases] [bit-patterns]
```

6. Apply the rewrite rule and classifier to the pseudowire interfaces.

For the *interface\_name* parameter, specify the pseudowire device name.

```
[edit class-of-service interfaces interface_name unit logical-unit-number]
user@host#set rewrite-rule (dscp | inet-precedence) (rewrite-name | default) protocol
protocol-types
user@host#set classifiers (dscp | inet-precedence) (classifier-name | default)
```

## RELATED DOCUMENTATION

[CoS on Ethernet Pseudowires in Universal Edge Networks Overview](#)

[Mapping CoS Component Inputs to Outputs](#)

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

[Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | 153](#)

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)

[CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | 154](#)

[Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Pseudowire Interface Set\) | 167](#)

[hierarchical-scheduler](#)

## Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces (Logical Interfaces over a Pseudowire Interface Set)

Before configuring three-level scheduling on pseudowire logical interfaces over a pseudowire logical interface set, you must first complete the following tasks:

1. Configure the pseudowire logical interfaces. See [Configuring a Pseudowire Subscriber Logical Interface](#).
2. Configure the pseudowire device count. See [Configuring the Maximum Number of Pseudowire Logical Interface Devices Supported on the Router](#).

3. Configure the pseudowire device including the logical tunnel anchor point. See *Configuring a Pseudowire Subscriber Logical Interface Device*.
4. Configure the pseudowire transport logical interface. See *Configuring the Transport Logical Interface for a Pseudowire Subscriber Logical Interface*.
5. Configure the pseudowire signaling (either Layer 2 circuit signaling or Layer 2 VPN signaling). See *Configuring Layer 2 Circuit Signaling for Pseudowire Subscriber Logical Interfaces* or *Configuring Layer 2 VPN Signaling for Pseudowire Subscriber Logical Interfaces*.
6. Configure the pseudowire logical interfaces. See *Configuring the Service Logical Interface for a Pseudowire Subscriber Logical Interface*.

Three-level scheduling on pseudowire logical interfaces over a pseudowire logical interface set requires you to apply the traffic-control profiles at both the pseudowire logical interface and the pseudowire logical interface-set. To configure CoS policies on MPLS pseudowire subscriber interfaces using three-level implicit hierarchical scheduling:

1. Configure the hierarchical scheduler for the physical interface used for the logical tunnel (anchor point). For three-level scheduling the hierarchical scheduler must be set to `implicit-hierarchy`.

```
[edit]
user@host#edit interfaces ps-anchor-device-name
user@host#set hierarchical-scheduler implicit-hierarchy
```

2. Specify the traffic-control profile to use on the pseudowire logical interfaces.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#set output-traffic-control-profile profile-name
```

3. Define a pseudowire logical interface set and configure the traffic-control profile used for the interface set.

```
[edit class-of-service]
user@host#edit interfaces
user@host#edit interface-set interface-set-name
user@host#edit output-traffic-control-profile profile-name
```



- Group the pseudowire logical interfaces in the pseudowire logical interface set.

```
[edit ]
user@host#edit interfaces
user@host#edit interface-set interface-set-name
user@host#edit interface ps ps-device-name
user@host#edit unit logical-unit-number
```

- Configure the rewrite rule.

The available rewrite rule types for pseudowire interfaces are dscp and inet-precedence.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#edit rewrite-rules (dscp | inet-precedence) rewrite-name
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-point (alias | bits)
```

- Configure the classifier.

The available classifier types for pseudowire interfaces are dscp and inet-precedence.

```
[edit class-of-service]
user@host#edit interfaces ps ps-device-name
user@host#edit unit logical-unit-number
user@host#edit classifiers (dscp | inet-precedence) classifier-name
user@host#edit forwarding-class class-name
user@host#set loss-priority class-name code-points [aliases] [bit-patterns]
```

- Apply the rewrite rule and classifier to the pseudowire interfaces.

For the *interface\_name* parameter, specify the ps device name.

```
[edit class-of-service interfaces interface_name unit logical-unit-number]
user@host#set rewrite-rule (dscp | inet-precedence) (rewrite-name | default) protocol
protocol-types
user@host#set classifiers (dscp | inet-precedence) (classifier-name | default)
```

## RELATED DOCUMENTATION

*CoS on Ethernet Pseudowires in Universal Edge Networks Overview*

---

*Mapping CoS Component Inputs to Outputs*

---

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

---

[Hierarchical CoS on MPLS Pseudowire Subscriber Interfaces Overview | 153](#)

---

[CoS Three-Level Hierarchical Scheduling on MPLS Pseudowire Subscriber Interfaces | 160](#)

---

[CoS Configuration Overview for MPLS Pseudowire Subscriber Interfaces | 154](#)

---

[Configuring CoS Three-Level Hierarchical Scheduling for MPLS Pseudowire Subscriber Interfaces \(Logical Interfaces over a Transport Logical Interface\) | 165](#)

---

*hierarchical-scheduler*

## Configuring Hierarchical Scheduling for L2TP

### IN THIS CHAPTER

- CoS for L2TP LAC Subscriber Interfaces Overview | 171
- Configuring Dynamic CoS for an L2TP LAC Tunnel | 174
- CoS for L2TP LNS Inline Services Overview | 176
- Configuring an Inline Service Interface for L2TP LNS | 177
- Configuring Dynamic CoS for an L2TP LNS Inline Service | 178

### CoS for L2TP LAC Subscriber Interfaces Overview

#### IN THIS SECTION

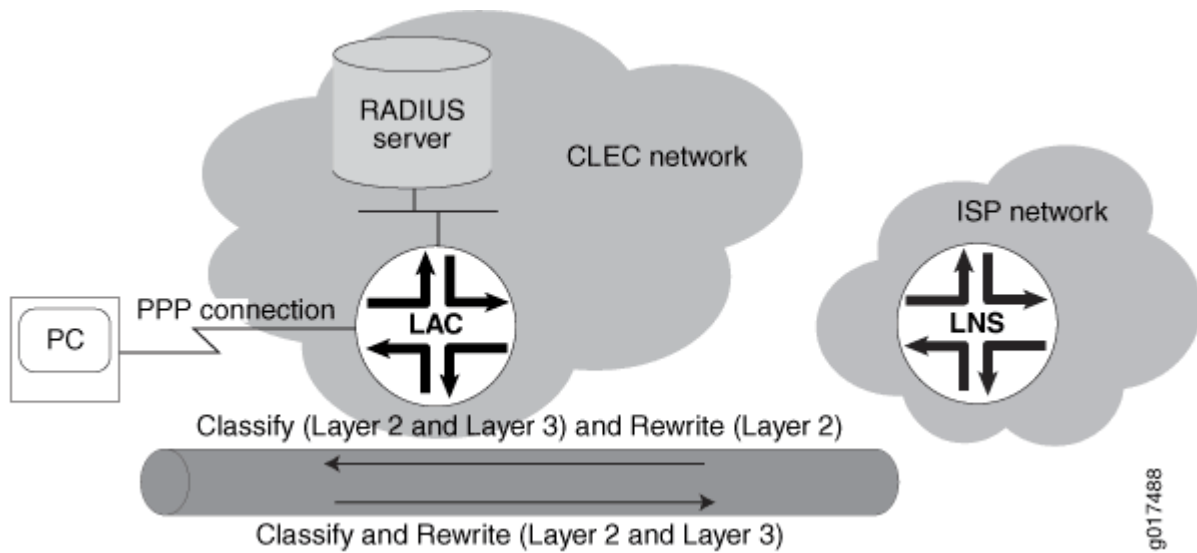
- Traffic from LAC to LNS | 172
- LAC Tunnels: Traffic from LNS to LAC | 173

You can apply CoS to the Layer 2 Tunnel Protocol (L2TP) access concentrator (LAC) component.

In Layer 2 Tunnel Protocol (L2TP) configurations, IP and L2TP headers are added to packets arriving at a PPP subscriber interface on the L2TP access concentrator (LAC) before being tunneled to the L2TP network server (LNS). You can manage the IP header by configuring classifiers and rewrite-rules that transfer the ToS (Type of Service) value or the 802.1p value from the *inner* IP header to the *outer* IP header of the L2TP packet.

[Figure 21 on page 172](#) shows the classifier and *rewrite rules* that you can configure from the LAC to the LNS, and from the LNS to the LAC.

Figure 21: CoS Configuration for L2TP LAC Topology



### Traffic from LAC to LNS

To set the ToS value or the 802.1p value on the inner IP header, you can configure both fixed and behavior aggregate (BA) classifiers for subscribers at Layer 2 or Layer 3 of the network.

[Table 15 on page 172](#) lists the configuration options for applying classifiers to a subscriber interface on an ingress LAC tunnel.

**Table 15: Ingress LAC Tunnel Classifier Options**

Classifier	Subscriber Interface
Fixed	Either of the following: <ul style="list-style-type: none"> <li>• PPP interface</li> <li>• Underlying VLAN interface</li> </ul>
Layer 2	Either of the following: <ul style="list-style-type: none"> <li>• PPP interface</li> <li>• Underlying VLAN interface</li> </ul>
Layer 3	Family of PPP interfaces

You cannot configure a Layer 2 and fixed classifier together.

The behavior of the Layer 2 and Layer 3 classifiers depends on the configuration. For example, a Layer 3 classifier for a family of PPP interfaces overrides a Layer 2 classifier configured at the PPP interface, except for the unknown packets and control packets.

If you do not configure a classifier for Layer 2, the system applies the default Layer 3 classifier so that tunneled and terminated subscribers have the same behavior. To prevent unknown packets and control packets from being discarded, the system assigns them to the best-effort forwarding class.

For egress tunnels, you configure rewrite rules at the PPP interface to set the ToS or 802.1p value of the outer IP header. Rewrite rules are applied accordingly to the forwarding class, packet loss priority (PLP), and code point.

## LAC Tunnels: Traffic from LNS to LAC

On a LAC, mapping the inner IP header to the outer IP header of the L2TP packet depends on the classifier and rewrite-rule configurations. For example, [Table 16 on page 173](#) lists the values for the classifier and rewrite rules for a VLAN interface. For assured forwarding, the inner 802.1p value (**ob001**) is classified with the assured-forwarding class and low loss priority at the ingress interface. Based on the assured-forwarding class and low loss priority in the rewrite rule, the ToS value in the outer IP header is set to **ob001**.

**Table 16: Sample Result for the Classifier and Rewrite Rules for a VLAN Interface**

Inner .1p Value	Forwarding Class	Loss Priority	Code Point	Outer ToS Value
ob000	best-effort	low	000	ob000
ob001	assured-forwarding	low	001	ob001
ob101	expedited-forwarding	low	101	ob101
ob111	network-control	low	11	ob111

## RELATED DOCUMENTATION

| [Configuring Dynamic CoS for an L2TP LAC Tunnel](#) | 174

## Configuring Dynamic CoS for an L2TP LAC Tunnel

Before you begin, configure the L2TP LAC. See *Configuring an L2TP LAC*.

In L2TP configurations, IP and L2TP headers are added to packets arriving at a PPP subscriber interface on the LAC before being tunneled to the L2TP network server (LNS).

Classifiers and rewrite rules enable you to properly transfer the ToS (Type of Service) value or the 802.1p value from the inner IP header to the outer IP header of the L2TP packet.

To manage the IP header values for a LAC tunnel:

1. Configure the classifier for the inner tunnel.
  - a. Define the fixed or behavior aggregate (BA) classifier.
    - To configure a fixed classifier:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
user@host# set forwarding-class class-name
```

- To configure a BA classifier:

```
[edit class-of-service]
user@host#set classifiers (ieee-802.1 | inet-precedence) classifier-name forwarding-
class class-name loss-priority level code-points [ aliases ] [ bit-patterns]
```

- b. Apply the classifier to the Layer 2 interface or Layer 3 interface. For Layer 2, you can apply the classifier at the PPP interface or an underlying VLAN interface. For Layer 3, you can apply classifiers to a family of PPP interfaces.
  - To apply the classifier for the IEEE 802.1p value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number classifiers]
user@host# set ieee-802.1 (classifier-name | default) vlan-tag (inner | outer)
```

- To apply the classifier for the ToS value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number classifiers]
user@host# set inet-precedence (classifier-name | default)
```

## 2. Configure the rewrite rule for the egress tunnel.

- Configure the rewrite rule with the forwarding class and the loss priority value.

```
[edit class-of-service]
user@host# set rewrite-rules (ieee-802.1 | inet-precedence) rewrite-name forwarding-class
class-name loss-priority level code-point (alias | bits)
```

- Apply the rewrite rule to the PPP interface for which the L2TP tunnel is configured.

- To apply the rewrite-rule for the IEEE 802.1p value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number rewrite-rules]
user@host# set ieee-802.1 (rewrite-name | default) vlan-tag (outer | outer-and-inner)
```

- To apply the rewrite rule for the ToS value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number rewrite-rules]
user@host# set inet-precedence (rewrite-name | default)
```

## RELATED DOCUMENTATION

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

[CoS for L2TP LAC Subscriber Interfaces Overview | 171](#)

## CoS for L2TP LNS Inline Services Overview

### IN THIS SECTION

- Guidelines for Applying CoS to the LNS | 176
- Hardware Requirements for Inline Services on the LNS | 177

You can apply hierarchical scheduling and per-session shaping to Layer 2 Tunnel Protocol (L2TP) network server (LNS) inline services using a static or dynamic CoS configuration.

This feature is supported on MIC and MPC interfaces on MX240, MX480, and MX960 routers.

### Guidelines for Applying CoS to the LNS

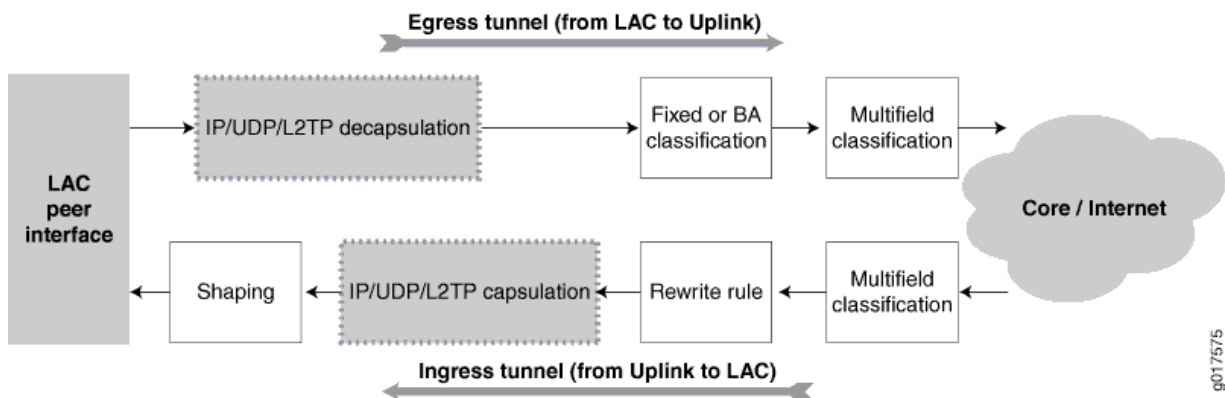
In L2TP configurations, IP, UDP, and L2TP headers are added to packets arriving at a PPP subscriber interface on the L2TP access concentrator (LAC) before being tunneled to the LNS.

When a service interface is configured for an L2TP LNS session, it has an *inner* IP header and an outer IP header. You can configure CoS for an LNS session that corresponds to the inner IP header only. The *outer* IP header is used for L2TP tunnel processing only.

However, we recommend that you configure classifiers and rewrite-rules to transfer the ToS (type of service) value from the inner IP header to the outer IP header of the L2TP packet.

Figure 22 on page 176 shows the classifier and *rewrite rules* that you can configure on an LNS inline service.

Figure 22: Processing of CoS Parameters in an L2TP LNS Inline Service





By default, the shaping calculation on the service interface includes the L2TP encapsulation. If necessary, you can configure additional adjustments for downstream ATM traffic from the LAC or differences in Layer 2 protocols.

## Hardware Requirements for Inline Services on the LNS

Hierarchical scheduling for L2TP LNS inline services is supported on MIC and MPC interfaces only. The services that you can configure depend on the hardware combination. [Table 17 on page 177](#) lists the supported inline services and peer interfaces for each MIC and MPC combination.

**Table 17: Hardware Requirements for L2TP LNS Inline Services**

MPC Module	Inline Service Support—With Per-Session Shaping	Inline Service Support—Without Per-Session Shaping
MPC2E-3D-NG	No	Yes
MPC2E-3D-NG-Q MX80	Yes	Yes
MPC-3D-16XGE-SFPP	No	No

### RELATED DOCUMENTATION

[Configuring Static CoS for an L2TP LNS Inline Service](#)

[Configuring Dynamic CoS for an L2TP LNS Inline Service | 178](#)

## Configuring an Inline Service Interface for L2TP LNS

The inline service interface is a virtual physical service interface that resides on the Packet Forwarding Engine. This *si* interface, referred to as an *anchor* interface, makes it possible to provide L2TP services without a special services PIC. The inline service interface is supported only by MPCs on MX Series routers. Four inline service interfaces are configurable per MPC-occupied chassis slot.

You can maximize the number of sessions that can be shaped in one service interface by setting the maximum number of hierarchy levels to two. In this case, each LNS session consumes one L3 node in the scheduler hierarchy for shaping.

If you do not specify the number of levels (two is the only option), then the number of LNS sessions that can be shaped on the service interface is limited to the number of L2 nodes, or 4096 sessions. Additional sessions still come up, but they are not shaped.

To configure an inline service interface:

1. Access the service interface.

```
[edit interfaces]
user@host# edit si-slot/pic/port
```

2. (Optional; for per-session shaping only) Enable the inline service interface for hierarchical schedulers and limit the number of scheduler levels to two.

```
[edit interfaces si-slot/pic/port]
user@host# set hierarchical-scheduler maximum-hierarchy-levels 2
```

3. (Optional; for per-session shaping only) Configure services encapsulation for inline service interface.

```
[edit interfaces si-slot/pic/port]
user@host# set encapsulation generic-services
```

4. Configure the IPv4 family on the reserved unit 0 logical interface.

```
[edit interfaces si-slot/pic/port]
user@host# set unit 0 family inet
```

## Configuring Dynamic CoS for an L2TP LNS Inline Service

Before you begin, configure the L2TP LNS inline service interface. See *Configuring an L2TP LNS with Inline Service Interfaces*.

You can configure hierarchical scheduling for an L2TP LNS inline service and manage the IP header values using rewrite rules and classifiers.

To configure CoS for an L2TP LNS inline service in a dynamic profile:

1. Configure the hierarchical scheduler for the service interface (si) interface.

```
[edit interfaces si-fpc/port/pic ]
user@host# set hierarchical-scheduler maximum-hierarchy-levels 2
```



**BEST PRACTICE:** To enable Level 3 nodes in the LNS scheduler hierarchy and to provide better scaling, we recommend that you also specify a maximum of two hierarchy levels.

2. Configure the LNS to reflect the IP ToS value in the inner IP header to the outer IP header.

```
[edit services l2tp tunnel-group name]
user@host# set tos-reflect
```

3. Configure the classifier for egress traffic from the LAC.
  - a. Define the fixed or behavior aggregate (BA) classifier.

- To configure a fixed classifier:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
user@host# set forwarding-class class-name
```

- To configure a BA classifier:

```
[edit class-of-service]
user@host# set classifiers (dscp | dscp-ipv6 | inet-precedence) classifier-name
forwarding-class class-name loss-priority level code-points [ aliases ] [ bit-patterns]
```

- b. Apply the classifier to the service interface.

- To apply the classifier for the DSCP or DSCP IPv6 value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number classifiers]
user@host# set dscp (classifier-name | default)
user@host# set dscp-ipv6 (classifier-name | default)
```

- To apply the classifier for the ToS value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number classifiers]
user@host# set inet-precedence (classifier-name | default)
```

#### 4. Configure and apply a rewrite-rule to ingress traffic to the LAC:

- Configure the rewrite rule with the forwarding class and the loss priority value.

```
[edit class-of-service]
user@host# set rewrite-rules (dscp | dscp-ipv6 | inet-precedence) rewrite-name forwarding-
class class-name loss-priority level code-point (alias | bits)
```

- Apply the rewrite rule to the service interface.

- To apply the rewrite rule for the DSCP or DSCP IPv6 value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number rewrite-rules]
user@host# setdscp (rewrite-name | default)
user@host# set dscp-ipv6 (rewrite-name | default)
```

- To apply the rewrite rule for the ToS value:

```
[edit dynamic-profiles profile-name class-of-service interfaces interface-name unit
logical-unit-number rewrite-rules]
user@host# set inet-precedence (rewrite-name | default)
```

#### 5. (Optional) Configure additional adjustments for downstream ATM traffic.

By default, the shaping calculation on the service interface includes the L2TP encapsulation. If necessary, you can configure additional adjustments for downstream ATM traffic from the LAC or differences in Layer 2 protocols.

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name]
user@host# set overhead-accounting (frame-mode | cell-mode | $junos-cos-shaping-mode) <bytes
(byte-value | $junos-cos-byte-adjust)
```

6. Apply the traffic-control profile.

```
[edit dynamic-profiles profile-name class-of-service interfaces $junos-interface-ifd-name  
unit $junos-interface-unit]  
user@host# set output-traffic-control-profile profile-name
```

## RELATED DOCUMENTATION

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

---

[CoS for L2TP LNS Inline Services Overview | 176](#)

---

*Example: Configuring an L2TP LNS*

---

*Configuring Dynamic Shaping Parameters to Account for Overhead in Downstream Traffic Rates*

# Preventing Bandwidth Contention on Subscriber Interfaces

## IN THIS CHAPTER

- [Hierarchical CoS Shaping-Rate Adjustments Overview | 182](#)
- [Shaping Rate Adjustments for Subscriber Local Loops Overview | 185](#)
- [Guidelines for Configuring Shaping-Rate Adjustments for Subscriber Local Loops | 186](#)
- [Configuring the Minimum Adjusted Shaping Rate on Scheduler Nodes for Subscribers | 187](#)
- [Configuring Shaping-Rate Adjustments on Queues | 188](#)
- [Enabling Shaping-Rate Adjustments for Subscriber Local Loops | 191](#)
- [Disabling Shaping-Rate Adjustments for Subscriber Local Loops | 197](#)
- [Disabling Hierarchical Bandwidth Adjustment for Subscriber Interfaces with Reverse-OIF Mapping | 198](#)
- [Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops | 198](#)
- [Verifying the Configuration of Shaping-Rate Adjustments for Subscriber Local Loops | 202](#)
- [Verifying the Configuration of ANCP for Shaping-Rate Adjustments | 203](#)
- [Using Hierarchical CoS to Adjust Shaping Rates Based on Multicast Traffic | 204](#)

## Hierarchical CoS Shaping-Rate Adjustments Overview

### IN THIS SECTION

- [Types of Shaping-Rate Adjustments | 183](#)
- [Levels of Shaping-Rate Adjustments | 183](#)

This overview describes how MX Series 5G Universal Routing Platforms installed in a subscriber access network can adjust hierarchical class-of-service (CoS) parameters to prevent bandwidth contention at subscriber interfaces.

Hierarchical CoS is supported only for subscriber interfaces on Enhanced Queueing (EQ) DPCs or MPC interfaces operating in hierarchical scheduler mode.

The characteristics of voice, data, and video applications vary widely in their requirements for traffic throughput, bandwidth management, delay and *jitter* tolerance, and buffer depth. To prevent bandwidth contention at subscriber interfaces, you can configure applications such as ANCP and Multicast to perform real-time adjustments to the shaping rate configured for subscriber interfaces for residential gateways. Enabling shaping-rate adjustments on the router can prevent bandwidth contention at the interface from causing degradation of the subscriber's voice, data, or video services.

## Types of Shaping-Rate Adjustments

The ANCP application supports *absolute* adjustments to a specific shaping-rate value. You can configure ANCP to communicate the subscriber local loop speed to the MX Series router, which in turn throttles traffic destined to the associated subscriber interface so that it matches the subscriber local loop speed. ANCP acquires subscriber line rate information from DSLAMs and then communicates this data transmission rate for use with CoS.

The OIF mapping and reverse OIF mapping multicast applications support *delta* adjustments that increase or decrease the current shaping rate by a certain value. The system adjusts traffic destined to the subscriber using reverse OIF mapping enabled on a specified multicast interface. Reverse OIF mapping is used to determine the subscriber VLAN interface and the multicast traffic bandwidth on the interface.

## Levels of Shaping-Rate Adjustments

Both absolute and delta adjustments are made to a subscriber's aggregate shaping rate on a level 3 scheduler node.

Adjustments that occur on the scheduler node can also impact the shaping rates for all queues. This adjustment can be undesirable for service providers who want to provide a premium level of service on specific queues.

For delta-based adjustments by multicast applications, you can control the distribution of shaping rates among queues by assigning the percentage of adjustment allowed for each queue. In addition, you can set a minimum adjusted shaping rate for each queue.

[Figure 23 on page 184](#) shows a sample multicast network with shaping rates adjusted at the scheduler node level. The shaping rate is reduced by 4 Mbps (from 41 Mbps to 37 Mbps) at the scheduler node for subscriber interface 1, which reduces the rates of both the best effort and video on demand (VoD) service queues.

Figure 23: Scheduler Node and Queues with Adjusted Shaping Rates

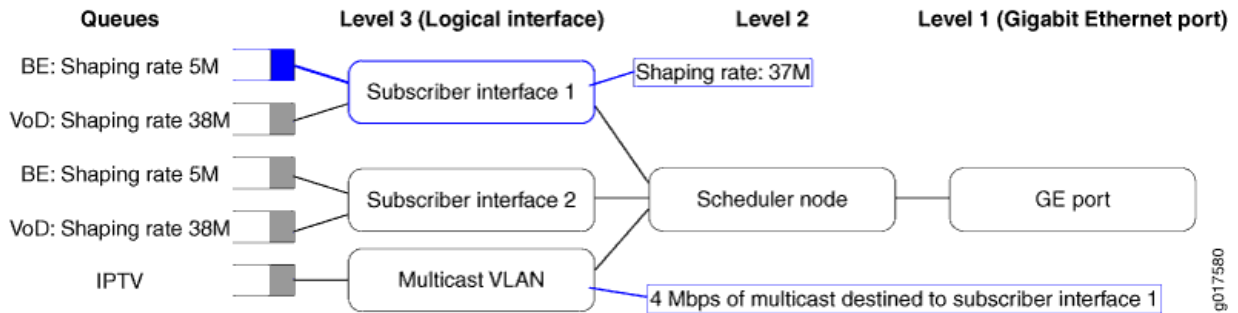
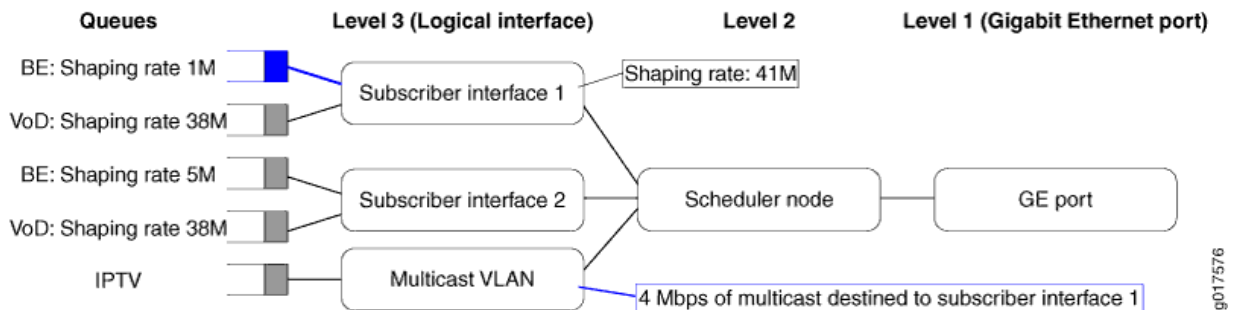


Figure 24 on page 184 shows the same network with queue-based adjustments enabled for the best-effort queue on subscriber 1. The shaping rate of the best-effort queue is reduced by 4 Mbps (from 5 Mbps to 1 Mbps). The VoD service queue is not affected.

Figure 24: Queue with Adjusted Shaping Rate



## RELATED DOCUMENTATION

[Configuring the Minimum Adjusted Shaping Rate on Scheduler Nodes for Subscribers | 187](#)

[Configuring Shaping-Rate Adjustments on Queues | 188](#)

[Shaping Rate Adjustments for Subscriber Local Loops Overview | 185](#)

[Disabling Hierarchical Bandwidth Adjustment for Subscriber Interfaces with Reverse-OIF Mapping | 198](#)

[Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops | 198](#)



## Shaping Rate Adjustments for Subscriber Local Loops Overview

This overview describes how an MX Series 5G Universal Routing Platform installed as an edge router can adjust hierarchical CoS policy for subscriber interfaces for subscriber local loops. You can configure the router to throttle the traffic sent to subscriber local loops so that the traffic does not exceed the current data transmission rate of those lines. This feature ensures that changes to subscriber local loop speeds do not cause bandwidth contention at the subscriber's residential gateway.

In a typical subscriber access network, traffic destined to a subscriber is delivered from the access network, through an edge router, to a DSLAM. The DSLAM multiplexes subscriber traffic through a DSL, also known as a *local loop*, to the subscriber's residential gateway. When line noise or cross talk in a subcarrier causes the error rate on a DSL to exceed a certain threshold, the DSLAM can adapt itself by lowering the data transmission rate to that carrier device. A lower data transmission rate is less susceptible to induced errors.

You can configure an MX Series router to adjust the configured shaping rates on scheduler nodes for subscriber interfaces that represent subscriber local loops. Whenever a DSLAM resynchronizes a subscriber local loop speed, the router adjusts the configured shaping rate for that line so that the aggregate egress traffic to those subscribers is shaped to the local loop speed before the traffic reaches the DSLAM. Unless the maximum amount of bandwidth allocated to the subscriber interface on the router is throttled to the local loop speed, bandwidth contention can occur at the subscriber's residential gateway, which can cause the DSLAM to drop packets. This type of shaping-rate adjustment requires the topology discovery and traffic-monitoring features of the Access Node Control Protocol (ANCP).

You can enable ANCP to communicate the subscriber local loop speed to CoS, which in turn throttles traffic destined to the associated subscriber interface so that it matches the subscriber local loop speed. The ANCP agent acquires unadjusted (net) subscriber line rate information from DSLAMs and then communicates this data transmission rate for use with CoS. You can also configure percentage and byte adjustments that the ANCP agent can make to the received net data rate for frame-mode DSL types before communicating the adjusted rate and overhead to CoS.

### RELATED DOCUMENTATION

---

[Hierarchical CoS Shaping-Rate Adjustments Overview | 182](#)

---

[Guidelines for Configuring Shaping-Rate Adjustments for Subscriber Local Loops | 186](#)

---

[Enabling Shaping-Rate Adjustments for Subscriber Local Loops | 191](#)

---

[Disabling Shaping-Rate Adjustments for Subscriber Local Loops | 197](#)

---

[Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops | 198](#)

---

*ANCP and the ANCP Agent Overview*

## Guidelines for Configuring Shaping-Rate Adjustments for Subscriber Local Loops

These guidelines apply to configuring an MX Series 5G Universal Routing Platform installed as an edge router to adjust the configured shaping rates on scheduler nodes for subscriber interfaces that represent subscriber local loops. This shaping-rate feature uses the topology discovery and traffic-monitoring features of ANCP.

When you enhance hierarchical CoS policy by configuring ANCP-driven shaping-rate adjustments, consider the following guidelines:

- Shaping-rate adjustments are supported only for subscriber local loops that terminate at DSLAMs that you have configured as ANCP neighbors of the MX Series router.
- Shaping-rate adjustments are supported only for scheduler nodes for which you have configured an initial shaping rate by including the `shaping-rate` statement in a traffic-control profile applied to the scheduler node. Specify the initial shaping rate as a peak rate, in bits per second (bps), and not as a percentage. Other methods of configuring a shaping rate are not supported with this feature.
- Shaping-rate adjustments are supported only for scheduler nodes that are static *logical interface* sets that you have configured to operate at Level 3 of the scheduler hierarchy on the router. If an interface set is configured with a logical interface (such as unit 0) and queue, then the interface set is an internal scheduler node (as opposed to a root node or a leaf node) at Level 2 of the hierarchy. However, if there are no traffic-control profiles are configured on logical interfaces in an interface set, then the interface set is an internal scheduler node at Level 3 of the hierarchy.
- Shaping-rate adjustments are supported only for subscriber interfaces over physical interfaces that you have configured to operate in hierarchical scheduler mode.
- After shaping-rate adjustments are enabled and the router has performed shaping-rate adjustments on a scheduler node, you can configure a new shaping rate by including the `shaping-rate` statement in a traffic-control profile and then applying that profile to that scheduler node. However, this new shaping-rate value does not immediately result in shaping traffic at the new rate. The scheduler node continues to be shaped at rate set by ANCP. Only when the ANCP shaping-rate adjustment feature is disabled is the scheduler node shaped at the newly configured shaping-rate.
- The Layer 2 Tunneling Protocol (L2TP) is often used to carry traffic securely between an L2TP Network Server (LNS) and an L2TP Access Concentrator (LAC). The QoS adjustment feature supports the shaping overhead options that you can use to add a specified number of bytes to the actual packet length when determining shaped session packet length. ANCP shaping-rate adjustments are not supported for ingress traffic, only for egress traffic. To configure the number of bytes to add to the packet at the egress side of the tunnel, include the **egress-shaping-overhead** and `mode` statements at the `[edit chassis fpc slot-number pic pic-number traffic-manager]` hierarchy level. Use the shaping overhead options if you need to account for encapsulation overhead.

For more information about the ANCP protocol, see the *ANCP and the ANCP Agent Overview*.

## RELATED DOCUMENTATION

[Hierarchical CoS Shaping-Rate Adjustments Overview | 182](#)

[Shaping Rate Adjustments for Subscriber Local Loops Overview | 185](#)

[Enabling Shaping-Rate Adjustments for Subscriber Local Loops | 191](#)

[Disabling Shaping-Rate Adjustments for Subscriber Local Loops | 197](#)

[Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops | 198](#)

## Configuring the Minimum Adjusted Shaping Rate on Scheduler Nodes for Subscribers

### IN THIS SECTION

- [Overview | 187](#)
- [Configuring a Static Minimum Adjusted Shaping Rate on Scheduler Nodes | 188](#)
- [Configuring a Dynamic Minimum Adjusted Shaping Rate on Scheduler Nodes | 188](#)

### Overview

Absolute adjustments and delta adjustments are performed at the scheduler node level. You can configure a minimum adjusted shaping rate at the scheduler node level using static or dynamic CoS parameters.

This feature is supported for adjustments performed by the ANCP and multicast applications.



**BEST PRACTICE:** For multicast traffic, you can configure a minimum adjusted shaping rate at the queue level. We recommend that you configure the minimum adjusted value at the scheduler node or the queue, but not both.

When you configure a minimum adjusted value for a node and for a scheduler that is referenced by a scheduler map in the same traffic-control-profile, the system uses the minimum value from the scheduler.

## Configuring a Static Minimum Adjusted Shaping Rate on Scheduler Nodes

To apply a minimum adjusted shaping rate for a scheduler node:

- Configure the `adjust-minimum` statement for the static traffic-control profile.

```
[edit class-of-service traffic-control-profiles profile-name]  
user@host# set adjust-minimum rate
```

## Configuring a Dynamic Minimum Adjusted Shaping Rate on Scheduler Nodes

To apply a minimum adjusted shaping rate for a scheduler node:

- Configure the `adjust-minimum` statement for the dynamic traffic-control profile.

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profiles profile-name]  
user@host# set adjust-minimum rate
```

### RELATED DOCUMENTATION

*Verifying the Scheduling and Shaping Configuration for Subscriber Access*

[Configuring Shaping-Rate Adjustments on Queues | 188](#)

[Hierarchical CoS Shaping-Rate Adjustments Overview | 182](#)

*Configuring Traffic Scheduling and Shaping for Subscriber Access*

## Configuring Shaping-Rate Adjustments on Queues

### IN THIS SECTION

- [Overview | 189](#)
- [Configuring a Static Shaping-Rate Adjustment for Queues | 189](#)
- [Configuring a Dynamic Shaping-Rate Adjustment for Queues | 190](#)

## Overview

By default, the multicast application adjusts the shaping rates at the scheduler node level. This adjustment also impacts the shaping rates for all queues, which can be undesirable for service providers who want to provide a premium level of service on specific queues.

For multicast applications, you can control the distribution of shaping rates among queues by assigning the percentage of adjustment allowed for each queue. In addition, you can set a minimum adjusted shaping rate for each queue.

This feature is supported for adjustments performed by the multicast application.



**BEST PRACTICE:** We recommend that you configure the minimum adjusted value at the scheduler node or the queue, but not both.

When you configure a minimum adjusted value for a node and for a scheduler that is referenced by a scheduler map in the same traffic-control-profile, the system uses the minimum value from the scheduler.

## Configuring a Static Shaping-Rate Adjustment for Queues

To configure adjustment parameters for a queue:

1. Configure the percentage of adjustment for the shaping rate.

```
[edit class-of-service schedulers scheduler-name]
user@host# set adjust-percent percentage
```

2. Configure the minimum adjusted value for the shaping rate.

Do one of the following:

- Configure the minimum adjusted value for the queue.

```
[edit class-of-service schedulers scheduler-name]
user@host# set adjust-minimum rate
```

- Configure the minimum adjusted value for the node.

```
[edit class-of-service traffic-control-profile profile-name]
user@host# set adjust-minimum rate
```



**BEST PRACTICE:** Ensure that the minimum adjusted value that you configure does not exceed the shaping rate and is not lower than the configured transmit rate.

## Configuring a Dynamic Shaping-Rate Adjustment for Queues

To configure adjustment parameters for a queue in a dynamic profile:

1. Configure the percentage of adjustment for the shaping rate.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set adjust-percent percentage
```

2. Configure the minimum adjusted value for the shaping rate.

Do one of the following:

- Configure the minimum adjusted value for the queue.

```
[edit dynamic-profiles profile-name class-of-service schedulers scheduler-name]
user@host# set adjust-minimum (rate | $junos-cos-adjust-minimum)
```

- Configure the minimum adjusted value for the node.

```
[edit dynamic-profiles profile-name class-of-service traffic-control-profile profile-name]
user@host# set adjust-minimum rate
```



**BEST PRACTICE:** Ensure that the minimum adjusted value that you configure does not exceed the shaping rate and is not lower than the configured transmit rate.

## RELATED DOCUMENTATION

*Verifying the Scheduling and Shaping Configuration for Subscriber Access*

[Configuring the Minimum Adjusted Shaping Rate on Scheduler Nodes for Subscribers](#) | 187

[Hierarchical CoS Shaping-Rate Adjustments Overview](#) | 182

*Configuring Traffic Scheduling and Shaping for Subscriber Access*

## Enabling Shaping-Rate Adjustments for Subscriber Local Loops

### IN THIS SECTION

- [Configuring Static Logical Interface Sets to Serve as CoS Hierarchical Scheduler Nodes for Subscriber Loops | 191](#)
- [Configuring the Logical Interfaces That Compose the Static Logical Interface Sets | 192](#)
- [Configuring Hierarchical CoS on the Static Logical Interface Sets That Serve as Hierarchical Scheduler Nodes for Subscriber Local Loops | 193](#)
- [Configuring ANCP Functionality That Supports and Drives Shaping-Rate Adjustments for Subscriber Local Loops | 195](#)

You can enhance a CoS implementation by enabling an MX Series 5G Universal Routing Platform to adjust the hierarchical CoS policy shaping rate configured for static interface sets that consist of two or more VLANs and represent subscriber local loops. Whenever the digital subscriber line access multiplexer (DSLAM) resynchronizes its data transmission rate to a digital subscriber line (DSL), the router adjusts the shaping rate for the associated subscriber interface so that the maximum bandwidth allocation cannot exceed the current data rate for the associated subscriber local loop. This feature ensures that data transmission rate adjustments by the DSLAM do not cause bandwidth contention at the subscriber's residential gateway.

This topic includes the following tasks:

### Configuring Static Logical Interface Sets to Serve as CoS Hierarchical Scheduler Nodes for Subscriber Loops

To configure a logical interface set, begin by including the `interface-set` statement with the *interface-set-name* option at the `[edit interfaces]` hierarchy level.

An interface set is composed of two or more logical interfaces on the same physical interface. Each logical interface in an interface set corresponds to an individual subscriber service, such as voice, video, or data. To specify either a list of logical unit numbers or the single outer VLAN tag used to identify the logical interfaces that compose the interface set, include statements at the `[edit interfaces interface-set interface-set-name]` hierarchy level:

- For an interface set composed of a list of logical interfaces identified by an inner VLAN tag on Ethernet frames (called the customer VLAN, or C-VLAN, tag), you must specify each logical interface by including the unit statement with the *logical-unit-number* option.

```
[edit]
interfaces {
  interface-set interface-set-name {
    interface ethernet-interface-name { # EQ DPC port
      unit logical-unit-number;
      unit logical-unit-number;
      . . .
    }
    . . .
  }
}
```

- For an interface set composed of a set of VLANs grouped at the DSLAM and identified by the same service VLAN (S-VLAN) tag), you must specify the S-VLAN tag as the outer VLAN tag for each VLAN by including the *vlan-tags-outer* statement with the *vlan-tag* option.

```
[edit]
interfaces {
  interface-set interface-set-name {
    interface ethernet-interface-name { # EQ DPC port
      vlan-tags-outer vlan-tag; # Identify the DSLAM
    }
    . . .
  }
}
```

For more information, see ["Hierarchical CoS for Metro Ethernet Environments"](#) on page 14.

## Configuring the Logical Interfaces That Compose the Static Logical Interface Sets

Each underlying physical interface must be configured to operate in hierarchical scheduler mode and to support stacked VLAN tagging on all logical interfaces. To configure, include the *hierarchical-scheduler* statement and the *stacked-vlan-tagging* statement at the `[edit interfaces ethernet-interface-name]` hierarchy level.

To associate the individual logical interfaces of an interface set with specific subscriber services provided by the subscriber local loop, bind an S-VLAN tag and a C-VLAN tag to each logical interface that belongs



to a scheduler node that represents a subscriber local loop. Ethernet frames sent from the logical interfaces contain an outer VLAN tag that identifies a DSLAM and an inner VLAN tag that identifies a subscriber port on the DSLAM. To configure, include the `vlan-tags` statement at each logical interface:

```
[edit]
interfaces {
  ethernet-interface-name { # EQ DPC port underlying an interface set
    hierarchical-scheduler;
    stacked-vlan-tagging; # Support 802.1Q VLAN dual-tagged frames
    unit logical-unit-number { # Bind S-VLAN and C-VLAN tags to logical interface
      vlan-tags inner tpid.vlan-id outer tpid.vlan-id;
    }
    . . .
  }
}
```

For more information, see [802.1Q VLANs Overview](#).

## Configuring Hierarchical CoS on the Static Logical Interface Sets That Serve as Hierarchical Scheduler Nodes for Subscriber Local Loops

To configure hierarchical CoS on the static logical interface set that serves as the hierarchical scheduler node for a subscriber local loop:

1. For each scheduler node that represents a subscriber local loop, configure an initial shaping rate.



**NOTE:** The CoS shaping-rate feature is supported only for scheduler nodes with a configured shaping rate. The initial shaping rate must be configured by applying a traffic-control profile that includes the `shaping-rate` statement. Specify the initial shaping rate as a peak rate, in bits per second (bps), and not as a percentage. Other methods of configuring a shaping rate are not supported with this feature.

- To enable traffic heading downstream (from the router to the DSLAM) to be gathered into an interface set, include the `interface-set` statement and define the logical interface set name as the `interface-set-name` option at the `[edit class-of-service interfaces]` hierarchy level.
- To apply output traffic scheduling and shaping parameters at the logical interface set level (rather than at the logical unit level), include the `output-traffic-control-profile` statement and specify the name of a traffic-control profile as the `profile-name` option at the `[edit class-of-service interfaces interface-set interface-set-name]` hierarchy level.

To configure, include the following statements:

```

interfaces { # Configure interface-specific CoS for incoming packets
  interface-set interface-set-name { # Configure a hierarchical scheduler
    output-traffic-control-profile tc-profile-name; # Level 3 scheduler node
  }
  . . .
}
traffic-control-profiles { # Define traffic-control profiles
  tc-profile-name { # Specify a scheduler map and traffic-shaping parameters
    scheduler-map map-name;
    shaping-rate rate; # This is the "configured shaping rate"
    guaranteed-rate (percent percentage | rate);
    delay-buffer-rate (percent percentage | rate);
  }
  . . .
}

```

You can include the statements at the following hierarchy levels:

- [edit class-of-service]
  - [edit dynamic-profiles *profile-name* class-of-service]
2. Configure the scheduler maps referenced in the traffic-control profiles applied to the interface sets, the schedulers referenced in those scheduler maps, and the drop profiles referenced in those schedulers.
- A scheduler map establishes the traffic output queues (forwarding classes) for a scheduler node and associates each queue with a specific scheduler map.
  - A scheduler defines queue properties (transmit rate, buffer size, priority, and drop profile) that specify how traffic is treated in the output queue.
  - A drop profile specifies how aggressively the MX Series router drops packets that are managed by a particular scheduler by defining either a segmented or interpolated graph that maps output queue fullness to packet drop probability.

To configure, include the statements at the static [edit class-of-service] hierarchy level:

```

[edit]
class-of-service {
  scheduler-maps { # Assign queuing characteristics to output queues
    map-name { # Map output queues to

```

```

        forwarding-class class-name scheduler scheduler-name;
        forwarding-class class-name scheduler scheduler-name;
        ...
    }
    ...
}
schedulers { # Define queuing characteristics
    scheduler-name { # Specify queuing and buffer management
        transmit-rate transmit-rate-option;
        buffer-size buffer-size-option;
        priority priority-level;
        drop-profile-map loss-priority loss-priority-option protocol any drop-profile drop-
profile-name;
        . . .
    }
}
drop-profiles { # Define random early detection (RED) for the delay buffer
    drop-profile-name { # Specify how to drop packets from an output queue
        drop-profile-name { # Map a queue fullness to a drop probability
            fill-level percentage drop-probability percentage; # Option 1: segmented
            fill-level percentage drop-probability percentage;
            . . .
        }
        interpolate { # Option 2: interpolated
            drop-probability [ values ];
            fill-level [ values ];
        }
    }
    . . .
}
}

```

For more information about configuring scheduler maps, schedulers, and drop profiles, see *Mapping CoS Component Inputs to Outputs*.

## Configuring ANCP Functionality That Supports and Drives Shaping-Rate Adjustments for Subscriber Local Loops

To configure the Access Node Control Protocol (ANCP) functionality that supports and drives the shaping-rate adjustments for subscriber local loops:

- Enable the ANCP agent to monitor subscriber local loop rates at the DSLAMs and communicate this information to CoS.

- For frame-mode DSL types, optionally configure adjustments that are made to the net data rates, the frame overhead, or both before the ANCP agent reports the values to CoS. Rates are adjusted by a percentage. Bytes are added to or subtracted from the overhead per frame.
- Configure each DSLAM as an ANCP neighbor of the router so that TCP connections can be established between the router and each DSLAM.
- Identify the subscriber interface sets whose traffic is monitored and shaped by the ANCP agent, and associate those interface sets with the corresponding identifiers configured on the access node (DSLAM) to uniquely identify the subscriber local loops within the access network.

The ANCP agent uses this information to build a mapping of subscribers to subscriber interfaces. When the ANCP agent receives port management messages from a DSLAM or other access node, it uses the access identifier contained in the message to determine which hierarchical scheduler node corresponds to the subscriber.

To configure, include statements at the [edit protocols ancp] hierarchy level:

```
[edit]
protocols {
  ancp {
    qos-adjust; # Enable ANCP to monitor and adjust CoS shaping rates
    other-bytes bytes; # Specify number of bytes to adjust OTHER access technology rate
    other-overhead-adjust percentage; # Specify percentage by which to adjust OTHER access
technology rate
    sdsl1-bytes bytes; # Specify number of bytes to adjust SDSL rate
    sdsl1-overhead-adjust percentage; # Specify percentage by which to adjust SDSL rate
    vdsl1-bytes bytes; # Specify number of bytes to adjust VDSL rate
    vdsl1-overhead-adjust percentage; # Specify percentage by which to adjust VDSL rate
    vdsl2-bytes bytes; # Specify number of bytes to adjust VDSL2 rate
    vdsl2-overhead-adjust percentage; # Specify percentage by which to adjust VDSL2 rate
  }
  neighbor ip-address; # Configure each DSLAM as an ANCP neighbor
  . . .
  interfaces { # Identify subscribers for which ANCP can adjust shaping rates
    interface-set {
      interface-set-name {
        access-identifier identifier-string; # DSLAM ID for the local loop
      }
    }
    . . .
  }
  . . .
}
```

```

    }
    . . .
}

```

## RELATED DOCUMENTATION

[Guidelines for Configuring Shaping-Rate Adjustments for Subscriber Local Loops | 186](#)

[Shaping Rate Adjustments for Subscriber Local Loops Overview | 185](#)

*Traffic Rate Reporting and Adjustment by the ANCP Agent*

*Configuring the ANCP Agent to Report Traffic Rates to CoS*

[Verifying the Configuration of ANCP for Shaping-Rate Adjustments | 203](#)

[Verifying the Configuration of Shaping-Rate Adjustments for Subscriber Local Loops | 202](#)

[Disabling Shaping-Rate Adjustments for Subscriber Local Loops | 197](#)

[Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops | 198](#)

## Disabling Shaping-Rate Adjustments for Subscriber Local Loops

To disable hierarchical CoS shaping-rate adjustments for subscriber local loops:

- Disable hierarchical CoS traffic-shaping adjustment by ANCP:

```

[edit protocols anc]
user@host# delete qos-adjust

```

Traffic-shaping parameters for all subscriber local loops revert to their current configured values.

## RELATED DOCUMENTATION

[Guidelines for Configuring Shaping-Rate Adjustments for Subscriber Local Loops | 186](#)

[Shaping Rate Adjustments for Subscriber Local Loops Overview | 185](#)

[Enabling Shaping-Rate Adjustments for Subscriber Local Loops | 191](#)

[Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops | 198](#)

## Disabling Hierarchical Bandwidth Adjustment for Subscriber Interfaces with Reverse-OIF Mapping

You can disable hierarchical bandwidth adjustment for all subscriber interfaces with reverse OIF mapping enabled on a specified multicast interface. Reverse OIF mapping is used to determine the subscriber VLAN interface and the multicast traffic bandwidth on the interface.

To disable hierarchical bandwidth adjustment:

1. Specify that you want to access the subscriber interfaces with reverse-OIF mapping enabled.

```
[edit routing-instances routing-instance routing-options multicast interface interface-name]  
user@host# edit reverse-oif-mapping
```

2. Disable hierarchical bandwidth adjustment for all subscriber interfaces on the interface.

```
user@host# set no-qos-adjust
```

### RELATED DOCUMENTATION

[Hierarchical CoS Shaping-Rate Adjustments Overview | 182](#)

*Example: Configuring Multicast with Subscriber VLANs*

## Example: Configuring Hierarchical CoS Shaping-Rate Adjustments for Subscriber Local Loops

This example shows how you can enable shaping-rate adjustments for static logical interface sets that represent subscriber local loops:

1. Configure static logical interface sets to serve as CoS hierarchical scheduler nodes for subscriber local loops.

This example uses a single scheduler node that represents two subscriber local loops. The scheduler node is a static logical interface composed of two logical interfaces. The underlying physical interface is port **0** on a Gigabit Ethernet EQ MPC in slot 4, PIC 0:

```
[edit]
interfaces {
  interface-set ifset-of-logical-interfaces {
    interface ge-4/0/0 {
      unit 1;
      unit 2;
    }
  }
  ge-4/0/0 {
    description "access interface ge-4/0/0";
    hierarchical-scheduler;
    stacked-vlan-tagging;
    unit 1 {
      description "DSL type ADSL1 = 0x01";
      proxy-arp;
      vlan-tags outer 1 inner 1; # S-VLAN tag is '1' and C-VLAN tag is '1'
      family inet { # Specify a secondary loopback address
        unnumbered-address lo0.0 preferred-source-address 192.168.7.3;
      }
    }
    unit 2 {
      description "DSL type ADSL1 = 0x01";
      proxy-arp;
      vlan-tags outer 1 inner 2; # S-VLAN tag is '1' and C-VLAN tag is '2'
      family inet { # Specify a secondary loopback address
        unnumbered-address lo0.0 preferred-source-address 192.168.7.4;
      }
    }
  }
}
```

2. Begin configuring hierarchical CoS on the static logical interface set that serves as the hierarchical scheduler node for the group of subscriber local loops.

```
[edit]
class-of-service {
  interfaces {
```

```

interface-set ifset-of-logical-interfaces {
    output-traffic-control-profile tcp-premium-with-4-queues;
}
}
}

```

3. Configure the traffic-control profiles that can be applied to the scheduler node:

```

[edit]
class-of-service {
    traffic-control-profiles {
        tcp-basic-rate { # Specify a scheduler map and traffic controls
            shaping-rate 10m;
        }
        tcp-premium-with-4-queues { # Specify a scheduler map and traffic controls
            scheduler-map smap-premium-4q;
            shaping-rate 20m;
            guaranteed-rate 10m;
            delay-buffer-rate 5m;
        }
    }
}
}

```

In this example, the `tcp-premium-with-4-queues` traffic-control profile is applied to the interface set. The other profile provides a lower shaping rate and no guaranteed rate.

4. Configure the scheduler map `smap-premium-4q` that is referenced in the traffic-control profile for the scheduler node:

```

[edit]
class-of-service {
    scheduler-maps { # Define the queues that comprise each scheduler node
        smap-premium-4q { # Map each queue in the scheduler node to a scheduler
            forwarding-class be scheduler be_sch;
            forwarding-class af scheduler af_sch;
            forwarding-class ef scheduler ef_sch;
            forwarding-class nc scheduler nc_sch;
        }
    }
}
}

```



5. Configure the four schedulers (referenced in the scheduler map) that define the four output queues for the scheduler node:

```
[edit]
class-of-service {
  schedulers { # Define scheduling characteristics of each queue
    be_sch { # Transmit rate and buffer management parameters
      transmit-rate percent 10;
      buffer-size remainder;
      priority low;
    }
    ef_sch { # Transmit rate and buffer management parameters
      ...
    }
    af_sch { # Transmit rate and buffer management parameters
      ...
    }
    nc_sch { # Transmit rate and buffer management parameters
      ...
    }
  }
}
```

6. Enable ANCP to communicate with the DSLAM to adjust the CoS shaping rate for the scheduler node.

You must enable the ANCP feature for performing CoS traffic shaping adjustments, configure the DSLAM as an ANCP neighbor, and specify the DSLAM-assigned identifier for the subscriber local loop represented by the scheduler node. Optionally specify byte or percentage adjustments for frame-mode DSL types.

```
[edit]
protocols {
  ancp {
    qos-adjust; # Enable ANCP to adjust CoS shaping rates and specify rate adjustments
    sdsl-bytes 30;
    sdsl-overhead-adjust 87;
    vdsl-bytes 20;
    vdsl-overhead-adjust 95;
    vdsl2-bytes 20;
    vdsl2-overhead-adjust 87;
```

```

}
neighbor 10.2.3.4; # Configure the DSLAM as an ANCP neighbor
interfaces { # Identify subscribers for which ANCP can adjust shaping rates
  interface-set {
    ifset-of-logical-interfaces {
      access-identifier "dslam port 2/3"; # DSLAM ID for the local loop
    }
  }
}
}
}
}

```



**NOTE:** If ANCP is not yet enabled, the process starts when you commit a configuration that contains the protocols ancp stanza.

7. You can display the configured shaping rate and the adjusted shaping rate for each logical interface set configured for hierarchical CoS, issue the `show class-of-service interface-set operational` command.

## RELATED DOCUMENTATION

[Hierarchical CoS Shaping-Rate Adjustments Overview | 182](#)

[Shaping Rate Adjustments for Subscriber Local Loops Overview | 185](#)

[Guidelines for Configuring Shaping-Rate Adjustments for Subscriber Local Loops | 186](#)

[Enabling Shaping-Rate Adjustments for Subscriber Local Loops | 191](#)

## Verifying the Configuration of Shaping-Rate Adjustments for Subscriber Local Loops

### IN THIS SECTION

● Purpose | 203

● Action | 203

## Purpose

Display the configured shaping rate and the adjusted shaping rate for each logical interface set configured for hierarchical CoS.



**NOTE:** After shaping-rate adjustments are enabled and the router has performed shaping-rate adjustments on a scheduler node, you can configure a new shaping rate by including the `shaping-rate` statement in a traffic-control profile and then applying that profile to that scheduler node. However, this new shaping-rate value does not immediately result in shaping traffic at the new rate. The scheduler node continues to be shaped at rate set by ANCP. Only when the ANCP shaping-rate adjustment feature is disabled is the scheduler node shaped at the newly configured shaping-rate.

## Action

Issue the `show class-of-service interface-set` operational command.

### RELATED DOCUMENTATION

| [Enabling Shaping-Rate Adjustments for Subscriber Local Loops | 191](#)

## Verifying the Configuration of ANCP for Shaping-Rate Adjustments

### IN THIS SECTION

● [Purpose | 203](#)

● [Action | 204](#)

## Purpose

Use to display or clear information about the ANCP configuration for shaping-rate adjustments.

## Action

- To display ANCP neighbor information, issue the **show ancp neighbor** operational command.
- To clear ANCP neighbors, issue the **clear ancp neighbor** operational command.
- To display ANCP subscriber information, issue the **show ancp subscriber** operational command.
- To display ANCP class-of-service information, issue the **show ancp cos** operational command.

If ANCP is not yet enabled, the process starts when you commit a configuration that contains the **protocols ancp** stanza.

## RELATED DOCUMENTATION

*ANCP and the ANCP Agent Overview*

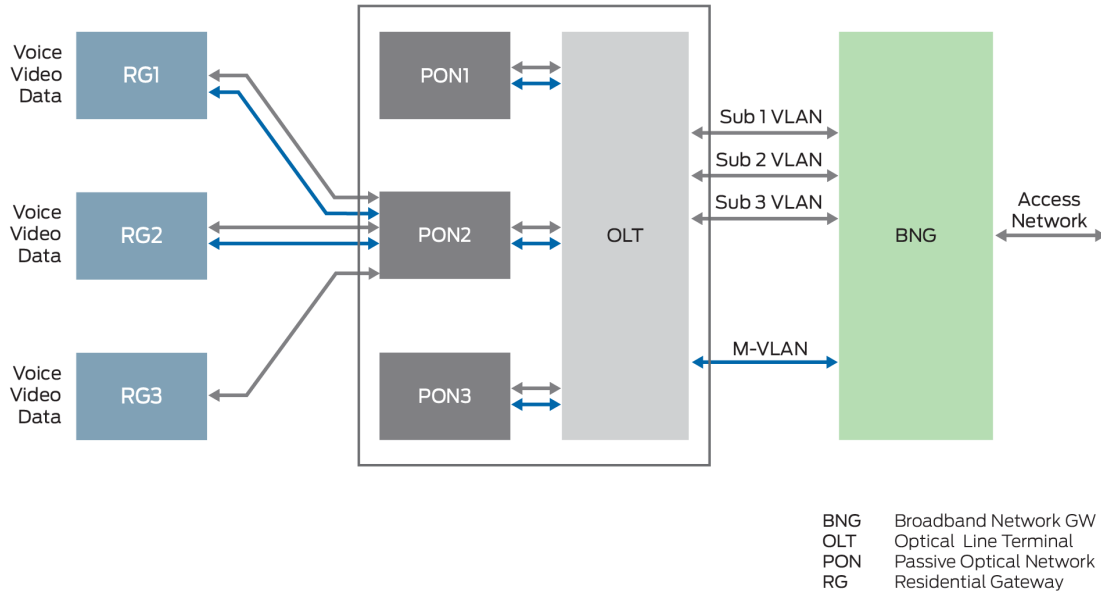
*Configuring the ANCP Agent*

## Using Hierarchical CoS to Adjust Shaping Rates Based on Multicast Traffic

For service providers that are using interface sets to deliver services such as voice and data and multicast VLANs (M-VLANs) to deliver broadcast television, you can set up CoS so that when a subscriber begins receiving multicast traffic, the shaping rate of the subscriber interface is adjusted to account for the multicast traffic. You can also set up the class of service (CoS) multicast adjustment to be propagated from the subscriber interface to the interface set, which is the parent in the scheduler hierarchy. This feature prevents oversubscription of the multicast replicator, such as a PON, which can result in dropped traffic and service disruption.

For broadcast television, instead of transporting separate video streams from the source to each subscriber receiving the same stream, the broadband network gateway (BNG) uses M-VLANs to send one stream to the access node. The access node, such as an Optical Line Terminal (OLT) or a DSLAM, replicates the video stream for each subscriber that is currently watching a particular television channel. In this scenario, M-VLANs are used to stream television and interface sets are used to manage traffic to the access node. An interface set contains an interface for each subscriber that is attached to the access node. See [Figure 25 on page 205](#) for a typical broadcast television network topology.

Figure 25: Typical Broadcast Television Network Topology.



When a subscriber begins watching a television channel, the BNG detects that the subscriber has joined a multicast group and has begun receiving traffic from an M-VLAN. The BNG adjusts traffic shaping on the subscriber interface to account for the bandwidth that the multicast traffic is using. For example, if a subscriber begins watching a television channel, the BNG reduces the bandwidth on the subscriber interface to account for the bandwidth being used by the multicast traffic. If you set up CoS to propagate the reduced bandwidth of the subscriber interface to the interface set, the interface set's shaping rate is also reduced.

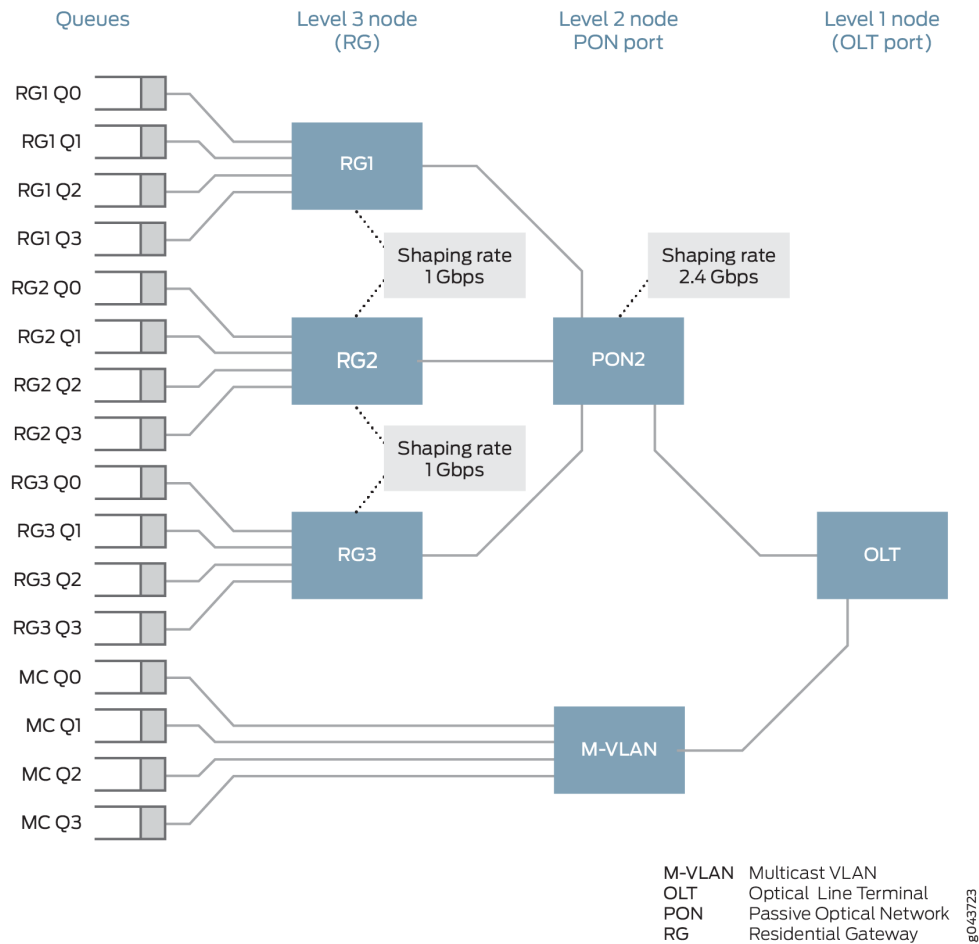
The BNG uses CoS adjustments based on three-level hierarchical CoS scheduling to implement this feature. Level 2 in the scheduler hierarchy is the interface set, and level 3 is the logical interface. To use this feature, you configure traffic-control profiles for logical interfaces (level 3) to adjust the parent interface set (level 2).

You can set up CoS adjustments to the parent interface set to happen once for each broadcast television channel being streamed over the M-VLAN by including the `qos-adjust-hierarchical interface-set` at the `[edit dynamic-profiles dynamic-profile-name access-cac interface $junos-interface-name]` hierarchy. For example:

```
user@host# set dynamic-profiles dynamic-profile-name access-cac interface $junos-interface-name
qos-adjust-hierarchical interface-set
```

To illustrate the benefit of this feature, consider two scenarios from [Figure 26 on page 206](#), which shows a possible subscriber scheduler hierarchy with M-VLANs.

Figure 26: Typical Subscriber Scheduler Hierarchy with M-VLANs



### Scenario 1

Each of 32 subscribers on PON2 are consuming as much data as they can. Each subscriber receives  $(2.4 \text{ Gbps} / 32 \text{ subscribers}) = 75 \text{ Mbps}$ . Subscriber 1 on PON2 starts watching a 120Mbps multicast video stream. Each subscriber receives  $(2.4 \text{ Gbps} - 120 \text{ Mbps}) / 32 \text{ subscribers} = 71.25 \text{ Mbps}$  of data. Subscriber 1 receives 120Mbps of video, and 71.25Mbps of data. Then Subscriber 2 on PON2 starts watching a *different* 120Mbps multicast video stream. Each subscriber then receives  $(2.4 \text{ Gbps} - 240 \text{ Mbps}) / 32 \text{ subscribers} = 67.5 \text{ Mbps}$  of data. Subscribers 1 and 2 receive 120Mbps of video and 67.5Mbps of data.

Final QoS adjustments:

- Subscriber 1's shaping-rate is reduced by 120Mbps to 880Mbps.
- Subscriber 2's shaping-rate is reduced by 120Mbps to 880Mbps.

- PON2's shaping-rate is reduced by 240Mbps to 2.160Gbps, leaving all 32 subscribers to equally share 2.160Gbps – which is 67.5Mbps each.

## Scenario 2

Each of 32 subscribers are consuming as much data as they can. Each subscriber receives  $(2.4 \text{ Gbps} / 32 \text{ subscribers}) = 75\text{Mbps}$ . Subscriber 1 on PON2 starts watching a 120Mbps multicast video stream. Each subscriber then receives  $(2.4\text{Gbps} - 120\text{Mbps}) / 32 \text{ subscribers} = 71.25\text{Mbps}$  of data. Subscriber 1 gets 120Mbps of video, and 71.25Mbps of data. Subscriber 2 on PON2 then starts watching the *same* 120Mbps multicast video stream. This is completely serviced. Since this is the same video stream as Subscriber 1 is watching, and the stream is being replicated at the PON, only one subscriber adjustment is needed. No additional PON adjustment is needed due to subscriber 2 watching the same 120Mbps multicast video steam.

Final QoS adjustments:

- Subscriber 1's shaping-rate is reduced by 120Mbps to 880Mbps.
- Subscriber 2's shaping-rate is reduced by 120Mbps to 880Mbps.
- PON2's shaping-rate is reduced by 240Mbps to 2.280Gbps, leaving all 32 subscribers to equally share 2.280Gbps – which is 71.25Mbps each.

When any one of the two subscribers receiving the 120Mbps video stream unsubscribes, the adjustment for that subscriber is reverted. The other subscriber's adjustment remains, as does the PON's adjustment. When the remaining subscriber unsubscribes the adjustment for that subscriber is reverted, as is the PON's adjustment.

## RELATED DOCUMENTATION

*qos-adjust-hierarchical*

*show access-cac interface-set*

# Configuring Targeted Distribution of Subscribers on Aggregated Ethernet Interfaces

## IN THIS CHAPTER

- [Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 208](#)
- [Providing Accurate Scheduling for a Demux Subscriber Interface of Aggregated Ethernet Links | 212](#)
- [Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces | 213](#)
- [Configuring Link and Module Redundancy for Demux Subscribers in an Aggregated Ethernet Interface | 214](#)
- [Configuring Rebalancing of Demux Subscribers in an Aggregated Ethernet Interface | 215](#)
- [Example: Separating Targeted Multicast Traffic for Demux Subscribers on Aggregated Ethernet Interfaces | 216](#)
- [Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 231](#)
- [Configuring the Distribution Type for PPPoE Subscribers on Aggregated Ethernet Interfaces | 232](#)
- [Verifying the Distribution of PPPoE Subscribers in an Aggregated Ethernet Interface | 233](#)

## Distribution of Demux Subscribers in an Aggregated Ethernet Interface

### IN THIS SECTION

- [Distribution Models | 209](#)
- [Sample Targeted Distribution Topology | 209](#)
- [Redundancy and Redistribution Mechanisms | 210](#)
- [Considerations and Best Practices | 211](#)

This topic describes the distribution options available for demux subscriber interfaces over aggregated Ethernet.



## Distribution Models

By default, the system supports hash-based distribution for all subscriber interface types in an aggregated Ethernet bundle configured without link protection. In this model, traffic for a *logical interface* can be distributed over multiple links in the bundle. This model is desirable when there are many flows through the logical interface and you need to load balance those flows.

Note that if the distribution flows are not even, egress CoS scheduling can be inaccurate. In addition, scheduler resources are required on every link of the aggregated Ethernet interface. For example, if subscriber traffic is allocated 10 MB for a triple-play service over four links in a bundle, each of the links could receive 2.5 MB of traffic. High-density services such as video could be limited by the bandwidth on one of the links.

Targeted distribution enables you to target the egress traffic for an IP or VLAN demux subscriber on a single member link, using a single scheduler. To achieve load balancing over the member links, the system distributes the subscriber interfaces equally among the links. This enables the subscriber that is allocated 10 MB to be accurately scheduled as the traffic flows through.

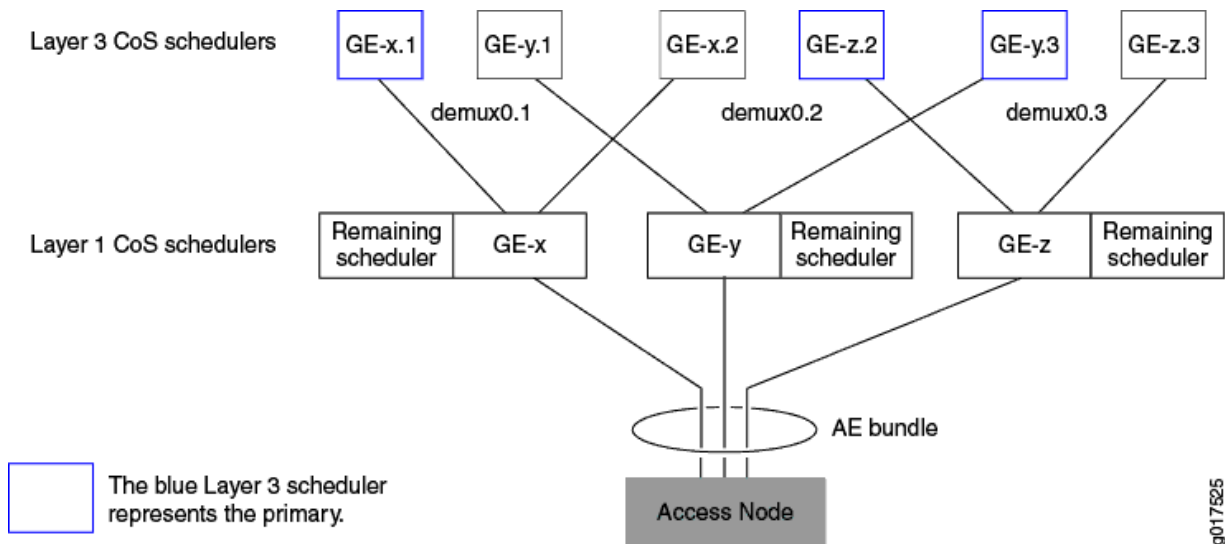


**NOTE:** Every child link in an aggregated Ethernet bundle consumes scheduler resources. That is, for example, a scheduler applied to a single physical interfaces consumes one scheduler resource. A scheduler applied to an aggregated Ethernet interface with five child links consumes five scheduler resources.

## Sample Targeted Distribution Topology

[Figure 27 on page 210](#) displays a sample targeted distribution of subscriber traffic across links in an aggregated Ethernet interface. A primary and backup link is allocated for each subscriber.

Figure 27: Targeted Subscriber Links



For example, if link **GE-x** went down, subscriber 1 can begin forwarding over the backup, which is link **GE-y**. When link **GE-x** comes back up, subscriber 1 switches back to its primary link, **GE-x**.

In the event that both **GE-x** and **GE-y** go down, subscriber 3 starts forwarding through its backup, **GE-z**. Subscriber 1 will have lost its primary and backup links, and will also begin forwarding out the **GE-z** link. A new level 3 scheduler is assigned for this subscriber on link **GE-z**. If there is a momentary lapse between the time that a new scheduler is allocated and forwarding switches to **GE-z**, the traffic will be forwarding through to the remaining scheduler. Subscriber 2 continues to forward through its primary link, **GE-z**.

## Redundancy and Redistribution Mechanisms

Two types of redundancy are available in the targeted distribution model: link redundancy and module redundancy.

By default, an aggregated Ethernet interface is enabled with link redundancy. Backup links for a subscriber are chosen based on the link with the least number of subscribers, which provides redundancy if a link fails.

The module redundancy option enables you to provide redundancy if a module or a link fails. Backup links for a subscriber are chosen on a different DPC or MPC from the primary link, based on the link with the least number of subscribers among the links on different modules. You can enable this for the aggregated Ethernet interface.

When links are removed, affected subscribers are redistributed among the active remaining backup links. When links are added to the system, no automatic redistribution occurs. New subscribers are assigned to the links with the fewest subscribers (which are typically the new links).

## Considerations and Best Practices

Keep the following guidelines in mind when configuring targeted distribution for demux subscribers:

- You can manage subscribers with both hash-based and targeted distribution models in the same network. For example, you can allocate subscribers with interface types such as PPPoE with hash-based distribution, and enable demux subscribers with targeted distribution.
- We recommend that you configure module redundancy to protect against module failures. When module redundancy is enabled, you can ensure an even distribution of subscribers if you allocate no more than 50 percent of the links on a single DPC or MPC.
- During normal network operations, the system maintains an even balance of subscribers among the links in a bundle, even as subscribers log in and out. However, if the distribution of a bundle becomes uneven (for example, when a link goes down and new subscribers are logging in), you can perform a manual rebalance of the bundle. In addition, you can configure periodic rebalancing of the bundle with a specific time interval.
- When you anticipate that a link will be down for an extended time, and you want to ensure that backup links are provisioned for all subscribers, we recommend that you remove the failed link from the bundle. This forces the affected subscribers to redistribute to other links.
- We recommend that you apply a remaining traffic-control profile to the logical interface to ensure that minimal scheduling parameters are applied to the remaining subscriber traffic. This provides scheduling for subscribers that do not have schedulers allocated because they have not been configured or they have been over-provisioned, or because of scheduler transitions on multiple link failures.
- If you perform a cold restart on the router when it is forwarding active subscribers, the subscriber interfaces with targeted distribution are assigned to the first links that become available when the system is initializing so forwarding can begin. To rebalance the system following a cold restart, perform a manual rebalance of the bundle. In addition, we recommend that you configure *Graceful Routing Engine switchover* (GRES) on the router to enable nonstop forwarding during switchover, and avoid performing cold restarts.
- To ensure appropriate and predictable targeted distribution, you must configure chassis network services to use `enhanced-ip` mode.
- Unless specifically separated, multicast traffic egresses in parallel with unicast traffic, sharing the CoS hierarchy and aggregated Ethernet flow distribution.

### RELATED DOCUMENTATION

[Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces](#) | 213

[Configuring Link and Module Redundancy for Demux Subscribers in an Aggregated Ethernet Interface | 214](#)

[Configuring Rebalancing of Demux Subscribers in an Aggregated Ethernet Interface | 215](#)

*Static or Dynamic Demux Subscriber Interfaces over Aggregated Ethernet Overview*

## Providing Accurate Scheduling for a Demux Subscriber Interface of Aggregated Ethernet Links

Unlike VLAN subscriber interfaces, enabling link protection is not required for configuring hierarchical CoS on demux interfaces. Instead, we recommend that you enable targeted distribution on the demux interface to provide accurate scheduling for the aggregated Ethernet links.

Before you begin, configure the subscriber interface with aggregated Ethernet:

- For static and dynamic IP demux interfaces, see *Configuring a Static or Dynamic IP Demux Subscriber Interface over Aggregated Ethernet*.
- For static and dynamic VLAN demux interfaces, see *Configuring a Static or Dynamic VLAN Demux Subscriber Interface over Aggregated Ethernet*.

To provide accurate scheduling for a demux subscriber interface of aggregated Ethernet links:

1. Enable targeted distribution for the demux interface.  
See "[Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces](#)" on page 213.
2. Enable hierarchical scheduling on the link aggregation bundle.  
See "[Configuring Hierarchical CoS for a Subscriber Interface of Aggregated Ethernet Links](#)" on page 101.
3. (Optional) Enable module redundancy to ensure that CoS resources are provisioned for the aggregated Ethernet links if a module or a link fails. By default, link redundancy is supported.  
See "[Configuring Link and Module Redundancy for Demux Subscribers in an Aggregated Ethernet Interface](#)" on page 214.
4. (Optional) Configure rebalancing periodically or manually for the subscribers. See "[Configuring Rebalancing of Demux Subscribers in an Aggregated Ethernet Interface](#)" on page 215.
5. Attach static or dynamic traffic shaping and scheduling parameters at the aggregated Ethernet logical interface or its underlying physical interface. See:
  - *Configuring Traffic Scheduling and Shaping for Subscriber Access*
  - *Configuring Schedulers in a Dynamic Profile for Subscriber Access*
  - *Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile*

- *Applying Minimal Shaping and Scheduling to Remaining Subscriber Traffic*

## RELATED DOCUMENTATION

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

[Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 231](#)

## Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces

By default, the system supports hash-based distribution of subscriber traffic in aggregated Ethernet bundles. You can configure the system to target the egress traffic for a subscriber on a single member link, using a single scheduler resource. The system distributes the subscriber interfaces equally among the member links.

To configure targeted distribution:

1. Edit the chassis hierarchy level.

```
[edit]
user@host#edit chassis
```

2. Enable chassis network services for enhanced-ip mode.

```
[edit chassis]
user@host#set network-services enhanced-ip
```

3. Access the logical interface.

```
[edit]
user@host#edit interfaces demux0 unit logical-unit-number
```

4. Enable targeted distribution for the interface.

```
[edit interfaces demux0 unit logical-unit-number]
user@host#set targeted-distribution
```

## RELATED DOCUMENTATION

[Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 231](#)

[Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 208](#)

## Configuring Link and Module Redundancy for Demux Subscribers in an Aggregated Ethernet Interface

By default, an aggregated Ethernet bundle with targeted distribution is enabled with link redundancy. Backup links for a subscriber are chosen based on the link with the fewest subscribers, which provides redundancy if a link fails.

We recommend that you configure the module redundancy option to provide redundancy if a module or a link fails. Backup links for a subscriber are chosen on a different DPC or MPC from the primary link, based on the link with the fewest subscribers among the links on different modules.

To configure module redundancy for an aggregated Ethernet bundle:

1. Access the aggregated Ethernet bundle for which you want to configure module redundancy.

```
edit
user@host# edit interfaces aex aggregated-ether-options
```

2. Enable module redundancy for the bundle.

```
[edit interfaces aex aggregated-ether-options]
user@host# logical-interface-fpc-redundancy
```

## RELATED DOCUMENTATION

[Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces | 213](#)

[Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 208](#)

## Configuring Rebalancing of Demux Subscribers in an Aggregated Ethernet Interface

### IN THIS SECTION

- [Configuring Periodic Rebalancing of Subscribers in an Aggregated Ethernet Interface | 215](#)
- [Configuring Manual Rebalancing of Subscribers on an Aggregated Ethernet Interface | 216](#)

In a targeted distribution model, the system allocates demux subscriber interfaces equally among the member links in the aggregated Ethernet interface. When links are removed, affected subscribers are redistributed among the active remaining backup links. When links are added to the system, no automatic redistribution occurs. New subscribers are assigned to the links with the fewest subscribers (which are typically the new links).

During normal network operations, the system maintains an even balance of traffic among the links in a bundle, even as subscribers log in and out. However, if the distribution of a bundle becomes uneven (for example, when a link goes down for a period of time and new subscribers are logging in), you can perform a manual rebalance of the bundle. In addition, you can configure periodic rebalancing of the bundle with a specific interval.

### Configuring Periodic Rebalancing of Subscribers in an Aggregated Ethernet Interface

If subscribers are frequently logging in and logging out of your network, you can configure the system to periodically rebalance the links based on a specific time and interval.

To configure periodic rebalancing:

1. Access the aggregated Ethernet interface for which you want to configure periodic rebalancing.

```
edit
user@host# edit interfaces aenumber aggregated-ether-options
```

2. Configure the rebalancing parameters for the interface, including the time and the interval between rebalancing actions.

```
[edit interfaces aenumber aggregated-ether-options]
user@host# rebalance-periodic time hour:minute <interval hours>
```

## SEE ALSO

[Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 231](#)

[Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces | 213](#)

[Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 208](#)

## Configuring Manual Rebalancing of Subscribers on an Aggregated Ethernet Interface

To manually rebalance the subscribers among the links in an aggregated Ethernet bundle with targeted distribution:

- Issue the `request interface rebalance` command:

```
user@host# request interface rebalance interface <interface-name>
```

## SEE ALSO

[Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 231](#)

[Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces | 213](#)

[Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 208](#)

## RELATED DOCUMENTATION

[Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 231](#)

[Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces | 213](#)

[Distribution of Demux Subscribers in an Aggregated Ethernet Interface | 208](#)

## Example: Separating Targeted Multicast Traffic for Demux Subscribers on Aggregated Ethernet Interfaces

### IN THIS SECTION

● [Requirements | 217](#)

● [Overview | 217](#)



- Configuration | 218
- Verification | 226

This example shows how to separate targeted multicast traffic from targeted unicast traffic and send that multicast traffic to a different interface through the use of OIF maps.

## Requirements

Before configuring this example, make sure to configure the distribution type for the interface. See ["Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces"](#) on page 213 for instructions.

## Overview

### IN THIS SECTION

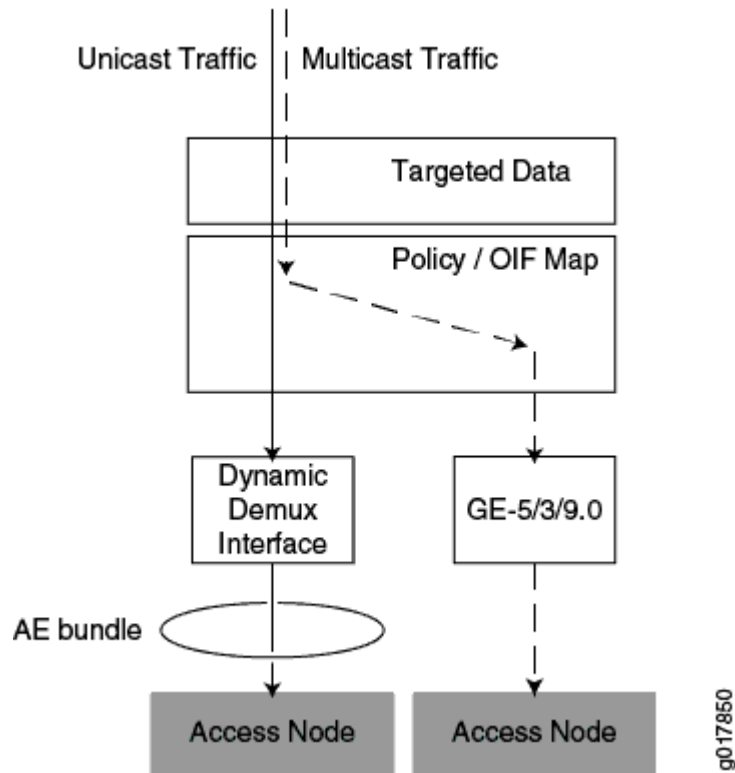
- Topology | 217

In this example, targeted traffic distribution is already configured on the router. Dynamically created interfaces each carry their unicast traffic but all multicast traffic is sent to the ge-5/3/9.0 interface.

## Topology

[Figure 28 on page 218](#) shows the sample network.

Figure 28: Multicast Traffic Separation Using OIF Mapping



## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 219](#)
- [Configure an OIF Map Policy | 219](#)
- [Configure a DHCP VLAN Dynamic Profile | 221](#)
- [Configure a VLAN Demux Dynamic Profile | 222](#)

## CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```

set policy-options policy-statement OIF-v4-all term oif539 from route-filter 192.168.20.0/4
orlonger
set policy-options policy-statement OIF-v4-all term oif539 then map-to-interface ge-5/3/9.0
set policy-options policy-statement OIF-v4-all term oif539 then accept
set dynamic-profiles dhcp-vlan-prof interfaces "$junos-interface-ifd-name" unit "$junos-
underlying-interface-unit" family inet unnumbered-address lo0.0
set dynamic-profiles dhcp-vlan-prof interfaces "$junos-interface-ifd-name" unit "$junos-
underlying-interface-unit" family inet unnumbered-address preferred-sour ce-address 10.20.0.2
set dynamic-profiles demux-vlan-prof interfaces demux0 unit "$junos-interface-un it" vlan-id
"$junos-vlan-id"
set dynamic-profiles demux-vlan-prof interfaces demux0 unit "$junos-interface-un it" demux-
options underlying-interface "$junos-interface-ifd-name"
set dynamic-profiles demux-vlan-prof interfaces demux0 unit "$junos-interface-un it" targeted-
distribution
set dynamic-profiles demux-vlan-prof interfaces demux0 unit "$junos-interface-un it" family inet
unnumbered-address lo0.0
set dynamic-profiles demux-vlan-prof interfaces demux0 unit "$junos-interface-un it" family inet
unnumbered-address preferred-source-address 10.20.0.2
set dynamic-profiles demux-vlan-prof protocols igmp interface "$junos-interface- name" version 2
set dynamic-profiles demux-vlan-prof protocols igmp interface "$junos-interface- name"
promiscuous-mode
set dynamic-profiles demux-vlan-prof protocols igmp interface "$junos-interface- name" passive
allow-receive
set dynamic-profiles demux-vlan-prof protocols igmp interface "$junos-interface- name" passive
send-group-query
set dynamic-profiles demux-vlan-prof protocols igmp interface "$junos-interface- name" oif-map
OIF-v4-all

```

## Configure an OIF Map Policy

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the OIF map:

1. Access the router policy options:

```
[edit]
user@host#edit policy-options
```

2. Edit a policy statement.

```
[edit policy-options]
user@host edit policy-statement OIF-v4-all
```

3. Create a term for mapping incoming multicast traffic to a specific interface.

```
[edit policy-options OIF-v4-all]
user@host edit term oif539
```

4. Define the match condition for the term. In this case, the term matches any route prefix of 192.168.20.0/4 or longer (all multicast traffic).

```
[edit policy-options OIF-v4-all term oif539]
user@host set from route-filter 192.168.20.0/4 orlonger
```

5. Define the action for the term. In this case, when a match occurs, the term accepts the traffic and maps it to interface ge-5/3/9.0.

```
[edit policy-options OIF-v4-all term oif539]
user@host set then map-to-interface ge-5/3/9.0
user@host set then accept
```

## Results

Confirm your configuration by issuing the `show policy-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show policy-options
policy-statement OIF-v4-all {
  term oif539 {
```

```

    from {
        route-filter 192.168.20.0/4 orlonger;
    }
    then {
        map-to-interface ge-5/3/9.0;
        accept;
    }
}
}
}

```

## Configure a DHCP VLAN Dynamic Profile

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure a DHCP VLAN dynamic profile for client access:

1. Create a dynamic VLAN demux profile.

```

[edit]
user@host#edit dynamic-profiles dhcp-vlan-prof

```

2. Edit the dynamic profile interface.

```

[edit dynamic-profiles dhcp-vlan-prof]
user@host edit interfaces $junos-ifd-name

```

3. Edit the interface unit dynamic variable.

```

[edit dynamic-profiles demux-vlan-prof interfaces $junos-ifd-name]
user@host edit unit $junos-underlying-interface-unit

```

4. Edit the interface family.

```

[edit dynamic-profiles demux-vlan-prof interfaces $junos-ifd-name unit $junos-underlying-
interface-unit]
user@host edit family inet

```

## 5. Define the loopback address.

```
[edit dynamic-profiles demux-vlan-prof interfaces $junos-ifd-name unit $junos-underlying-
interface-unit ]
user@host set unnumbered-address lo0.0 preferred-source-address 10.20.0.2
```

## Results

Confirm your configuration by issuing the `show dynamic-profiles` command. If the output for the `dhcp-vlan-prof` dynamic profile does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show dynamic-profiles
dhcp-vlan-prof {
  interfaces {
    "$junos-interface-ifd-name" {
      unit "$junos-underlying-interface-unit" {
        family inet {
          unnumbered-address lo0.0 preferred-source-address 10.20.0.2;
        }
      }
    }
  }
}
```

## Configure a VLAN Demux Dynamic Profile

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure the OIF map:

1. Create a dynamic VLAN demux profile.

```
[edit]
user@host#edit dynamic-profiles demux-vlan-prof
```

2. Edit the dynamic profile demux0 interface.

```
[edit dynamic-profiles demux-vlan-prof]
user@host edit interfaces demux0
```

3. Edit the interface unit dynamic variable.

```
[edit dynamic-profiles demux-vlan-prof interfaces demux0]
user@host edit unit $junos-interface-unit
```

4. Specify the VLAN ID dynamic variable.

```
[edit dynamic-profiles demux-vlan-prof interfaces demux0 unit ♦$junos-interface-unit♦]
user@host set vlan-id $junos-vlan-id
```

5. Access the demux options.

```
[edit dynamic-profiles demux-vlan-prof interfaces demux0 unit ♦$junos-interface-unit♦]
user@host edit demux-options
```

6. Define the demux underlying interface.

```
[edit dynamic-profiles demux-vlan-prof interfaces demux0 unit ♦$junos-interface-unit♦ demux-
options]
user@host set underlying-interface $junos-interface-ifd-name
```

7. Specify that dynamically created VLANs are using targeted distribution.

```
[edit dynamic-profiles demux-vlan-prof interfaces demux0 unit ♦$junos-interface-unit♦]
user@host set targeted-distribution
```

8. Edit the interface family.

```
[edit dynamic-profiles demux-vlan-prof interfaces demux0 unit ♦$junos-interface-unit♦]
user@host edit family inet
```

9. Define the loopback address.

```
[edit dynamic-profiles demux-vlan-prof interfaces demux0 unit ◆$junos-interface-unit◆
family inet]
user@host set unnumbered-address 100.0 preferred-source-address 10.20.0.2
```

10. Edit the dynamic profile IGMP protocol.

```
[edit dynamic-profiles demux-vlan-prof]
user@host edit protocols igmp
```

11. Enable IGMP on dynamically created interfaces.

```
[edit dynamic-profiles demux-vlan-prof protocols igmp]
user@host edit interface $junos-interface-name
```

12. Specify the IGMP version that you want dynamically created interfaces to use.

```
[edit dynamic-profiles demux-vlan-prof protocols igmp interface $junos-interface-name]
user@host set version 2
```

13. Specify the OIF map that you want dynamically created IGMP interfaces to use.

```
[edit dynamic-profiles demux-vlan-prof protocols igmp interface $junos-interface-name]
user@host set oif-map OIF-v4-all
```

14. Specify that IGMP selectively sends and receives control traffic such as IGMP reports, queries, and leaves.

```
[edit dynamic-profiles demux-vlan-prof protocols igmp interface $junos-interface-name]
user@host set passive allow-receive send-group-query
```



15. Specify that the interface accepts IGMP reports from hosts on any subnetwork.

```
[edit dynamic-profiles demux-vlan-prof protocols igmp interface $junos-interface-name]
user@host set promiscuous-mode
```

## Results

Confirm your configuration by issuing the `show dynamic-profiles` commands. If the output for the `dhcp-vlan-prof` dynamic profile does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show dynamic-profiles
demux-vlan-prof {
  interfaces {
    demux0 {
      unit "$junos-interface-unit" {
        vlan-id "$junos-vlan-id";
        demux-options {
          underlying-interface "$junos-interface-ifd-name";
        }
        targeted-distribution;
        family inet {
          unnumbered-address lo0.0 preferred-source-address 10.20.0.2;
        }
      }
    }
  }
  protocols {
    igmp {
      interface "$junos-interface-name" {
        version 2;
        promiscuous-mode;
        passive allow-receive send-group-query;
        oif-map OIF-v4-all;
      }
    }
  }
}
...
```

## Verification

### IN THIS SECTION

- [Locate the Multicast Group Member | 226](#)
- [Ensure the Targeting Aggregated Ethernet Interface for the Subscriber is Functional | 227](#)
- [View the Packets for the Targeted Interface | 228](#)

Confirm that the configuration is working properly.

### Locate the Multicast Group Member

#### Purpose

Locate the dynamic interface and ensure that it is associated with the appropriate IGMP group.

#### Action

```
user@host>show igmp group

Interface: demux0.1073741824, Groups: 1
  Group: 192.0.2.1
    Source: 0.0.0.0
    Last reported by: 10.20.0.10
    Timeout:      52 Type: Dynamic
Interface: local, Groups: 2
  Group: 192.0.2.2
    Source: 0.0.0.0
    Last reported by: Local
    Timeout:      0 Type: Dynamic
  Group: 192.0.2.22
    Source: 0.0.0.0
    Last reported by: Local
    Timeout:      0 Type: Dynamic
```

## Meaning

The first Interface field shows the dynamically created demux interface, `demux0.1073741824`, and the Group field immediately below the first Interface field shows the group, `192.0.2.1`, to which the subscriber belongs.

## Ensure the Targeting Aggregated Ethernet Interface for the Subscriber is Functional

### Purpose

Use the dynamic subscriber interface value to ensure that the targeting aggregated interface is functional.

### Action

```
user@host>show interfaces demux0.1073741824 extensive

Logical interface demux0.1073741824 (Index 810) (SNMP ifIndex 1613)
(Generation 170)
  Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ] Encapsulation: ENET2
  Demux:
    Underlying interface: ae0 (Index 708)
  Link:
    ge-1/0/0
    ge-5/3/7
  Targeting summary:
    ge-1/0/0, backup, Physical link is Up
    ge-5/3/7, primary, Physical link is Up
  Traffic statistics:
    Input bytes :           862
    Output bytes :          3160
    Input packets:           3
    Output packets:          30
  Local statistics:
    Input bytes :           862
    Output bytes :          3160
    Input packets:           3
    Output packets:          30
  Transit statistics:
    Input bytes :           0           0 bps
    Output bytes :           0           0 bps
```

```

Input packets:          0          0 pps
Output packets:        0          0 pps
Protocol inet, MTU: 1500, Generation: 212, Route table: 0
  Flags: Sendbcst-pkt-to-re, Unnumbered
  Donor interface: lo0.0 (Index 802)
  Preferred source address: 10.20.0.2

```

## Meaning

The Targeting summary field shows that the primary interface, `ge-5/3/7`, is up.

## View the Packets for the Targeted Interface

## Purpose

Verify that packet traffic sent to targeted interface `ge-5/3/9` consists only of multicast packets.

## Action

```

user@host>show interfaces ge-5/3/9 extensive
Physical interface: ge-5/3/9, Enabled, Physical link is Up
  Interface index: 704, SNMP ifIndex: 1605, Generation: 197
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled, Auto-negotiation: Enabled, Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues    : 8 supported, 8 maximum usable queues
  Schedulers    : 0
  Hold-times    : Up 0 ms, Down 0 ms
  Current address: 00:21:59:ab:85:2a, Hardware address: 00:21:59:ab:85:2a
  Last flapped  : 2012-09-26 17:32:24 EDT (6d 20:44 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes  :          97857650          1320 bps
    Output bytes :              0          0 bps
    Input packets:          889615          1 pps
    Output packets:          0          889620 pps
  IPv6 transit statistics:
    Input bytes  :              0

```

```

Output bytes :          0
Input packets:          0
Output packets:         0
Dropped traffic statistics due to STP State:
Input bytes :          0
Output bytes :          0
Input packets:         0
Output packets:         0
Input errors:
  Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0,
  L3 incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0,
  FIFO errors: 0, Resource errors: 0
Output errors:
  Carrier transitions: 1, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
  FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Egress queues: 8 supported, 4 in use
Queue counters:      Queued packets  Transmitted packets  Dropped packets
  0 best-effort      0                    0                    0
  1 expedited-fo     0                    0                    0
  2 assured-forw     0                    0                    0
  3 network-cont     0                    0                    0
Queue number:      Mapped forwarding classes
  0                best-effort
  1                expedited-forwarding
  2                assured-forwarding
  3                network-control
Active alarms : None
Active defects : None
MAC statistics:
                Receive          Transmit
Total octets      0          113871616
Total packets    0          889620
Unicast packets  0           0
Broadcast packets  0           0
Multicast packets  0          889620
CRC/Align errors  0           0
FIFO errors       0           0
MAC control frames 0           0
MAC pause frames  0           0
Oversized frames  0           0
Jabber frames     0           0
Fragment frames   0           0
VLAN tagged frames 0           0
Code violations    0           0

```

```

Total errors                                0                0
Filter statistics:
Input packet count                          0
Input packet rejects                        0
Input DA rejects                            0
Input SA rejects                            0
Output packet count                          889620
Output packet pad count                     0
Output packet error count                   0
CAM destination filters: 0, CAM source filters: 0
Autonegotiation information:
Negotiation status: Complete
Link partner:
  Link mode: Full-duplex, Flow control: Symmetric, Remote fault: OK
Local resolution:
  Flow control: None, Remote fault: Link OK
Packet Forwarding Engine configuration:
  Destination slot: 0 (0x00)
CoS information:
Direction : Output
CoS transmit queue      Bandwidth      Buffer Priority  Limit
                        %      bps      %      usec
0 best-effort           95      950000000    95      0      low  none
3 network-control       5       500000000    5       0      low  none
Interface transmit statistics: Disabled

Logical interface ge-5/3/9.0 (Index 818) (SNMP ifIndex 1597) (Generation 149)
Flags: SNMP-Traps 0x4004000 Encapsulation: ENET2
Traffic statistics:
Input bytes :                0
Output bytes :              97857650
Input packets:                0
Output packets:             889620
Local statistics:
Input bytes :                0
Output bytes :                0
Input packets:                0
Output packets:               0
Transit statistics:
Input bytes :                0                0 bps
Output bytes :              97857650          1320 bps
Input packets:                0                0 pps
Output packets:             889615            1 pps

```

```
Protocol aenet, AE bundle: ae4.0, Generation: 180, Route table: 0
```

## Meaning

The MAC statistics `Unicast packet` field shows that the interface is not transmitting any unicast packet traffic and the `Multicast packet` field shows that the total number of packets being transmitted from the interface are multicast packets.

## RELATED DOCUMENTATION

| [Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces](#) | 213

# Verifying the Distribution of Demux Subscribers in an Aggregated Ethernet Interface

## IN THIS SECTION

- [Purpose](#) | 231
- [Action](#) | 231

## Purpose

View the distribution status of subscribers that are targeted to links in an aggregated Ethernet interface.

## Action

- To display a summary of the distribution of links on the demux interface:

```
user@host> show interfaces demux0 extensive
```

- To display the targeted distribution on a specific aggregated Ethernet interface:

```
user@host> show interfaces targeting aex
```

## RELATED DOCUMENTATION

[Configuring the Distribution Type for Demux Subscribers on Aggregated Ethernet Interfaces | 213](#)

[Configuring Rebalancing of Demux Subscribers in an Aggregated Ethernet Interface | 215](#)

## Configuring the Distribution Type for PPPoE Subscribers on Aggregated Ethernet Interfaces

By default, the system supports hash-based distribution of subscriber traffic in aggregated Ethernet bundles. You can configure the system to target the egress traffic for a subscriber on a single member link, using a single scheduler resource. The system distributes the subscriber interfaces equally among the member links.

To configure targeted distribution:

1. Edit the chassis hierarchy level.

```
[edit]  
user@host#edit chassis
```

2. Enable chassis network services for enhanced-ip mode.

```
[edit chassis]  
user@host#set network-services enhanced-ip
```

3. Access the logical interface.

```
[edit]  
user@host#edit interfaces pp0 unit logical-unit-number
```



4. Enable targeted distribution for the interface.

```
[edit interfaces pp0 unit logical-unit-number]  
user@host#set targeted-distribution
```

## RELATED DOCUMENTATION

[CoS for PPPoE Subscriber Interfaces Overview](#)

[Verifying the Distribution of PPPoE Subscribers in an Aggregated Ethernet Interface | 233](#)

## Verifying the Distribution of PPPoE Subscribers in an Aggregated Ethernet Interface

### IN THIS SECTION

- [Purpose | 233](#)
- [Action | 233](#)

### Purpose

View the distribution status of subscribers that are targeted to links in an aggregated Ethernet interface.

### Action

- To display a summary of the distribution of links on the demux interface:

```
user@host> show interfaces pp0 extensive
```

- To display the targeted distribution on a specific aggregated Ethernet interface:

```
user@host> show interfaces targeting aex
```

## RELATED DOCUMENTATION

*CoS for PPPoE Subscriber Interfaces Overview*

---

[Configuring the Distribution Type for PPPoE Subscribers on Aggregated Ethernet Interfaces | 232](#)

# Applying CoS Using Parameters Received from RADIUS

## IN THIS CHAPTER

- [Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS | 235](#)
- [Changing CoS Services Overview | 240](#)
- [CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview | 244](#)
- [Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions | 246](#)
- [Configuring Initial CoS Parameters Dynamically Obtained from RADIUS | 247](#)
- [Configuring Static Default Values for Traffic Scheduling and Shaping | 248](#)
- [Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions | 250](#)
- [CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets | 253](#)
- [Example: Configuring Dynamic Hierarchical Scheduling for Subscribers | 259](#)

## Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS

### IN THIS SECTION

- [Dynamic Configuration of Initial CoS in Access Profiles | 236](#)
- [Predefined Variables for Dynamic Configuration of Initial Traffic Shaping | 236](#)
- [Predefined Variables for Dynamic Configuration of Initial Scheduling and Queuing | 237](#)

You can configure interface-specific CoS parameters that the router obtains when subscribers log in at appropriately configured static or dynamic subscriber interfaces. This feature is supported only for

interfaces on Enhanced Queuing Dense Port Concentrators (EQ DPCs) in MX Series 5G Universal Routing Platforms.

To configure a dynamic profile to provide initial CoS Services, make sure you understand the following concepts:

## Dynamic Configuration of Initial CoS in Access Profiles

When a router interface receives a join message from a DHCP subscriber, the Junos OS applies the values configured in the dynamic profile associated with that router interface. A dynamic profile that is activated through its association with a subscriber interface is known as an *access dynamic profile*. You can associate a dynamic profile with a subscriber interface on the router by including statements at the [edit dynamic-profiles *profile-name* class-of-service interfaces] hierarchy level.

The Junos OS supports predefined variables for obtaining CoS parameters from the RADIUS authentication server. When a client authenticates over a router interface associated with the access dynamic profile, the router replaces the predefined variables with interface-specific values obtained from the RADIUS server.



**NOTE:** To associate dynamically configured initial CoS features with a subscriber interface, reference *Junos OS predefined variables*—and not *user-defined variables*—in an *access* dynamic profile for that interface.

## Predefined Variables for Dynamic Configuration of Initial Traffic Shaping

You can configure an access dynamic profile that provides initial traffic-shaping parameters when a subscriber logs in. The Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.

If you define the Juniper Networks authentication and authorization VSA for CoS traffic-shaping parameter values (attribute number 26–108) on the RADIUS authentication server, the RADIUS server includes the values in RADIUS Access-Accept messages it sends to the router when a subscriber successfully authenticates over the interface.

To provide an initial scheduler map name and traffic shaping parameters obtained from the RADIUS authentication server when a subscriber logs in, reference the Junos OS predefined variables for CoS listed in [Table 18 on page 237](#) in an access dynamic profile associated with the subscriber interface.

**Table 18: CoS Predefined Variables for Scheduler Map and Traffic Shaping**

Variable	Description
\$junos-cos-scheduler-map	<p>Scheduler-map name to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in.</p> <p>NOTE: The scheduler map referenced by the <code>scheduler-map</code> statement can be defined dynamically (at the <code>[edit dynamic-profiles profile-name class-of-service scheduler-maps]</code> hierarchy level) or statically (at the <code>[edit class-of-service scheduler-maps]</code> hierarchy level).</p>
\$junos-cos-shaping-rate	<p>Shaping rate to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.</p>
\$junos-cos-guaranteed-rate	<p>Guaranteed rate to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.</p>
\$junos-cos-delay-buffer-rate	<p>Delay-buffer rate to be dynamically configured in a traffic-control profile in the access dynamic profile when a subscriber logs in. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.</p>

### Predefined Variables for Dynamic Configuration of Initial Scheduling and Queuing

You can configure an access dynamic profile that provides initial traffic-shaping parameters when a subscriber logs in. The Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.

If you define the Juniper Networks authentication and authorization VSA for CoS scheduling and queuing parameter values (attribute number 26–146) on the RADIUS authentication server, the RADIUS server includes the values in RADIUS Access-Accept messages it sends to the router when a subscriber successfully authenticates over the interface.

To provide an initial scheduler name and scheduler and queuing parameters obtained from the RADIUS authentication server when a subscriber logs in, reference the Junos OS predefined variables listed in [Table 19 on page 238](#) in an access dynamic profile associated with the subscriber interface.

**Table 19: CoS Predefined Variables for Scheduling and Queuing**

Variable	Description
\$junos-cos-scheduler	Name of a scheduler to be dynamically configured in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
\$junos-cos-scheduler-transmit-rate	Transmit rate to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
\$junos-cos-scheduler-bs	Buffer size, as a percentage of total buffer, to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
\$junos-cos-scheduler-pri	Packet-scheduling priority value to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.
\$junos-cos-scheduler-dropfile-low	Name of the drop profile for RED for loss-priority level low to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.  NOTE: The drop profile must be configured statically (at the [edit class-of-service schedulers <i>scheduler-name</i> drop-profiles] hierarchy level) for loss-priority low.

Table 19: CoS Predefined Variables for Scheduling and Queuing (Continued)

Variable	Description
\$junos-cos-scheduler-dropfile-medium-low	<p>Name of the drop profile for RED for loss-priority level medium-low to be dynamically configured for the scheduler in the access dynamic profile. The Junos OS obtains this information from the RADIUS server when a subscriber authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.</p> <p>NOTE: The drop profile must be configured statically (at the [edit class-of-service schedulers <i>scheduler-name</i> drop-profiles] hierarchy level).</p>
\$junos-cos-scheduler-dropfile-medium-high	<p>Name of the drop profile for RED for loss-priority level medium-high to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.</p> <p>NOTE: The drop profile must be configured statically (at the [edit class-of-service schedulers <i>scheduler-name</i> drop-profiles] hierarchy level).</p>
\$junos-cos-scheduler-dropfile-high	<p>Name of the drop profile for RED for loss-priority level high to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.</p> <p>NOTE: The drop profile must be configured statically (at the [edit class-of-service schedulers <i>scheduler-name</i> drop-profiles] hierarchy level).</p>
\$junos-cos-scheduler-dropfile-any	<p>Name of the drop profile for RED for loss-priority level any to be dynamically configured for the scheduler in the access dynamic profile. You can configure a RADIUS authentication server to include this information in the Accept-Accept message when a subscriber successfully authenticates over the static or dynamic subscriber interface to which the access dynamic profile is attached.</p> <p>NOTE: The drop profile must be configured statically (at the [edit class-of-service schedulers <i>scheduler-name</i> drop-profiles] hierarchy level).</p>

## RELATED DOCUMENTATION

*Subscriber Activation and Service Management in an Access Network*

*Dynamic Profiles Overview*

*Dynamic Variables Overview*

*Junos OS Predefined Variables*

*Configuring Initial CoS Parameters Dynamically Obtained from RADIUS*

*Example: Configuring Initial CoS Parameters Dynamically Obtained from RADIUS*

## Changing CoS Services Overview

### IN THIS SECTION

- [Types of CoS Variables Used in a Service Profile | 241](#)
- [Static and Dynamic CoS Configurations | 241](#)
- [Scenarios for Static and Dynamic Configuration of CoS Parameters | 242](#)

This topic describes how to provide CoS when subscribers dynamically upgrade or downgrade services in an access environment.

You can configure your network with a *dynamic client profile* that provides all subscribers with default CoS parameters when they log in. For example, all subscribers can receive a basic data service. By configuring the client profile with Junos OS predefined variables for RADIUS-provided CoS parameters, you also enable the service to be activated for those subscribers at login.



**NOTE:** The dynamic client profile is also referred to as a dynamic client access profile, or sometimes just access profile for brevity. Do not confuse this profile, configured at the [edit dynamic-profiles *profile-name*] hierarchy level, with the access profile configured at the [edit access profile *profile-name*] hierarchy level. These static access profiles are used to configure authentication, accounting, and authorization parameters for subscriber access, some session attributes, and client-specific properties for L2TP and PPP sessions. Access profiles are applied at various configuration levels with the access-profile statement.

To enable subscribers to activate a service or upgrade to different services through RADIUS change-of-authorization (CoA) messages after login, configure a *dynamic service profile* that includes user-defined variables.



## Types of CoS Variables Used in a Service Profile

You can configure variables for the following CoS parameters in a service profile:

- Shaping rate
- Delay buffer rate
- Guaranteed rate
- Scheduler map

For each CoS parameter, you must associate a RADIUS vendor ID. For each vendor ID, you must assign an attribute number and a tag. The tag is used to differentiate between values for different CoS variables when you specify the same attribute number for those variables. These values are matched with the values supplied by RADIUS during subscriber authentication. All of the values in the dynamic profile must be defined in RADIUS or none of the values are passed.

Optionally, you can configure default values for each parameter. Configuring default values is beneficial if you do not configure RADIUS to enable service changes. During service changes, RADIUS takes precedence over the default value that is configured.

## Static and Dynamic CoS Configurations

Depending on how you configure CoS parameters in the access and service profiles, certain CoS parameters are replaced or merged when subscribers change or activate new services.

Static configuration is when you configure the scheduler map and schedulers in the static [edit class-of-service] hierarchy and reference the scheduler map in the dynamic profile. Dynamic configuration is when you configure the scheduler map and schedulers within the dynamic profile.

The CoS configuration also depends on whether you have enabled multiple subscribers on the same *logical interface* using the `aggregate-clients` statements in the dynamic profile referenced by DHCP. When you specify the `aggregate-clients replace` statement, the scheduler map names are replaced. In both cases, if the length of the scheduler map name exceeds 128 characters, subscribers cannot log in. When you specify the `aggregate-clients merge` statement, the scheduler map names specified in the dynamic profile are appended.



**BEST PRACTICE:** To improve CoS performance in IPv4, IPv6, and dual-stack networks, we recommend that you use the `aggregate-clients replace` statement rather than the `aggregate-clients merge` statement.

## Scenarios for Static and Dynamic Configuration of CoS Parameters

Table 20 on page 242 lists the scenarios for static and dynamic configuration of CoS parameters in access profiles and service profiles at subscriber login. The table also lists the behavior for each configuration for service activation and service modification using RADIUS CoA messages.

**Table 20: CoS Services and Variables**

Scenario	Static CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the aggregate-clients merge Statement)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the aggregate-clients replace Statement)
Subscriber login	<ul style="list-style-type: none"> <li>Configure RADIUS values or default values for all parameters in access profile</li> <li>Configure scheduler map in edit class-of-service hierarchy and reference in access profile</li> </ul>	<ul style="list-style-type: none"> <li>Configure RADIUS values or default values for all parameters in access profile</li> <li>Configure scheduler map and schedulers in access profile</li> </ul>	<ul style="list-style-type: none"> <li>Configure RADIUS values or default values for all parameters in access profile</li> <li>Configure scheduler map and schedulers in access profile</li> </ul>	<ul style="list-style-type: none"> <li>Configure RADIUS values or default values for all parameters in access profile</li> <li>Configure scheduler map and schedulers in access profile</li> </ul>

Table 20: CoS Services and Variables (Continued)

Scenario	Static CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Single Subscriber)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the aggregate-clients merge Statement)	Dynamic CoS Configuration (Multiple Subscribers Enabled on a Logical Interface with the aggregate-clients replace Statement)
RADIUS CoA for service or variable change	Replaces the following parameters: <ul style="list-style-type: none"> <li>• Delay buffer rate</li> <li>• Guaranteed rate</li> <li>• Scheduler map</li> <li>• Shaping rate</li> </ul>	Replaces the following parameters: <ul style="list-style-type: none"> <li>• Delay buffer rate</li> <li>• Guaranteed rate</li> <li>• Shaping rate</li> <li>• Scheduler map</li> </ul>	Combines the values of the following parameters to their maximum scalar value: <ul style="list-style-type: none"> <li>• Delay buffer rate</li> <li>• Guaranteed rate</li> <li>• Shaping rate</li> </ul> Appends the scheduler map parameter	Replaces the following parameters: <ul style="list-style-type: none"> <li>• Delay buffer rate</li> <li>• Guaranteed rate</li> <li>• Shaping rate</li> <li>• Scheduler map</li> </ul>
RADIUS CoA for service activation	Does not merge queues  NOTE: In this case, use a similar configuration to the access profile, including the same name for the traffic-control-profile. During service activation, this configuration replaces the original configuration in the access profile.	Merge queues if the queue specified in the service profile is not already in use for the subscriber  NOTE: Do not instantiate a CoA request using a service dynamic profile that is already in use on the same logical interface.	Merge queues if the queue specified in the service profile is not already in use for the subscriber  NOTE: Do not instantiate a CoA request using a service dynamic profile that is already in use on the same logical interface.	Merge queues if the queue specified in the service profile is not already in use for the subscriber  NOTE: Do not instantiate a CoA request using a service dynamic profile that is already in use on the same logical interface.

## RELATED DOCUMENTATION

[Configuring Static Hierarchical Scheduling in a Dynamic Profile](#)

[Dynamic Profile Attachment to DHCP Subscriber Interfaces Overview](#)

[RADIUS Attributes and Juniper Networks VSAs Supported by the AAA Service Framework](#)

[Guidelines for Configuring Dynamic CoS for Subscriber Access](#)

## CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview

### IN THIS SECTION

- [Supported Network Configurations | 244](#)
- [Traffic-Control Profiles in Subscriber Interface Dynamic Profiles | 245](#)
- [CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets and Member Subscriber Sessions | 245](#)

To control bandwidth at a household level in a subscriber access network, you can apply RADIUS dynamic *class of service* (CoS) traffic-shaping attributes to a dynamic interface set and its member subscriber sessions when the subscriber sessions are authenticated. (The dynamic interface set itself does not go through the authentication process.)

A *household* is represented by either a dynamic interface set or a dynamic agent-circuit-identifier (ACI) interface set from which the subscriber sessions originate. For this feature, dynamic interface sets and dynamic ACI interface sets are mapped to Level 2 of the Junos OS CoS scheduler hierarchy, which enables you to use CoS traffic-shaping to shape the bandwidth at the household (interface set) level.

The *subscriber sessions*, also referred to as *subscriber interfaces* or *client sessions*, can be dynamic VLAN, PPPoE, or IP demultiplexing (IP demux) subscriber interfaces. The subscriber interfaces are mapped to Level 3 of the Junos OS CoS scheduler hierarchy.

### Supported Network Configurations

Applying RADIUS dynamic CoS traffic-shaping attributes to a dynamic interface set and its member subscriber sessions is supported for the following network configurations:

- Dynamic IP demux subscriber interfaces (for DHCP subscribers) over either a dynamic interface set or a dynamic ACI interface set

- Dynamic PPPoE subscriber interfaces over either a dynamic interface set or a dynamic ACI interface set

## Traffic-Control Profiles in Subscriber Interface Dynamic Profiles

To apply dynamic CoS traffic-shaping attributes to a dynamic interface set and its member subscriber sessions, you must define and attach the traffic-control profiles for *both* the dynamic interface set and the dynamic subscriber sessions within the dynamic profile for the subscriber interface.

At the [edit dynamic-profiles *profile-name* class-of-service traffic-control-profiles] hierarchy level in the dynamic profile, configure both of the following:

- Traffic-control profile for the dynamic VLAN, PPPoE, or IP demux subscriber interfaces
- Traffic-control profile for the dynamic interface set or dynamic ACI interface set to which the subscriber interfaces belong

RADIUS tag values for the Junos OS CoS traffic shaping predefined variables used in both traffic-control profiles must be in the 100s range, as described in *CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets*.

At the [edit dynamic-profiles *profile-name* interfaces] hierarchy level in the dynamic profile, use the output-traffic-control-profile statement to apply the traffic-control profiles to the dynamic subscriber interface and the dynamic interface set or dynamic ACI interface set.

## CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets and Member Subscriber Sessions

The set of \$junos-cos-*parameter* predefined dynamic variables has been duplicated and assigned a RADIUS tag value in the 100s range for use with this feature. The RADIUS tag value is the only difference between the existing CoS traffic-shaping predefined dynamic variables and the predefined dynamic variables that you must use with this feature.

Both RADIUS instances of the \$junos-cos-*parameter* predefined dynamic variables are available, but you must use the dynamic variables with tag values in the 100s range to apply CoS traffic-shaping attributes to both the dynamic interface set and member subscriber sessions in a subscriber interface dynamic profile.

For example, the existing \$junos-cos-shaping-rate predefined variable is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 2. To apply CoS traffic-shaping attributes to the dynamic interface set and its member subscriber sessions, you must instead use the \$junos-cos-shaping-rate predefined variable that is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 102.



**NOTE:** Do not configure a combination of `$junos-cos-parameter` predefined dynamic variables with RADIUS tag values in the 100s range and `$junos-cos-parameter` predefined dynamic variables with tag values not in the 100s range in the same traffic-control profile. If you do so, the subscriber authentication process fails.

## RELATED DOCUMENTATION

*Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions*

*Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions*

*CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets*

## Guidelines for Configuring CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions

Observe the following guidelines when you apply dynamic CoS traffic-shaping attributes to a dynamic interface set or a dynamic ACI interface set and its member subscriber sessions. For complete information about the Junos OS CoS traffic-shaping predefined dynamic variables and RADIUS tag values used with this feature, see *CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets*.

- This feature is supported only for dynamically configured and instantiated subscriber interfaces.
- Do not configure a combination of `$junos-cos-parameter` predefined dynamic variables with RADIUS tag values in the 100s range and `$junos-cos-parameter` predefined dynamic variables with tag values not in the 100s range in the same traffic-control profile. If you do so, the subscriber authentication process fails.
- Use the `$junos-cos-adjust-minimum` predefined variable (tag 109) only in traffic-control profiles for dynamic subscriber interfaces. Using this variable in a traffic-control profile for a dynamic interface set or dynamic ACI interface set has no effect.
- Do not configure the `$junos-cos-excess-rate-high` predefined variable (tag 110) when the `$junos-cos-excess-rate` predefined variable (tag 105) is configured, and vice-versa.
- Do not configure the `$junos-cos-excess-rate-low` predefined variable (tag 111) when the `$junos-cos-excess-rate` predefined variable (tag 105) is configured, and vice-versa.
- Do not configure the `$junos-cos-byte-adjust-frame` predefined variable (tag 114) when the `$junos-cos-byte-adjust` predefined variable (tag 108) is configured, and vice-versa.

- Do not configure the `$junos-cos-byte-adjust-cell` predefined variable (tag 115) when the `$junos-cos-byte-adjust` predefined variable (tag 108) is configured, and vice-versa.
- Use the per-priority `$junos-cos-shaping-rate-parameter` predefined variables (tags 116 through 125) only in traffic-control profiles for dynamic interface sets or dynamic ACL interface sets. Using these variables in traffic-control profiles for a dynamic logical subscriber interface causes the subscriber session to fail.

## RELATED DOCUMENTATION

*Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions*

*CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets*

*CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview*

## Configuring Initial CoS Parameters Dynamically Obtained from RADIUS

You can configure a subscriber interface so that subscribers receive initial CoS parameters that the router obtains from the RADIUS authentication server when subscribers log in using that logical interface on the router.

1. Configure external RADIUS server VSAs with values that you expect subscribers to log in with.
  - To configure a RADIUS authentication server to include CoS traffic-shaping parameters in authentication grants on certain subscriber interfaces, configure Juniper Networks VSA 26–108.
  - To configure a RADIUS authentication server to include CoS scheduling and queuing parameters in authentication grants a certain subscriber interfaces, configure Juniper Networks VSA 28–146.

See [Configuring Router or Switch Interaction with RADIUS Servers](#) and *RADIUS Servers and Parameters for Subscriber Access*.

2. Configure a subscriber interface that supports hierarchical CoS.
3. Associate a traffic-control profile with the interface.
 

See *Applying Traffic Shaping and Scheduling to a Subscriber Interface in a Dynamic Profile*.
4. Configuring initial traffic-shaping parameters to be obtained from RADIUS.
 

See *Configuring Dynamic Traffic Shaping and Scheduling Parameters in a Dynamic Profile*.
5. Configure forwarding classes and scheduler maps statically.
 

See [Configuring a Custom Forwarding Class for Each Queue](#) and [Configuring Scheduler Maps](#).
6. Configure a scheduler to specify initial scheduling and queuing parameters to be dynamically obtained from RADIUS when a subscriber logs in.

See *Configuring Dynamic Schedulers with Variables in a Dynamic Profile*.

## RELATED DOCUMENTATION

*Subscriber Interfaces That Provide Initial CoS Parameters Dynamically Obtained from RADIUS*

*Example: Configuring Initial CoS Parameters Dynamically Obtained from RADIUS*

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

*Subscriber Activation and Service Management in an Access Network*

*Juniper Networks VSAs Supported by the AAA Service Framework*

*Dynamic Profiles Overview*

*Dynamic Variables Overview*

*Junos OS Predefined Variables*

## Configuring Static Default Values for Traffic Scheduling and Shaping

To provide subscribers with default values for CoS parameters, configure user-defined variables for CoS parameters and assign static default values to the variables. If you have configured values to be supplied by a RADIUS CoA, subscribers receive the default value when deactivating a service.

To configure user-defined variables with default values for CoS in a dynamic profile:

1. Specify that you want to configure variables in the dynamic profile.

```
[edit dynamic-profiles residential-silver variables]
```

2. Configure a default value for the shaping rate.

```
[edit dynamic-profiles residential-silver variables]
user@host# set srate default-value 5m
```

3. Configure a default value for the guaranteed rate.

```
[edit dynamic-profiles residential-silver variables]
user@host# set grate default-value 5m
```



4. Configure a default value for the delay buffer rate.

```
[edit dynamic-profiles residential-silver variables]
user@host# set dbrate default-value 10m
```

5. Configure a default value for the scheduler map.

```
[edit dynamic-profiles residential-silver variables]
user@host# set smap default-value triple-play
```

6. Configure the variables for the CoS parameters in the traffic-control profile.  
Either the shaping rate or the guaranteed rate is required in the traffic-control profile.

- a. Access the traffic-control profile in the dynamic profile.

```
user@host# edit dynamic-profiles residential-silver class-of-service traffic-control-
profiles tcp1
```

- b. Configure the scheduler map variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set scheduler-map "$smap"
```

- c. Configure the shaping rate variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set shaping-rate "$rate"
```

- d. Configure the guaranteed rate variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set guaranteed-rate "$grate"
```

- e. Configure the delay buffer rate variable.

```
[edit dynamic-profiles residential-silver class-of-service traffic-control-profiles tcp1]
user@host# set delay-buffer-rate "$dbrate"
```

## RELATED DOCUMENTATION

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

*Changing CoS Services Overview*

## Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions

To control bandwidth at a household level in a subscriber access network, you can apply RADIUS dynamic class of service (CoS) traffic-shaping attributes to a dynamic interface set or agent-circuit-identifier (ACI) interface set and its member subscriber sessions when the member sessions are authenticated. The dynamic interface set or ACI interface set represents the *household* from which the subscriber sessions originate. The *subscriber sessions*, also referred to as *client sessions* or *subscriber interfaces*, can be dynamic VLAN, PPPoE, or IP demultiplexing (IP demux, for DHCP) subscriber interfaces.

To apply RADIUS dynamic CoS traffic-shaping attributes to both the dynamic interface set and its member subscriber sessions, you must configure two traffic-control profiles in the dynamic profile for the subscriber interface: one traffic-control profile for the “parent” dynamic interface set, and a second traffic-control profile for the dynamic subscriber interfaces. RADIUS tag values for the Junos OS CoS traffic shaping predefined variables used in both traffic-control profiles must be in the 100s range.

Before you begin:

- Create a dynamic profile that defines the VLAN, PPPoE, or IP demux logical subscriber interface.

See the following topics:

- *Configuring a Basic Dynamic Profile*
- *Configuring a Dynamic Profile Used to Create Single-Tag VLANs*
- *Configuring a Dynamic Profile Used to Create Stacked VLANs*
- *Configuring Dynamic PPPoE Subscriber Interfaces*
- *Configuring Dynamic Subscriber Interfaces Using IP Demux Interfaces in Dynamic Profiles*

To apply dynamic CoS traffic-shaping attributes to a dynamic ACI or non-ACI interface set and its member subscriber sessions in a dynamic profile for the subscriber interface:

1. Configure two traffic-control profiles at the [edit dynamic-profiles *profile-name* class-of-service traffic-control profiles] hierarchy level:
  - Traffic-control profile for the VLAN, PPPoE, or IP demux dynamic subscriber interfaces

- Traffic-control profile for the dynamic interface set or dynamic ACI interface set to which the subscriber interfaces belong
2. In the traffic-control profiles configured for the dynamic interface set and the subscriber interfaces, reference Junos OS CoS traffic-shaping predefined variables with RADIUS tag values in the 100s range.  
See *CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets* for a complete list of the Junos OS predefined variables and RADIUS tag values that you must use in the traffic-control profiles for the dynamic subscriber interfaces and the dynamic interface set.
  3. At the [edit dynamic-profiles *profile-name* interfaces] hierarchy level, use the output-traffic-control-profile statement to apply the traffic-control profiles to the dynamic subscriber interface and the dynamic interface set or dynamic ACI interface set.

### Example: Dynamic PPPoE Subscriber Interface over Dynamic ACI Interface Set

The following example shows a dynamic profile named pppoe-subscriber that configures a dynamic PPPoE (pp0) subscriber interface over a dynamic ACI interface set.

The traffic-control-profiles stanza defines two traffic-control profiles: tcp-pppoe-session for the dynamic PPPoE subscriber interface, and tcp-parent-aci-set for the dynamic “parent” ACI interface set. The \$junos-cos-shaping-rate predefined variable included in each of these traffic-control profiles is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 102. The \$junos-cos-shaping-mode variable is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 107.

The interfaces stanza applies output traffic-control profile tcp-pppoe-session to the dynamic PPPoE (pp0) subscriber interface, and output traffic-control profile tcp-parent-aci-set to the dynamic ACI interface set.

```
[edit dynamic-profiles]
pppoe-subscriber {
  interfaces {
    interface-set "$junos-interface-set-name" {
      interface pp0 {
        unit "$junos-interface-unit";
      }
    }
    pp0 {
      unit "$junos-interface-unit" {
        ppp-options {
          pap;
        }
        pppoe-options {
          underlying-interface "$junos-underlying-interface";
          server;
        }
      }
    }
  }
}
```

```

        }
        no-keepalives;
        family inet {
            unnumbered-address lo0.0;
        }
    }
}
}
class-of-service {
    traffic-control-profiles {
        tcp-pppoe-session {
            scheduler-map smap-1;
            shaping-rate $junos-cos-shaping-rate;
            overhead-accounting $junos-cos-shaping-mode frame-mode-bytes -4 cell-mode-bytes
12;
        }
        tcp-parent-aci-set {
            shaping-rate $junos-cos-shaping-rate;
            overhead-accounting $junos-cos-shaping-mode frame-mode-bytes -4 cell-mode-bytes
12;
        }
    }
}
interfaces {
    pp0 {
        unit "$junos-interface-unit" {
            output-traffic-control-profile tcp-pppoe-session;
        }
    }
    interface-set $junos-interface-set-name {
        output-traffic-control-profile tcp-parent-aci-set;
    }
}
}
}
}
}

```

## RELATED DOCUMENTATION

*CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets*

*CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview*

## CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets

To control bandwidth at a household level in a subscriber access network, you can apply RADIUS CoS traffic-shaping attributes to a dynamic interface set and its member subscriber sessions when the member sessions are authenticated. The dynamic interface set, which represents the household level in a subscriber access network, can be either a dynamic agent-circuit-identifier (ACI) interface set or a non-ACI-based dynamic interface set. The subscriber sessions belonging to the interface set can be dynamic VLAN, DHCP, or PPPoE subscriber interfaces.

To apply RADIUS CoS traffic-shaping attributes to both the dynamic interface set and its member subscriber sessions, you must configure two traffic-control profiles in the dynamic profile for the subscriber interface: one traffic-control profile for the “parent” dynamic interface set, and a second traffic-control profile for the dynamic subscriber interfaces. RADIUS tag values for the Junos OS CoS traffic-shaping predefined variables used in these traffic-control-profiles must be in the 100s range, as described in [Table 21 on page 254](#).

To accommodate this feature, the set of existing `$junos-cos-parameter` predefined dynamic variables for traffic shaping have been duplicated and assigned a tag value in the 100s range, as listed in [Table 21 on page 254](#). The tag value is the only difference between the existing predefined dynamic variables and the predefined dynamic variables that you must use with this feature.

For example, the existing `$junos-cos-shaping-rate` predefined variable is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 2. To apply RADIUS CoS traffic-shaping attributes to the dynamic interface set and its member subscriber sessions, you must instead use the `$junos-cos-shaping-rate` predefined variable that is assigned RADIUS vendor ID 4874, attribute number 108, and tag value 102.

[Table 21 on page 254](#) describes the Junos OS predefined dynamic variables and RADIUS tag values that you can use in a dynamic profile to apply RADIUS CoS traffic-shaping attributes to the dynamic interface set and its member subscriber sessions. The table lists the predefined dynamic variables in ascending order by tag value.



**NOTE:** All of the predefined variables listed in [Table 21 on page 254](#) use RADIUS vendor ID 4874 and RADIUS attribute value 108.

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets

Predefined Variable	RADIUS Tag Value	Description
\$junos-cos-scheduler-map	101	Scheduler-map name configured in a traffic-control profile in a dynamic profile.
\$junos-cos-shaping-rate	102	Shaping rate configured in a traffic-control profile in a dynamic profile. Represents the maximum bandwidth of a CoS scheduler node.
\$junos-cos-guaranteed-rate	103	Guaranteed rate configured in a traffic-control profile in a dynamic profile. Represents the minimum bandwidth of a CoS scheduler node.
\$junos-cos-delay-buffer-rate	104	Delay-buffer rate configured in a traffic-control profile in a dynamic profile.
\$junos-cos-excess-rate	105	Excess rate configured in a traffic-control profile in a dynamic profile; scheduler weighting when operating in the excess region between the guaranteed rate and the shaping rate.  <b>NOTE:</b> Do not configure the \$junos-cos-excess-rate variable when either the \$junos-cos-excess-rate-high variable or the \$junos-cos-excess-rate-low variable is configured.
\$junos-cos-traffic-control-profile	106	Traffic-control profile configured in a dynamic profile for subscriber access.
\$junos-cos-shaping-mode	107	Overhead-accounting mode configured in a traffic-control profile in a dynamic profile to shape downstream ATM traffic based on either frames (frame-mode) or cells (cell-mode).

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets *(Continued)*

Predefined Variable	RADIUS Tag Value	Description
\$junos-cos-byte-adjust	108	Byte adjustment value for the cell or frame shaping mode configured in a traffic-control profile in a dynamic profile.  <b>NOTE:</b> Do not configure the \$junos-cos-byte-adjust variable when either the \$junos-cos-byte-adjust-frame variable or the \$junos-cos-byte-adjust-cell variable is configured.
\$junos-cos-adjust-minimum	109	Minimum adjusted shaping rate configured in a traffic-control profile for a dynamic subscriber interface. Specifying this variable in a traffic-control profile for a dynamic interface set has no effect.
\$junos-cos-excess-rate-high	110	Shaping rate configured for excess high-priority traffic in a traffic-control profile in a dynamic profile.  <b>NOTE:</b> Do not configure the \$junos-cos-excess-rate-high variable when the \$junos-cos-excess-rate variable is configured.
\$junos-cos-excess-rate-low	111	Shaping rate configured for excess low-priority traffic in a traffic-control profile in a dynamic profile.  <b>NOTE:</b> Do not configure the \$junos-cos-excess-rate-low variable when the \$junos-cos-excess-rate variable is configured.
\$junos-cos-shaping-rate-burst	112	Burst size for the shaping rate configured in a traffic-control profile in a dynamic profile.

Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets *(Continued)*

Predefined Variable	RADIUS Tag Value	Description
\$junos-cos-guaranteed-rate-burst	113	Burst size for the guaranteed rate configured in a traffic-control profile in a dynamic profile.
\$junos-cos-byte-adjust-frame	114	Overhead bytes when downstream ATM traffic is in frame-mode.  <b>NOTE:</b> Do not configure the \$junos-cos-byte-adjust-frame variable when the \$junos-cos-byte-adjust variable is configured.
\$junos-cos-byte-adjust-cell	115	Overhead bytes when downstream ATM traffic is in cell-mode.  <b>NOTE:</b> Do not configure the \$junos-cos-byte-adjust-cell variable when the \$junos-cos-byte-adjust variable is configured.
\$junos-cos-shaping-rate-priority-high	116	Shaping rate configured for high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
\$junos-cos-shaping-rate-priority-high-burst	117	Shaping rate burst size configured for high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.



**Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets (Continued)**

Predefined Variable	RADIUS Tag Value	Description
\$junos-cos-shaping-rate-priority-medium	118	Shaping rate configured for medium-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
\$junos-cos-shaping-rate-priority-medium-burst	119	Shaping rate burst size configured for medium-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
\$junos-cos-shaping-rate-priority-low	120	Shaping rate configured for low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
\$junos-cos-shaping-rate-priority-low-burst	121	Shaping rate burst size configured for low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.

**Table 21: Junos OS CoS Traffic Shaping Predefined Variables for Dynamic Interface Sets (Continued)**

Predefined Variable	RADIUS Tag Value	Description
\$junos-cos-shaping-rate-excess-high	122	Shaping rate configured for excess high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
\$junos-cos-shaping-rate-excess-high-burst	123	Shaping rate burst size configured for excess high-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
\$junos-cos-shaping-rate-excess-low	124	Shaping rate configured for excess low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.
\$junos-cos-shaping-rate-excess-low-burst	125	Shaping rate burst size configured for excess low-priority traffic in a traffic-control profile for a dynamic interface set or dynamic ACI interface set at a household level. Specifying this variable in a traffic-control profile for a dynamic subscriber interface is prohibited.

**RELATED DOCUMENTATION**

*Applying CoS Traffic-Shaping Attributes to Dynamic Interface Sets and Member Subscriber Sessions*  
*CoS Traffic Shaping Attributes for Dynamic Interface Sets and Member Subscriber Sessions Overview*

## Example: Configuring Dynamic Hierarchical Scheduling for Subscribers

In this example, subscribers are provided with a data and voice service defined in an access profile when they initially log in. The RADIUS administrator supplies the initial values on the RADIUS server, and the service activation is performed at subscriber login.

After the initial login, the subscriber adds an assured forwarding service that is not defined in the original access profile. A service profile is used to configure the schedulers and a RADIUS CoA activates the service. The queues defined for the schedulers in the initial scheduler map and the new scheduler map are merged.

In addition, the values for the initial data and voice service are upgraded by the RADIUS administrator through a separate RADIUS CoA message.

To configure the initial service and enable the activation through a RADIUS CoA:

1. Configure the access profile for the service activation.
  - a. Configure the VLAN interface for the access profile.

```
[edit]
dynamic-profiles access-profile {
  interfaces {
    $junos-interface-ifd-name {
      unit $junos-underlying-interface-unit {
        family inet;
      }
    }
  }
}
```

- b. Configure the class of service parameters in the access profile. In this example, you configure Junos OS predefined variables that provide the initial scheduler name and scheduler parameters obtained from the RADIUS authentication server when the subscriber logs in.

Include the configurations for the interfaces, schedulers, and the scheduler maps.

```
[edit]
dynamic-profiles access-profile {
  class-of-service {
    traffic-control-profiles {
      tcp1 {
        scheduler-map $junos-cos-scheduler-map;
        shaping-rate $junos-cos-shaping-rate;
        guaranteed-rate $junos-cos-guaranteed-rate;
        delay-buffer-rate $junos-cos-delay-buffer-rate;
      }
    }
  }
  interfaces {
    $junos-interface-ifd-name {
      unit "$junos-underlying-interface-unit" {
        classifiers {
          ieee-802.1 l2_classifier;
        }
        rewrite-rules {
          ieee-802.1 l2_rewrite;
        }
        output-traffic-control-profile tcp1;
      }
    }
  }
  schedulers {
    $junos-cos-scheduler {
      buffer-size percent $junos-cos-scheduler-bs;
      priority $junos-cos-scheduler-pri;
      transmit-rate percent $junos-cos-scheduler-tx;
      drop-profile-map loss-priority low protocol any $junos-cos-scheduler-low;
      drop-profile-map loss-priority medium-low protocol any $junos-cos-
scheduler-medium-low;
      drop-profile-map loss-priority medium-high protocol any $junos-cos-
scheduler-medium-high;
      drop-profile-map loss-priority high protocol any $junos-cos-scheduler-high;
    }
  }
  scheduler-maps {
    data_voice_smap {
      forwarding-class be scheduler be_sch;
    }
  }
}
```

```

        forwarding-class ef scheduler ef_sch;
    }
}
}
}

```

Table 22 on page 261 lists the initial values defined by the RADIUS administrator for the scheduler map and shaping rates.

**Table 22: Initial Scheduler Map and Shaping Values at Subscriber Login**

Predefined Variable	RADIUS Tag	Value
\$junos-cos-scheduler-map	T01	data_voice_smap
\$junos-cos-shaping-rate	T02	6m
\$junos-cos-guaranteed-rate	T03	4m
\$junos-cos-delay-buffer-rate	T04	4m

Table 23 on page 261 lists the initial values defined by the RADIUS administrator for the voice (expedited forwarding) scheduler.

**Table 23: Initial CoS Values for the Voice Scheduler at Subscriber Login**

Predefined Variable	Tag	Value
\$junos-cos-scheduler	—	ef_sch
\$junos-cos-scheduler-tx	T01	10
\$junos-cos-scheduler-bs	T02	10
\$junos-cos-scheduler-pri	T03	medium-high
\$junos-cos-scheduler-dropfile-low	T04	d3

**Table 23: Initial CoS Values for the Voice Scheduler at Subscriber Login (Continued)**

Predefined Variable	Tag	Value
\$junos-cos-scheduler-dropfile-medium-low	T05	d2
\$junos-cos-scheduler-dropfile-medium-high	T06	d1
\$junos-cos-scheduler-dropfile-high	T07	d0

[Table 24 on page 262](#) lists the initial values defined by the RADIUS administrator for the data (best effort) scheduler.

**Table 24: Initial CoS Values for the Data Scheduler at Subscriber Login**

Predefined Variable	Tag	Value
\$junos-cos-scheduler	—	be_sch
\$junos-cos-scheduler-tx	T01	10
\$junos-cos-scheduler-bs	T02	10
\$junos-cos-scheduler-pri	T03	low
\$junos-cos-scheduler-dropfile-low	T04	d0
\$junos-cos-scheduler-dropfile-medium-low	T05	d1
\$junos-cos-scheduler-dropfile-medium-high	T06	d2
\$junos-cos-scheduler-dropfile-high	T07	d3

2. Configure the classifiers, drop profiles, forwarding classes, and rewrite rules in the static [edit class-of-service] hierarchy.

```
[edit]
class-of-service {
  classifiers {
    dscp dscp_classifier {
      forwarding-class be {
        loss-priority low code-points 000000;
      }
      forwarding-class af {
        loss-priority medium-low code-points 000001;
      }
    }
    ieee-802.1 l2_classifier {
      forwarding-class be {
        loss-priority medium-low code-points 000;
      }
      forwarding-class ef {
        loss-priority medium-low code-points 100;
      }
      forwarding-class af {
        loss-priority medium-low code-points 010;
      }
    }
  }
  drop-profiles {
    d0 {
      fill-level 25 drop-probability 100;
      fill-level 0 drop-probability 0;
    }
    d1 {
      fill-level 50 drop-probability 100;
      fill-level 0 drop-probability 0;
    }
    d2 {
      fill-level 75 drop-probability 100;
      fill-level 0 drop-probability 0;
    }
    d3 {
      fill-level 0 drop-probability 0;
      fill-level 100 drop-probability 100;
    }
  }
}
```

```

    }
}
forwarding-classes {
    queue 0 be;
    queue 1 ef;
    queue 2 af;
    queue 3 nc;
}
interfaces {
    ge-1/2/9 {
        shaping-rate 100m;
    }
}
rewrite-rules {
    ieee-802.1 l2_rewrite {
        forwarding-class be {
            loss-priority medium-low code-point 000;
        }
        forwarding-class ef {
            loss-priority medium-low code-point 001;
        }
        forwarding-class af {
            loss-priority medium-low code-point 100;
        }
    }
    dscp l2_rewrite {
        forwarding-class be {
            loss-priority medium-low code-points 000;
        }
        forwarding-class ef {
            loss-priority medium-low code-points 001;
        }
        forwarding-class af {
            loss-priority medium-low code-points 001;
        }
    }
}
}

```

3. Configure the service profile enable RADIUS to activate the video service after login. The video service corresponds to assured forwarding PHB.



In this example, you configure Junos OS predefined variables that provide the initial scheduler name and scheduler parameters obtained from the RADIUS authentication server when the subscriber logs in.

```
[edit]
dynamic-profiles service-af {
  variables {
    af_fc default-value video;
    af_sch default-value af_sch;
    sch-drop-any default-value all;
    sch-pri-2 default-value strict-high;
    sch-bs-2 default-value 40;
    sch-tx-2 default-value 3m;
    smap default-value any
  }
  class-of-service {
    scheduler-maps {
      "$smap" {
        forwarding-class "$af_fc" scheduler "$af_sch";
      }
    }
    schedulers {
      "$af_sch" {
        transmit-rate percent "$sch-tx-2";
        buffer-size percent "$sch-bs-2";
        priority "$sch-pri-2";
        drop-profile-map loss-priority any protocol any drop-profile "$sch-drop-
any";
      }
    }
  }
}
```

After the three services are activated, subscribers receive upgraded values for the data and voice service when RADIUS sends a change of authorization (CoA). In this case, the CoS parameters are replaced, because multiple subscribers were not enabled on the logical interface.

[Table 25 on page 266](#) lists the upgraded values defined by the RADIUS administrator.

**Table 25: Upgraded CoS Values for the Video Service**

Variable	RADIUS Tag	Value
junos-cos-scheduler-map	T01	data_voice_smap
junos-cos-shaping-rate	T02	14m
junos-cos-guaranteed-rate	T03	13m
junos-cos-delay-buffer-rate	T04	12m

[Table 26 on page 266](#) lists the values defined by the RADIUS administrator for the video (assured forwarding) scheduler.

**Table 26: Upgraded CoS Values for the Video Scheduler**

Predefined Variable	Tag	Value
\$junos-cos-scheduler	—	af_sch
\$junos-cos-scheduler-tx	T01	10
\$junos-cos-scheduler-bs	T02	10
\$junos-cos-scheduler-pri	T03	medium
\$junos-cos-scheduler-dropfile-low	T04	d3
\$junos-cos-scheduler-dropfile-medium-low	T05	d2
\$junos-cos-scheduler-dropfile-medium-high	T06	d1
\$junos-cos-scheduler-dropfile-high	T07	d0

[Table 27 on page 267](#) lists the values defined by the RADIUS administrator for the expedited forwarding scheduler in the CoA message. The values are the same as the initial service.

**Table 27: Initial CoS Values for the Expedited Forwarding Scheduler at Subscriber Login**

Predefined Variable	Tag	Value
\$junos-cos-scheduler	–	ef_sch
\$junos-cos-scheduler-tx	T01	10
\$junos-cos-scheduler-bs	T02	10
\$junos-cos-scheduler-pri	T03	medium-high
\$junos-cos-scheduler-dropfile-low	T04	d3
\$junos-cos-scheduler-dropfile-medium-low	T05	d2
\$junos-cos-scheduler-dropfile-medium-high	T06	d1
\$junos-cos-scheduler-dropfile-high	T07	d0

Table 28 on page 267 lists the values defined by the RADIUS administrator for the best effort scheduler in the CoA message. The values are the same as the initial service.

**Table 28: Initial CoS Values for the Best Effort Scheduler at Subscriber Login**

Predefined Variable	Tag	Value
\$junos-cos-scheduler	–	be_sch
\$junos-cos-scheduler-tx	T01	10
\$junos-cos-scheduler-bs	T02	10
\$junos-cos-scheduler-pri	T03	low

**Table 28: Initial CoS Values for the Best Effort Scheduler at Subscriber Login (Continued)**

Predefined Variable	Tag	Value
\$junos-cos-scheduler-dropfile-low	T04	d0
\$junos-cos-scheduler-dropfile-medium-low	T05	d1
\$junos-cos-scheduler-dropfile-medium-high	T06	d2
\$junos-cos-scheduler-dropfile-high	T07	d3

**RELATED DOCUMENTATION**

[Changing CoS Services Overview | 240](#)

*Guidelines for Configuring Dynamic CoS for Subscriber Access*

[Understanding Hierarchical CoS for Subscriber Interfaces | 90](#)

# 3

PART

## Configuration Statements and Operational Commands

---

[Junos CLI Reference Overview | 270](#)

---

# Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- [Junos CLI Reference](#)

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- [Configuration Statements](#)
- [Operational Commands](#)