Junos® OS

CoS Hierarchical Queue Scheduling

Release
13.1

Published: 2013-02-11

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS CoS Hierarchical Queue Scheduling
13.1

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

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- T Series
- M Series
- MX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```


2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   [edit]
   user@host# **edit system scripts**
   [edit system scripts]

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   [edit system scripts]
   user@host# **load merge relative /var/tmp/ex-script-snippet.conf**
   load complete

For more information about the **load** command, see the CLI User Guide.

Documentation Conventions

Table 1 on page xi defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xi defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the**configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Italic text like this	• Introduces or emphasizes important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		    route default {
		      nexthop *address*;
		      retain;
		    }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/

- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Hierarchical Schedulers

Hierarchical Schedulers Terminology

Hierarchical schedulers introduce some new terms into a discussion of CoS capabilities. They also use some familiar terms in different contexts. This section presents a complete overview of the terms used with hierarchical schedulers.

The following terms are important for hierarchical schedulers:

- 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.

These terms are especially important when applied to a scheduler hierarchy. Scheduler hierarchies are composed of nodes and queues. Queues terminate the CLI 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. For example, if an **interface-set** statement is configured with a logical interface (such as **unit 0**) and queue, then the **interface-set** is an internal node at Level 2 of the hierarchy. However, if there are no traffic control profiles configured on logical interfaces, then the interface set is at Level 3 of the hierarchy.

Table 3 on page 4 shows how the configuration of an interface set or logical interface affects the terminology of hierarchical scheduler nodes.

Table 3: Hierarchical Scheduler Nodes

Root Node (Level 1)	Level 2	Level 3	Queue (Level 4)
Physical interface	Interface set	Logical interfaces	One or more queues
Physical interface		Interface set	One or more queues
Physical interface		Logical interfaces	One or more queues

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).

Interface Set Caveats

When configuring interface sets, consider the following guidelines:

- Interface sets can be defined in two major ways: as a list of logical interfaces or groups of aggregated Ethernet logical interfaces (**unit 100**, **unit 200**, and so on), or at the stacked VLAN level using a list of outer VLAN IDs (**vlan-tags-outer 210**, **vlan-tags-outer 220**, and so on). You can configure sets of aggregated Ethernet interfaces on MIC or MPC interfaces only.
- You cannot specify an interface set mixing the logical interface, aggregated Ethernet, S-VLAN, or VLAN outer tag list forms of the **interface-set** statement.
- Keep the following guidelines in mind when configuring interface sets of logical interfaces over aggregated Ethernet:
  - Sets of aggregated Ethernet interfaces are supported on MIC and MPC interfaces 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.
- A logical interface can only belong to one interface set. If you try to add the same logical interface to different interface sets, the commit operation fails.

  This example generates a commit error:

```
[edit interfaces]
interface-set set-one {
  interface ge-2/0/0 {
    unit 0;
    unit 2;
  }
}
interface-set set-two {
  interface ge-2/0/0 {
    unit 1;
    unit 3;
    unit 0; # COMMIT ERROR! Unit 0 already belongs to set-one.
  }
}
```


- Members of an interface set cannot span multiple physical interfaces. Only one physical interface is allowed to appear in an interface set.

  This configuration is not supported:

```
[edit interfaces]
interface-set set-group {
  interface ge-0/0/1 {
    unit 0;
    unit 1;
  }
  interface ge-0/0/2 { # This is NOT supported in the same interface set!
    unit 0;
    unit 1;
  }
}
```

Related Documentation

- Configuring Interface Sets on page 17

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.

Hierarchical schedulers add CoS parameters to the new 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

For more information on traffic control profiles see Oversubscribing Interface Bandwidth and Providing a Guaranteed Minimum Rate. For more information on scheduler maps, see Configuring Scheduler Maps.

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.

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 on page 16
- **output-traffic-control-profile-remaining on page 35**
- Controlling Remaining Traffic on page 19

PIR-Only and CIR Mode

The actual behavior of many CoS parameters, especially the shaping rate and guaranteed rate, depend on whether the physical interface is operating in PIR-only or CIR mode.

In PIR-only mode, one or more nodes perform shaping. The physical interface is in the PIR-only mode if no child (or grandchild) node under the port has a guaranteed rate configured.

The mode of the port is important because in PIR-only mode, the scheduling across the child nodes is in proportion to their shaping rates (PIRs) and not the guaranteed rates (CIRs). This can be important if the observed behavior is not what is anticipated.

In CIR mode, one or more nodes applies a guaranteed rate and might perform shaping. A physical interface is in CIR mode if at least one child (or grandchild) node has a guaranteed rate configured.

In CIR mode, one or more nodes applies the guaranteed rates. In addition, any child or grandchild node under the physical interface can have a shaping rate configured. Only the guaranteed rate matters. In CIR mode, nodes that do not have a guaranteed rate configured are assumed to have a very small guaranteed rate (queuing weight).

Priority Propagation

Priority propagation is performed for MX Series router output Interfaces on Enhanced Queuing DPCs, MICs, and MPCs, and for M Series and T Series router output interfaces on IQ2E PICs. 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 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 (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 4 on page 9.

Table 4: Queue Priority

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

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 5 on page 9.

Table 5: 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

Table 5: 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
Medium-high	1	3
Medium-low	1	3
Low	2	3

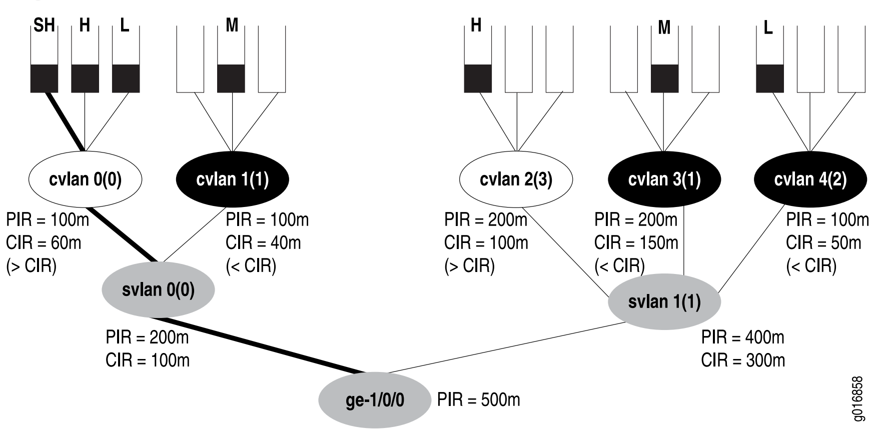
In PIR-only mode, nodes cannot send if they are above the configured shaping rate. The mapping between the configured priority and the hardware priority is for PIR-only mode is shown in Table 6 on page 10.

Table 6: 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 1 on page 10. 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 1: 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 5 on page 9 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 cvlan2 (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 Hardware Properties
- CoS Features on MIC and MPC Interfaces Overview
- Scheduler Node Scaling on MIC and MPC Interfaces Overview
- priority (Schedulers)
- schedulers (Class of Service)

PART 2

Configuration

CHAPTER 2

Configuration for Hierarchical Scheduling

Configuring Hierarchical Schedulers for CoS

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.

On Juniper Networks MX Series 3D Universal Edge Routers and systems with Enhanced IQ2 (IQ2E) PICs, you can apply CoS shaping and scheduling at one of four different levels, including the VLAN set level. You can only use this configuration on MX Series routers or IQ2E PICs. For more information about configuring CoS on IQ2E PICs, see CoS on Enhanced IQ2 PICs Overview.

The supported scheduler hierarchy is as follows:

- The physical interface (level 1)
- The service VLAN (level 2 is unique to MX Series routers)
- 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 schedulers, include the following statements at the **[edit class-of-service interfaces]** and **[edit interfaces]** hierarchy levels:

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

[edit interfaces]
hierarchical-scheduler;
interface-set interface-set-name {
  interface ethernet-interface-name {
    (unit unit-number | vlan-tags-outer vlan-tag);
  }
}
```

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

5. Configure scheduler mode.

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

Related Documentation

- Hierarchical Schedulers on Aggregated Ethernet Interfaces Overview on page 7

Configuring Interface Sets

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

```
[edit class-of-service interfaces]
interface-set interface-set-name {
  ...interface-cos-configuration-statements ...
}
```

To apply the interface set to interfaces, include the **interface-set** statement at the **[edit interfaces]** hierarchy level:

```
[edit interfaces]
interface-set interface-set-name {
  interface ethernet-interface-name {
    ... interface-cos-configuration-statements ...
  }
}
```

Interface sets can be defined in two major ways:

- As a list of logical interfaces or aggregated Ethernet interfaces (**unit 100**, **unit 200**, and so on)
- At the stacked VLAN level using a list of outer VLAN IDs (**vlan-tags-outer 210**, **vlan-tags-outer 220**, and so on).

  The **svlan *number*** listing option with a single outer VLAN tag is a convenient way to specify a set of VLAN members having the same outer VLAN tags. Service providers can use these statements to group interfaces to apply scheduling parameters such as guaranteed rate and shaping rate to the traffic in the groups.

Whether using the logical interface listing option for a group of customer VLANs, aggregated Ethernet interfaces, or the S-VLAN set listing option for a group of VLAN outer tags, all traffic heading downstream must be gathered into an interface set with the **interface-set** statement at the **[edit class-of-service interfaces]** hierarchy level.

Regardless of listing convention, you can only use one of the types in an interface set. Examples of this limitation appear later in this section.

NOTE: Interface sets are currently used only by CoS, but they are applied at the **[edit interfaces]** hierarchy level to make them available to other services that might use them in future.

```
[edit interfaces]
interface-set interface-set-name {
  interface ethernet-interface-name {
    (unit logical-unit-number | vlan-tags-outer vlan-tag) {
      ...
    }
  }
}
```

The logical interface naming option lists Ethernet interfaces:

```
[edit interfaces]
interface-set unitl-set-ge-0 {
  interface ge-0/0/0 {
    unit 0;
    unit 1;
    ...
  }
}
```

The interface naming option lists aggregated Ethernet interfaces:

```
[edit interfaces]
  interface-set demuxset1 {
    interface demux0 {
      unit 1;
      ..
    }
  }
  demux0 {
    unit 1 {
      demux-options {
        underlying-interface ae0.1;
      }
      family inet {
        demux-source {
          100.1.1.1/24;
        }
        address 100.1.1.1/24;
      }
    }
    ..
    ae0 {
      unit 1 {
      }
      ..
    }
  }
  class-of-service {
    interface-set demuxset1 {
      output-traffic-control-profile tcp2;
```



```
        }
    }
}
```

The S-VLAN option lists only one S-VLAN (outer) tag value:

```
[edit interfaces]
interface-set svlan-set {
    interface ge-1/0/0 {
        vlan-tags-outer 2000;
    }
}
```

The S-VLAN naming option lists S-VLAN (outer) tag values:

```
[edit interfaces]
interface-set svlan-set-tags {
    interface ge-2/0/0 {
        vlan-tags-outer 2000;
        vlan-tags-outer 2001;
        vlan-tags-outer 2002;
        ...
    }
}
```

NOTE: Ranges are not supported: you must list each VLAN or logical interface separately.

Related Documentation

- Interface Set Caveats on page 5

Applying Interface Sets

Although the interface set is applied at the **[edit interfaces]** hierarchy level, the CoS parameters for the interface set are defined at the **[edit class-of-service interfaces]** hierarchy level, usually with the **output-traffic-control-profile** ***profile-name*** statement.

This example applies a traffic control profile called **tcp-set1** to an interface set called **set-ge-0**:

```
[edit class-of-service interfaces]
interface-set set-ge-0 {
    output-traffic-control-profile tcp-set1;
}
```

Related Documentation

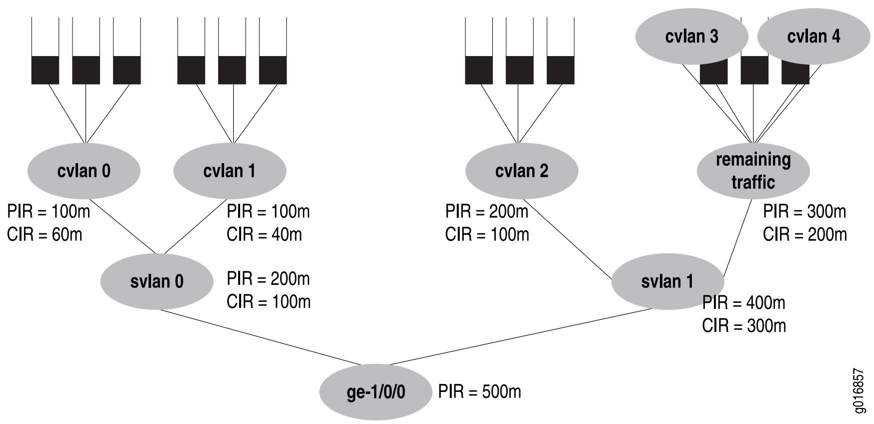
- output-traffic-control-profile on page 34

Controlling Remaining Traffic

You can configure many logical interfaces under an interface. However, only a subset of them might have a traffic control profile attached. For example, you can configure three logical interfaces (units) over the same service VLAN, but apply a traffic control profile specifying best-effort and voice queues to only one of the logical interface units. Traffic from the two remaining logical interfaces is considered *remaining traffic*. To configure transmit rate guarantees for the remaining traffic, you configure the **output-traffic-control-profile-remaining** statement specifying a guaranteed rate for the remaining traffic. Without this statement, the remaining traffic gets a default, minimal bandwidth. In the same way, the **shaping-rate** and **delay-buffer-rate** statements can be specified in the traffic control profile referenced with the **output-traffic-control-profile-remaining** statement in order to shape and provide buffering for remaining traffic.

Consider the interface shown in Figure 2 on page 20. Customer VLANs 3 and 4 have no explicit traffic control profile. However, the service provider might want to establish a shaping and guaranteed transmit rate for aggregate traffic heading for those customer VLANs. The solution in to configure and apply a traffic control profile for all remaining traffic on the interface.

Figure 2: Handling Remaining Traffic



This example considers the case where customer VLANs 3 and 4 have no explicit traffic control profile, yet need to establish a shaping and guaranteed transmit rate for traffic heading for those customer VLANs. The solution is to add a traffic control profile to the **svlan1** interface set. This example builds on the earlier example and so does not repeat all configuration details, only those at the service VLAN level.

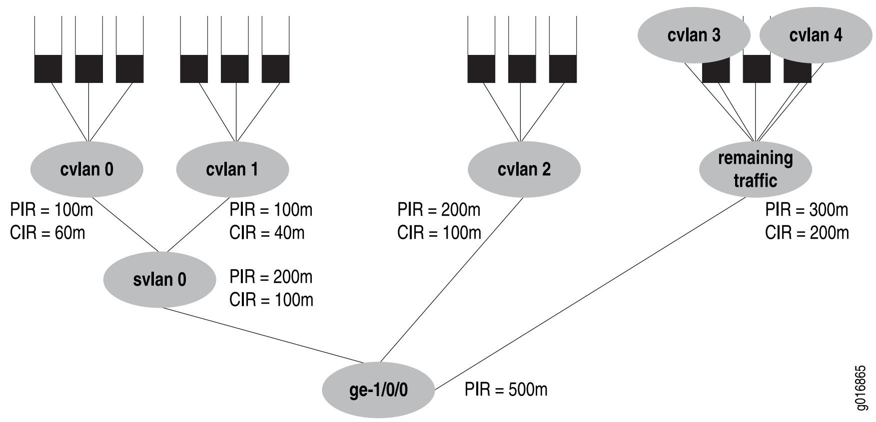
```
[edit class-of-service interfaces]
interface-set svlan0 {
   output-traffic-control-profile tcp-svlan0;
}
interface-set svlan1 {
   output-traffic-control-profile tcp-svlan1; # For explicitly shaped traffic.
   output-traffic-control-profile-remaining tcp-svlan1-remaining; # For all remaining traffic.
}

[edit class-of-service traffic-control-profiles]
tcp-svlan1 {
   shaping-rate 400m;
   guaranteed-rate 300m;
}
tcp-svlan1-remaining {
```



```
    shaping-rate 300m;
    guaranteed-rate 200m;
    scheduler-map smap-remainder; # this smap is not shown in detail
}
```

Next, consider the example shown in Figure 3 on page 21.

Figure 3: Another Example of Handling Remaining Traffic



In this example, **ge-1/0/0** has three logical interfaces (unit 1, unit 2, and unit 3), and SVLAN 2000, which are covered by the interface set:

- Scheduling for the interface set **svlan0** is specified by referencing an **output-traffic-control-profile** statement which specifies the **guaranteed-rate**, **shaping-rate**, and **delay-buffer-rate** statement values for the interface set. In this example, the output traffic control profile called **tcp-svlan0** guarantees 100 Mbps and shapes the interface set **svlan0** to 200 Mbps.
- Scheduling and queuing for remaining traffic of **svlan0** is specified by referencing an **output-traffic-control-profile-remaining** statement which references a **scheduler-map** statement that establishes queues for the remaining traffic. The specified traffic control profile can also configure guaranteed, shaping, and delay-buffer rates for the remaining traffic. In this example, **output-traffic-control-profile-remaining tcp-svlan0-rem** references **scheduler-map smap-svlan0-rem**, which calls for a best-effort queue for remaining traffic (that is, traffic on unit 3 and unit 4, which is not classified by the **svlan0** interface set). The example also specifies a **guaranteed-rate** of 200 Mbps and a **shaping-rate** of 300 Mbps for all remaining traffic.
- Scheduling and queuing for logical interface **ge-1/0/0 unit 1** is configured "traditionally" and uses an **output-traffic-control-profile** specified for that unit. In this example, **output-traffic-control-profile tcp-ifl1** specifies scheduling and queuing for **ge-1/0/0 unit 1**.

This example does not include the **[edit interfaces]** configuration.

```
[edit class-of-service interfaces]
interface-set {
  svlan0 {
    output-traffic-control-profile tcp-svlan0; # Guarantee & shaper for svlan0.
  }
}
```



```
ge-1/0/0 {
  output-traffic-control-profile-remaining tcp-svlan0-rem;
  # Unit 3 and 4 are not explicitly configured, but captured by "remaining"
  unit 1 {
    output-traffic-control-profile tcp-ifl1; # Unit 1 be & ef queues.
  }
}
```

Here is how the traffic control profiles for this example are configured:

```
[edit class-of-service traffic-control-profiles]
tcp-svlan0 {
  shaping-rate 200m;
  guaranteed-rate 100m;
}
tcp-svlan0-rem {
  shaping-rate 300m;
  guaranteed-rate 200m;
  scheduler-map smap-svlan0-rem; # This specifies queues for remaining traffic
}
tcp-ifl1 {
  scheduler-map smap-ifl1;
}
```

Finally, here are the scheduler maps and queues for the example:

```
[edit class-of-service scheduler-maps]
smap-svlan0-rem {
  forwarding-class best-effort scheduler sched-foo;
}
smap-ifl1 {
  forwarding-class best-effort scheduler sched-bar;
  forwarding-class assured-forwarding scheduler sched-baz;
}
```

The configuration for the referenced schedulers are not given for this example.

Configuring Internal Scheduler Nodes

A node in the hierarchy is considered internal if either of the following conditions apply:

- Any one of its children nodes has a traffic control profile configured and applied.
- You include the **internal-node** statement at the **[edit class-of-service interfaces interface-set *set-name*]** hierarchy level.

Why would it be important to make a certain node internal? Generally, there are more resources available at the logical interface (unit) level than at the interface set level. Also, it might be desirable to configure all resources at a single level, rather than spread over several levels. The **internal-node** statement provides this flexibility. This can be a helpful configuration device when interface-set queuing without logical interfaces is used exclusively on the interface.

The **internal-node** statement can be used to raise the interface set without children to the same level as the other configured interface sets with children, allowing them to compete for the same set of resources.

In summary, using the **internal-node** statement allows statements to all be scheduled at the same level with or without children.

The following example makes the interfaces sets **if-set-1** and **if-set-2** internal:

```
[edit class-of-service interfaces]
interface-set {
  if-set-1 {
    internal-node;
    output-traffic-control-profile tcp-200m-no-smap;
  }
  if-set-2 {
    internal-node;
    output-traffic-control-profile tcp-100m-no-smap;
  }
}
```

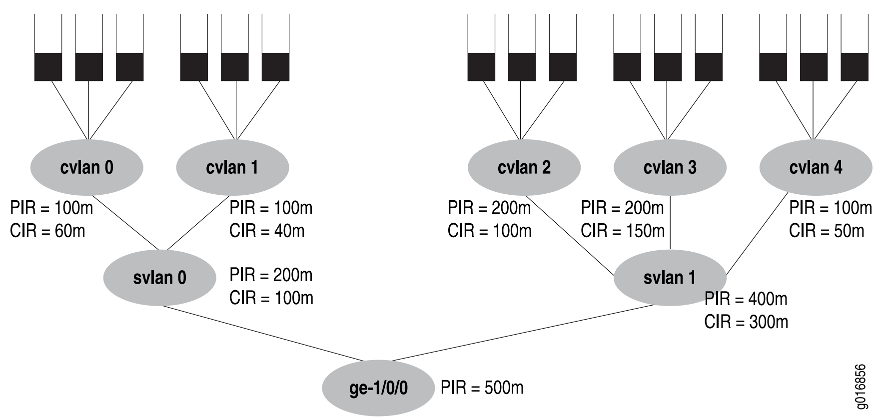
If an interface set has logical interfaces configured with a traffic control profile, then the use of the **internal-node** statement has no effect.

Internal nodes can specify a **traffic-control-profile-remaining** statement.

Example: Four-Level Hierarchy of Schedulers

This section provides a more complete example of building a 4-level hierarchy of schedulers. The configuration parameters are shown in Figure 4 on page 23. The queues are shown at the top of the figure with the other three levels of the hierarchy below.

Figure 4: 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 on page 24
- Configuring the Interfaces on page 24
- Configuring the Traffic Control Profiles on page 25
- Configuring the Schedulers on page 25
- Configuring the Drop Profiles on page 26
- Configuring the Scheduler Maps on page 26
- Applying the Traffic Control Profiles on page 27

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 Overview.

```
[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 25.

```
[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) but 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.

CHAPTER 3

Configuration Statements

excess-bandwidth-share

Syntax excess-bandwidth-share (proportional *value* | equal);

Hierarchy Level [edit class-of-service interfaces interface-set *interface-set-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Determines the method of sharing excess bandwidth in a hierarchical scheduler environment. If you do not include this statement, the node shares excess bandwidth proportionally at 32.64 Mbps.

Options **proportional** ***value***—(Default) Share excess bandwidth proportionally (default value is 32.64 Mbps).

equal—Share excess bandwidth equally.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring MDRR on Enhanced Queuing DPCs
- Configuring Hierarchical Schedulers for CoS on page 15
- Configuring Interface Sets on page 17

hierarchical-scheduler

Syntax hierarchical-scheduler;

Hierarchy Level [edit class-of-service interfaces]

Release Information Statement introduced in Junos OS Release 8.5.

Description On MX Series, M Series, and T Series routers with IQ2E PIC, enables the use of hierarchical schedulers.

NOTE: To enable hierarchical scheduling on MX80 routers, configure the **hierarchical-scheduler** statement at each member physical interface level of a particular aggregated Ethernet interface as well as at that aggregated Ethernet interface level. On other routing platforms, it is enough if you include this statement at the aggregated Ethernet interface level.

Default If you do not include this statement, the interfaces on the MX Series router cannot use hierarchical interfaces.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 15

interface-set (Ethernet Interfaces)

Syntax

```
interface-set interface-set-name {
  interface ethernet-interface-name {
    (unit unit-number | vlan-tags-outer vlan-tag);
  }
}
```

Hierarchy Level [edit interfaces]

Release Information Statement introduced in Junos OS Release 8.5.

Description The set of interfaces used to configure hierarchical CoS schedulers on Ethernet interfaces on the MX Series router and IQ2E PIC on M Series and T Series routers.

The remaining statements are described separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring E-LINE and E-LAN Services for a PBB Network on MX Series Routers
- Junos OS Class of Service Configuration Guide

interface-set (Hierarchical Schedulers)

Syntax

```
interface-set interface-set-name {
  excess-bandwidth-share (proportional value | equal);
  internal-node;
  output-traffic-control-profile profile-name;
  output-traffic-control-profile-remaining profile-name;
}
```

Hierarchy Level [edit class-of-service interfaces]

Release Information Statement introduced in Junos OS Release 8.5.

Description For Enhanced Queuing DPC, MIC, or MPC interfaces on MX Series routers, or for IQ2E PIC interfaces on M Series and T Series routers, configure hierarchical schedulers for an interface set.

Options *interface-set-name*—Name of the interface set.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Interface Sets on page 17
- Configuring Hierarchical Schedulers for CoS on page 15

internal-node

Syntax internal-node;

Hierarchy Level [edit class-of-service interfaces interface-set *interface-set-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description The statement is used to raise the interface set without children to the same level as the other configured interface sets with children, allowing them to compete for the same set of resources.

Default If you do not include this statement, the node is internal only if its children have a traffic control profile configured.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Internal Scheduler Nodes on page 22

member-link-scheduler

Syntax member-link-scheduler (replicate | scale);

Hierarchy Level [edit class-of-service interfaces],
[edit logical-systems *logical-system-name* class-of-service interfaces *interface-name*],
[edit routing-instances *routing-instance-name* class-of-service interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6.

Description Determines whether scheduler parameters for aggregated interface member links are applied in a replicated or scaled manner.

Default By default, scheduler parameters are scaled (in “equal division mode”) among aggregated interface member links.

Options **replicate**—Scheduler parameters are copied to each level of the aggregated interface member links.

scale—Scheduler parameters are scaled based on number of member links and applied each level of the aggregated interface member links.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 15

output-traffic-control-profile

Syntax output-traffic-control-profile *profile-name* shared-instance *instance-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name* unit *logical-unit-number*],
[edit class-of-service interfaces *interface-name* interface-set *interface-set-name*]

Release Information Statement introduced in Junos OS Release 7.6.
interface-set option added for Enhanced Queuing DPCs on MX Series routers in Junos OS Release 8.5.
interface-set option added for MIC and MPC interfaces on MX Series routers in Junos OS Release 10.2.

Description For Channelized IQ PIC interfaces, for Gigabit Ethernet IQ, Gigabit Ethernet IQ2, and IQ2E PIC interfaces, for link services IQ (LSQ) interfaces on AS PICs, and for Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers, apply an output traffic scheduling and shaping profile to the logical interface.

The **shared-instance** statement is supported on Gigabit Ethernet IQ2 PICs only.

Options ***profile-name***—Name of the traffic-control profile to be applied to this interface

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Oversubscribing Interface Bandwidth
- Configuring Traffic Control Profiles for Shared Scheduling and Shaping
- Example: Configuring CoS for a PBB Network on MX Series Routers
- Configuring Hierarchical Schedulers for CoS on page 15 (Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers)
- Configuring Interface Sets on page 17 (Enhanced Queuing DPC, MIC, and MPC interfaces on MX Series routers)
- **output-traffic-control-profile-remaining on page 35**
- traffic-control-profiles

output-traffic-control-profile-remaining

Syntax output-traffic-control-profile-remaining *profile-name*;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* interface-set *interface-set-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description For Enhanced Queuing DPC, MICs, and MPC interfaces on MX Series routers, and for IQ2E PIC interfaces on M Series and T Series routers, apply an output traffic scheduling and shaping profile for remaining traffic to the logical interface or interface set. The remaining traffic is transmitted by the default interface or interface set.

You can map the TCP to the interface or interface set by using the **output-traffic-control-profile-remaining** statement to explicitly configure the queues of the default interface or interface set scheduler that transmits the remaining traffic.

Options *profile-name*—Name of the traffic-control profile for remaining traffic to be applied to this interface or interface set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Hierarchical Schedulers for CoS on page 15
- Configuring Remaining Common Queues on MIC and MPC Interfaces
- output-traffic-control-profile on page 34

PART 3

Index

Index

Symbols

B

C

D

E

F

H

I

M

O

P