



Port-Shaper and LLQ in the Presence of EFPs

The Port-Shaper and LLQ in the Presence of EFPs feature allows network designers to configure port and class policies on ports that contain Ethernet Flow Points (EFPs). These policies support Low Latency Queuing (LLQ) and traffic prioritization across the EFPs.

- [Restrictions for Port-Shaper and LLQ in the Presence of EFPs, on page 1](#)
- [Information About Port-Shaper and LLQ in the Presence of EFPs, on page 2](#)
- [How to Configure Port-Shaper and LLQ in the Presence of EFPs, on page 2](#)
- [Additional References, on page 9](#)

Restrictions for Port-Shaper and LLQ in the Presence of EFPs

- When a policy-map with shaper or queuing action is attached to 2 different interfaces and unknown unicast or multicast traffic flows from both the directions of the interface, then the policy-map might not work as expected.
- For port-shaper in the presence of EFPs, policy on the port must be applied first followed by policy on EFP. Applying policy on EFP first followed by the port does not work.
- If you configure a class-based policy on the port, then you cannot configure service-policies on Ethernet Flow Points (EFPs).
- Attaching a service policy to the BDI is not supported.
- Usage of bandwidth remaining percentage (BRP) in the absence of priority class, allocates the available bandwidth in an iterative way. For example, the bandwidth is allocated for the first BRP class as per the percentage of share that is configured in the respective class-map and the remaining bandwidth is iteratively allocated to all other BRP classes until the bandwidth is exhausted.
- When the port is down, if you manually change the port speed then it may result in unexpected behavior.
- You must remove bandwidth statement from the child policy before it is removed from the parent policy.

Information About Port-Shaper and LLQ in the Presence of EFPs

Ethernet Flow Points and LLQ

An Ethernet Flow Point (EFP) is a forwarding decision point in the provider edge (PE) router, which gives network designers flexibility to make many Layer 2 flow decisions within the interface. Many EFPs can be configured on a single physical port. (The number varies from one device to another.) EFPs are the logical demarcation points of an Ethernet virtual connection (EVC) on an interface. An EVC that uses two or more User-Network Interfaces (UNIs) requires an EFP on the associated ingress and egress interfaces of every device that the EVC passes through.

The Egress HQoS with Port Level Shaping feature allows network designers to configure port and class policies on ports that contain EFPs. These policies support Low Latency Queueing (LLQ) and traffic prioritization across the EFPs.

For information on how to configure LLQ, see the *QoS Congestion Management Configuration Guide*.

How to Configure Port-Shaper and LLQ in the Presence of EFPs

To configure the Port-Shaper and LLQ in the Presence of EFPs feature, you first create either a hierarchical or flat policy map that supports Low Latency Queueing (LLQ), which you then attach to an EFP interface.

Configuring Hierarchical Policy Maps

To configure hierarchical policy maps, you create child policies which you then attach to a parent policy. The parent policy is then attached to an interface.

Procedure

Step 1

enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2

configure terminal

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3

policy-map *policy-map-name*

Example:

```
Device(config)# policy-map child-llq
```

Creates or modifies the child policy and enters QoS policy-map configuration mode.

- child-llq is the name of the child policy map.

Step 4 **class** *class-map-name***Example:**

```
Device(config-pmap)# class qos-1
```

Assigns the traffic class you specify to the policy map and enters QoS policy-map class configuration mode.

- qos-1 is the name of a previously configured class map and is the traffic class for which you want to define QoS actions.

Step 5 **bandwidth percent** *percent***Example:**

```
Device(config-pmap-c)# bandwidth percent 20
```

(Optional) Specifies a bandwidth percent for class-level queues to be used during congestion to determine the amount of excess bandwidth (unused by priority traffic) to allocate to nonpriority queues.

Step 6 **exit****Example:**

```
Device(config-pmap-c)# exit
```

Exits QoS policy-map class configuration mode.

Step 7 **class** *class-map-name***Example:**

```
Device(config-pmap)# class qos-2
```

Assigns the traffic class you specify to the policy map and enters QoS policy-map class configuration mode.

- qos-2 is the name of a previously configured class map and is the traffic class for which you want to define QoS actions.

Step 8 **bandwidth percent** *percent***Example:**

```
Device(config-pmap-c)# bandwidth percent 80
```

(Optional) Specifies a bandwidth percent for class-level queues to be used during congestion to determine the amount of excess bandwidth (unused by priority traffic) to allocate to nonpriority queues.

Step 9 **exit****Example:**

```
Device(config-pmap-c)# exit
```

Exits QoS policy-map class configuration mode.

Step 10 **policy-map** *policy-map-name*

Example:

```
Device(config-pmap)# policy-map parent-llq
```

Creates or modifies the parent policy.

- parent-llq is the name of the parent policy map.

Step 11 **class** *class-default*

Example:

```
Device(config-pmap)# class class-default
```

Configures or modifies the parent class-default class and enters QoS policy-map class configuration mode.

- You can configure only the class-default class in a parent policy. Do not configure any other traffic class.

Step 12 **service-policy** *policy-map-name*

Example:

```
Device(config-pmap-c)# service-policy child-llq
```

Applies the child policy to the parent class-default class.

- child-llq is the name of the child policy map configured in step 1.

Configuring an LLQ Policy Map

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **policy-map** *policy-map-name*

Example:

```
Device(config)# policy-map llq-flat
```

Creates a policy and enters QoS policy-map configuration mode.

Step 4 **class** *class-map-name*

Example:

```
Device(config-pmap)# class qos-group
```

Assigns the traffic class you specify to the policy map and enters policy-map class configuration mode.

Step 5 **priority**

Example:

```
Device(config-pmap-c)# priority
```

Configures LLQ, providing strict priority queueing (PQ) for class-based weighted fair queueing (CBWFQ).

Step 6 **exit**

Example:

```
Device(config-pmap-c)# exit
```

Exits QoS policy-map class configuration mode.

Step 7 **class** *class-map-name*

Example:

```
Device(config-pmap)# class qos-group
```

Assigns the traffic class you specify to the policy map and enters QoS policy-map class configuration mode.

Step 8 **shape average** *value*

Example:

```
Device(config-pmap-c)# shape average 200000000
```

Configures a shape entity with a Committed Information Rate of 200 Mb/s.

Step 9 **exit**

Example:

```
Device(config-pmap-c)# exit
```

Exits QoS policy-map class configuration mode.

Step 10 **class** *class-map-name*

Example:

```
Device(config-pmap)# class qos-group
```

Assigns the traffic class you specify to the policy map and enters QoS policy-map class configuration mode.

Step 11 **bandwidth** *percent*

Example:

```
Device(config-pmap-c)# bandwidth 4000000
```

(Optional) Specifies a bandwidth percent for class-level queues to be used during congestion to determine the amount of excess bandwidth (unused by priority traffic) to allocate to non-priority queues.

Step 12 **exit**

Example:

```
Device(config-pmap-c)# exit
```

Exits QoS policy-map class configuration mode.

Configuring Port Level Shaping on the Main Interface with Ethernet Flow Points

To configure port level shaping on the main interface with EFPS, first you enable the autonegotiation protocol on the interface, then you attach a policy map to the interface and finally you configure the Ethernet service instance.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface GigabitEthernet 0/0/1
```

Configures an interface type and enters interface configuration mode.

- Enter the interface type number.

Step 4 **no ip address**

Example:

```
Device(config-if)# no ip address
```

Disables IP routing on the interface.

Step 5 **negotiation auto**

Example:

```
Device(config-if)# negotiation auto
```

Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.

Step 6 **service-policy output *policy-map-name***

Example:

```
Device(config-if)# service-policy output parent-llq
```

Specifies the name of the policy map to be attached to the input or output direction of the interface.

- You can enter the name of a hierarchical or a flat policy map.

Step 7 **service instance *id* ethernet**

Example:

```
Device(config-if)# service instance 1 ethernet
```

Configures an Ethernet service instance on an interface and enters service instance configuration mode.

Step 8 **encapsulation dot1q *vlan-id***

Example:

```
Device(config-if-srv)# encapsulation dot1q 100
```

Defines the matching criteria to map 802.1Q frames' ingress on an interface to the service instance.

Step 9 **bridge-domain *bridge-domain-id***

Example:

```
Device(config-if-srv)# bridge-domain 100
```

Binds the bridge domain to the service instance.

Step 10 **exit**

Example:

```
Device(config-if-srv)# exit
```

Exits service instance configuration mode.

Step 11 **service instance** *id ethernet*

Example:

```
Device(config-if)# service instance 2 ethernet
```

Configures an Ethernet service instance on an interface and enters service instance configuration mode.

Step 12 **encapsulation dot1q** *vlan-id*

Example:

```
Device(config-if-srv)# encapsulation dot1q 101
```

Defines the matching criteria to map 802.1Q frames' ingress on an interface to the service instance.

Step 13 **bridge-domain** *bridge-domain-id*

Example:

```
Device(config-if-srv)# bridge-domain 101
```

Binds the bridge domain to the service instance.

Step 14 **exit**

Example:

```
Device(config-if-srv)# exit
```

Exits QoS policy-map class configuration mode.

Step 15 **end**

Example:

```
Device(config-if)# end
```

(Optional) Exits interface configuration mode.

Example: Configuring Hierarchical QoS

The router supports hierarchical QoS policies with up to three levels:

- Port policy—Uses class-default only and should be applied on physical interface as initial configuration.
- EFP policy—Uses a nested policy with parent policy matching on class-default and child policy matching on QoS-group values.

The following example displays the hierarchical QoS configuration using port policy and EFP policy:


```

policy-map parent1
class class-default
shape average 1000000000
!
policy-map EFP30
class class-default
shape average 500000000
service-policy pq_test
!
end

```

Use the **show run policy-map** to verify the configuration:

```

router#sh run policy-map pq_test
Building configuration...

Current configuration : 590 bytes
!
policy-map pq_test
class qq1
priority level 1 percent 20
queue-limit percent 10
class qq0
priority level 2 percent 10
queue-limit percent 10
class qq2
bandwidth remaining percent 15
queue-limit percent 10
class qq3
bandwidth remaining percent 20
queue-limit percent 10
class qq4
bandwidth remaining percent 25
queue-limit percent 10
class qq5
bandwidth remaining percent 10
queue-limit percent 10
class qq6
bandwidth remaining percent 25
queue-limit percent 10
class class-default
bandwidth remaining percent 5
queue-limit percent 10
!
end

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS QoS Command Reference
Policing and shaping	"Policing and Shaping Overview" module
Class maps	"Applying QoS Features Using the MQC" module
Policy maps	"Applying QoS Features Using the MQC" module
Low Latency Queueing	QoS Congestion Management Configuration Guide

Standards and RFCs

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html