

# Shared-Queue QoS Policies

---

## In This Section

This section provides information to configure shared-queue QoS policies using the command line interface.

Topics in this section include:

- [Overview on page 690](#)
- [Basic Configurations on page 697](#)
- [Default Shared Queue Policy Values on page 702](#)

## Overview

Shared-queue QoS policies can be implemented to facilitate queue consumption on the router. It is especially useful when VPLS, IES, and VPRN services are scaled to very high numbers. Instead of allocating multiple hardware queues for each unicast queue defined in a SAP ingress QoS policy, SAPs with the shared-queuing feature enabled only allocate one hardware queue for each SAP ingress QoS policy unicast queue.

However, as a trade-off, the total amount of traffic throughput at the ingress of the node is reduced because any ingress packet serviced by a shared-queuing SAP is recirculated for further processing. This can reduce the bandwidth by half. Shared-queuing can add latency. Network planners should consider these restrictions while trying to scale services on the router.

---

## Multipoint Shared Queuing

Multipoint shared queuing is supported to minimize the number of multipoint queues created for ingress VPLS, IES or VPRN SAPs or ingress subscriber SLA profiles. Normally, ingress multipoint packets are handled by multipoint queues created for each SAP or subscriber SLA profile instance. In some instances, the number of SAPs or SLA profile instances are sufficient for the in use multipoint queues to represent many thousands of queues on an ingress forwarding plane. If multipoint shared queuing is enabled for the SAPs or SLA profile instances on the forwarding plane, the multipoint queues are not created. Instead, the ingress multipoint packets are handled by the unicast queue mapped to the forwarding class of the multipoint packet.

Functionally, multipoint shared queuing is a superset of shared queuing. With shared queuing on a SAP or SLA profile instance, only unicast packets are processed twice, once for the initial service level queuing and a second time for switch fabric destination queuing. Shared queuing does not affect multipoint packet handling. Multipoint packet handling in normal (service queuing) is the same as shared queuing. When multipoint shared queuing is enabled, shared queuing for unicast packets is automatically enabled.

---

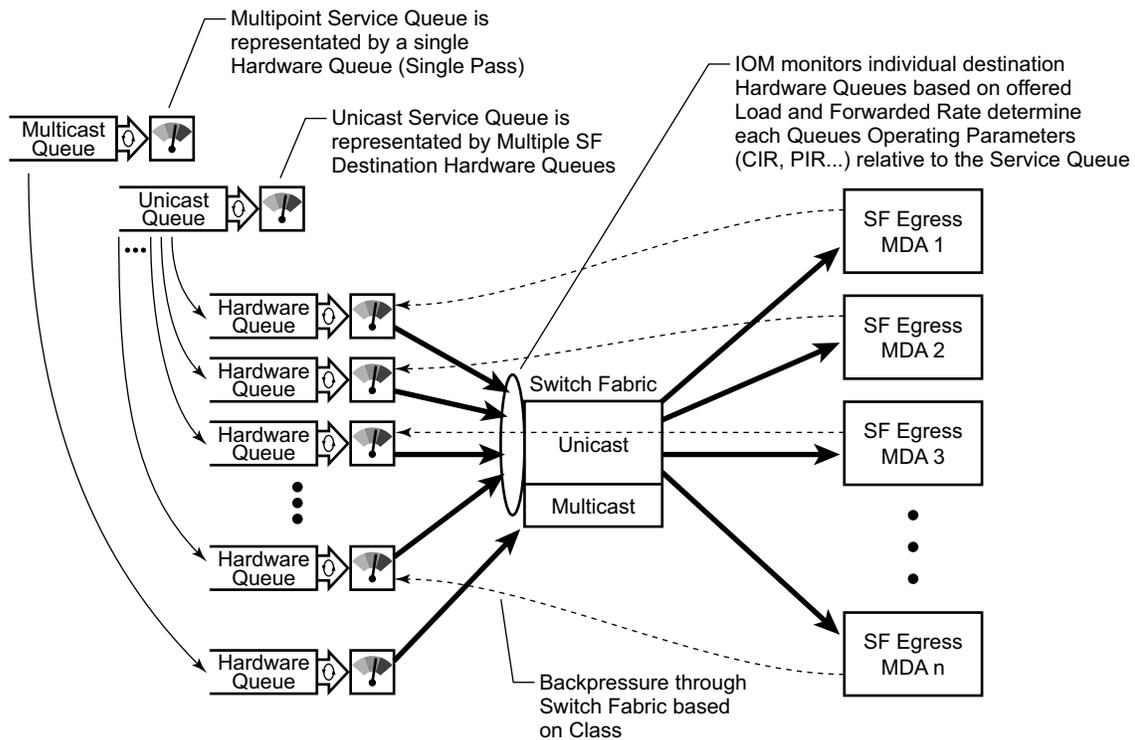
## Ingress Queuing Modes of Operation

Three modes of ingress SAP queuing are supported for multipoint services (IES, VPLS and VPRN); service, shared, and multipoint shared. The same ingress queuing options are available for IES and VPLS subscriber SLA profile instance queuing.

## Ingress Service Queuing

Normal or service queuing is the default mode of operation for SAP ingress queuing. Service queuing preserves ingress forwarding bandwidth by allowing a service queue defined in an ingress SAP QoS policy to be represented by a group of hardware queues. A hardware queue is created for each switch fabric destination to which the logical service queue must forward packets. For a VPLS SAP with two ingress unicast service queues, two hardware queues are used for each destination forwarding engine the VPLS SAP is forwarding to. If three switch fabric destinations are involved, six queues are allocated (2 unicast service queues multiplied by 3 destination forwarding complexes equals six hardware queues). **Figure 35** demonstrates unicast hardware queue expansion. Service multipoint queues in the ingress SAP QoS policy are not expanded to multiple hardware queues, each service multipoint queue defined on the SAP equates to a single hardware queue to the switch fabric.

When multiple hardware queues represent a single logical service queue, the system automatically monitors the offered load and forwarding rate of each hardware queue. Based on the monitored state of each hardware queue, the system imposes an individual CIR and PIR rate for each queue that provides an overall aggregate CIR and PIR reflective of what is provisioned on the service queue.



Fig\_22

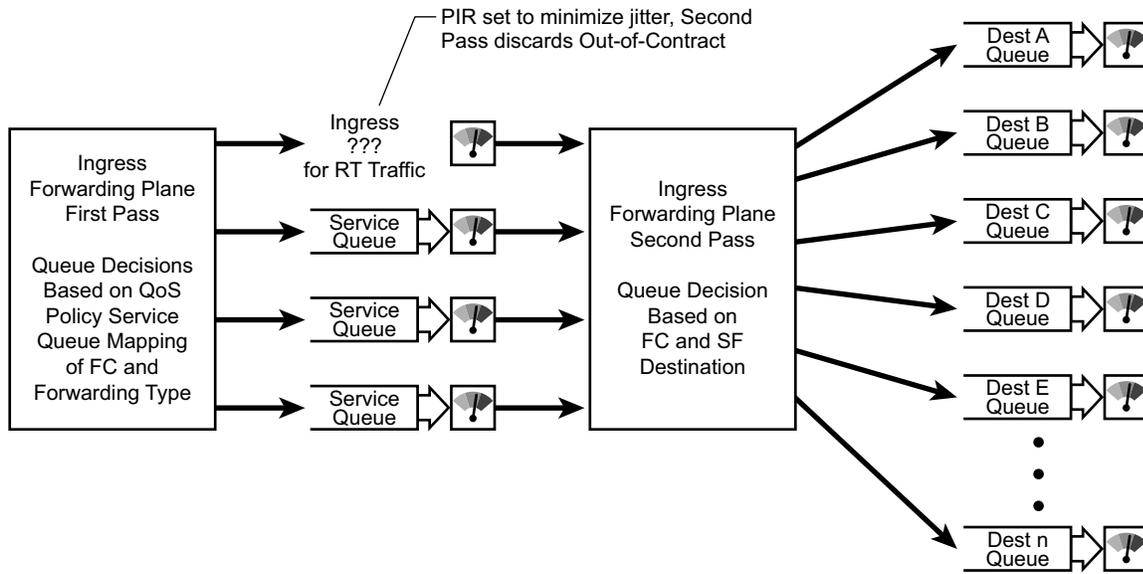
**Figure 35: Unicast Service Queue Mapping to Multiple Destination Based Hardware Queues**

### Ingress Shared Queuing

To avoid the hardware queue expansion issues associated with normal service based queuing, the system allows an ingress logical service queue to map to a single hardware queue when shared queuing is enabled. Shared queuing uses two passes through the ingress forwarding plane to separate ingress per service queuing from the destination switch fabric queuing. In the case of shared queuing, ingress unicast service queues are created one-for-one relative to hardware queues. Each hardware queue representing a service queue is mapped to a special destination in the traffic manager that 'forwards' the packet back to the ingress forwarding plane allowing a second pass through the traffic manager. In the second pass, the packet is placed into a 'shared' queue for the destination forwarding plane. The shared queues are used by all services configured for shared queuing.

When the first SAP or SLA profile instance is configured for shared queuing on an ingress forwarding plane, the system allocates eight hardware queues per available destination forwarding plane, one queue per forwarding class. Twenty-four hardware queues are also allocated for multipoint shared traffic, but that is discussed in the following section. The shared queue parameters that define the relative operation of the forwarding class queues are derived from the Shared Queue policy defined in the QoS CLI node. [Figure 36](#) demonstrates shared unicast queuing. SAP or SLA profile instance multipoint queuing is not affected by enabling shared queuing. Multipoint queues are still created as defined in the ingress SAP QoS policy and ingress multipoint packets only traverse the ingress forwarding plane a single time.

Enabling shared queuing may affect ingress performance due to double packet processing through the service and shared queues.



Fig\_23

Figure 36: Unicast Service Queuing With Shared Queuing Enabled

# Multipoint Shared Queuing

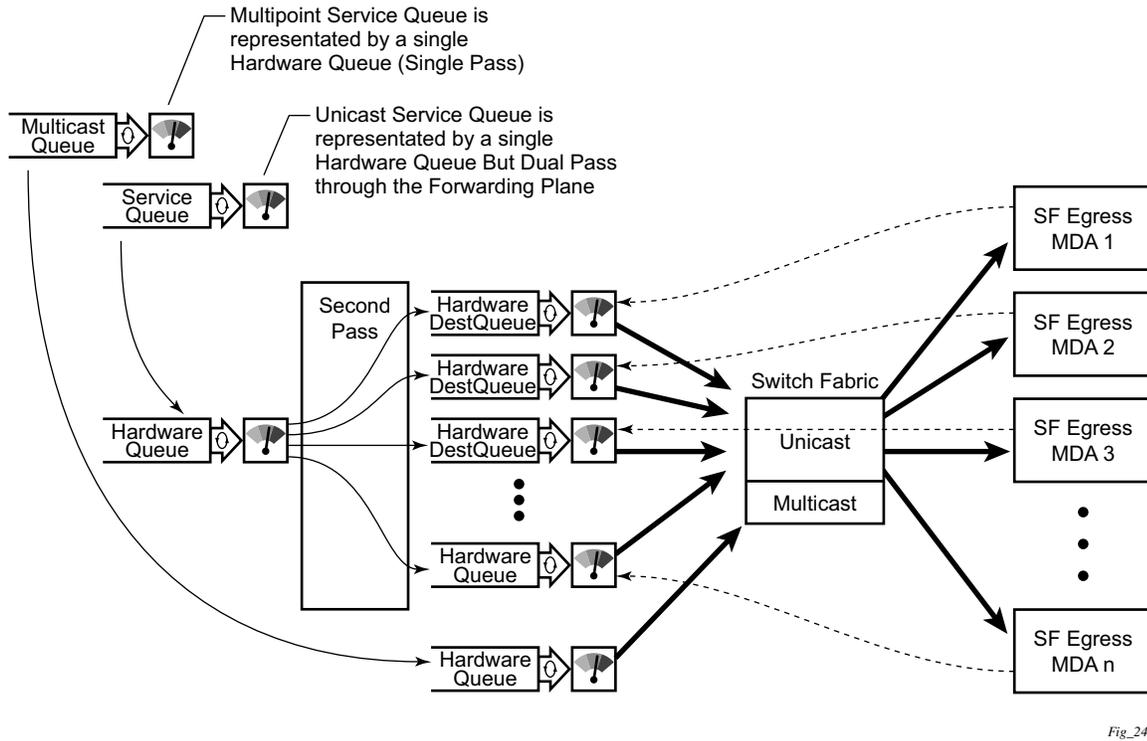


Figure 37: Multipoint Queue Behavior with Shared Queuing Enabled

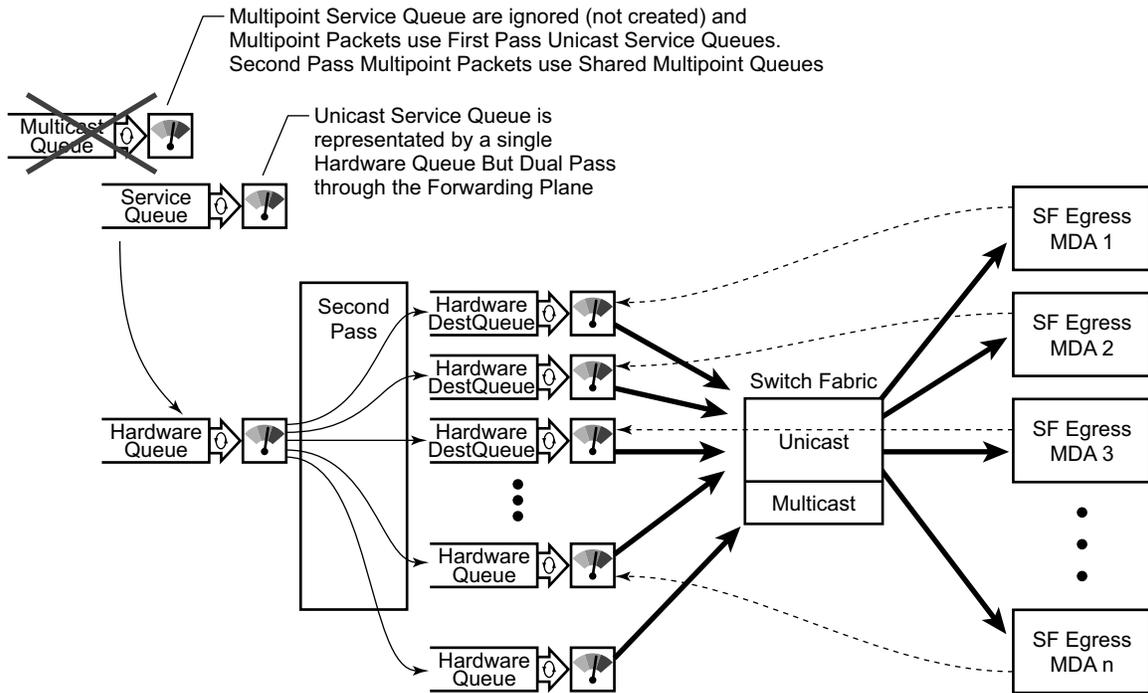
## Ingress Multipoint Shared Queuing

Ingress multipoint shared queuing is a variation to the unicast shared queuing defined in [Ingress Shared Queuing on page 692](#). Ingress unicast service queues are mapped one-for-one with hardware queues and unicast packets traverse the ingress forwarding plane twice. In addition to the above, the multipoint queues defined in the ingress SAP QoS policy are not created. Instead, multipoint packets (broadcast, multicast and unknown unicast destined) are treated to the same dual pass ingress forwarding plane processing as unicast packets. In the first pass, the forwarding plane uses the unicast queue mappings for each forwarding plane. The second pass uses the multipoint shared queues to forward the packet to the switch fabric for special replication to all egress forwarding planes that need to process the packet.

The benefit of defining multipoint shared queuing is the savings of the multipoint queues per service. By using the unicast queues in the first pass and then the aggregate shared queues in the second pass, per service multipoint queues are not required. The predominant scenario where multipoint shared queuing may be required is with subscriber managed QoS environments using a subscriber per SAP model. Usually, ingress multipoint traffic is minimal per subscriber and the extra multipoint queues for each subscriber reduces the overall subscriber density on the ingress forwarding plane. Multipoint shared queuing eliminates the multipoint queues sparing hardware queues for better subscriber density. Figure 2.3 demonstrates multipoint shared queuing.

One caveat of enabling multipoint shared queuing is that multipoint packets are no longer managed per service (although the unicast forwarding queues may provide limit benefit in this area). Multipoint packets in a multipoint service (VPLS, IES and VPRN) use significant resources in the system, consuming ingress forwarding plane multicast bandwidth and egress replication bandwidth. Usually, the per service unicast forwarding queues are not rate limited to a degree that allows adequate management of multipoint packets traversing them when multipoint shared queuing is enabled. It is possible to minimize the amount of aggregate multipoint bandwidth by setting restrictions on the multipoint queue parameters in the QoS nodes Shared Queue policy. Aggregate multipoint traffic can be managed per forwarding class for each of the three forwarding types (broadcast, multicast or unknown unicast – broadcast and unknown unicast are only used by VPLS).

Another caveat for multipoint shared queuing is that multipoint traffic now consumes double the ingress forwarding plane bandwidth due to dual pass ingress processing.



Fig\_25

**Figure 38: Multipoint Shared Queuing Using First Pass Unicast Queues**

Note that multipoint shared queuing cannot be enabled on the following services:

- Epipe
- Apipe
- Fpipe
- Ipipe
- Routed CO

For information about the tasks and commands necessary to access the command line interface and to configure and maintain your router, refer to CLI Usage chapter in the Basic System Configuration Guide.

## Basic Configurations

The default shared queue QoS policy conforms to the following:

- There is only one default shared queue policy in the system.
- The default shared queue policy has fixed forwarding classes, queues and FC-queue mapping, which cannot be modified, added, or deleted.

The only configurable entities in the default shared queue policy are the queue attributes, queue priority, and the description string. The queue priority for a shared queue can be changed to expedited, best-effort or auto-expedited.

---

## Modifying the Default Shared-Queue Policy

The only configurable entities in the default shared queue policy are the queue attributes and the description string. The changes are applied immediately to all services where this policy is applied. Use the following CLI syntax to modify a shared-queue policy:

**CLI Syntax:**

```
config>qos#
  shared-queue name
    description description-string
    queue queue-id [queue-type] [multipoint]
      cbs percent
      high-prio-only percent
      mbs percent
      rate percent [cir percent]
```

The following displays a shared-queue policy configuration example:

```
A:ALA-48>config>qos>shared-queue# info
-----
description "test1"
queue 1 create
  cbs 2
  high-prio-only 20
exit
-----
A:ALA-48>config>qos>shared-queue#
```

## Applying Shared-Queue Policies

The default shared queue policy is applied at the SAP level just as `sap-ingress` and `sap-egress` QoS policies are specified. If the **shared-queuing** keyword is not specified in the `qos policy-id` command then the SAP is assumed to use single-pass queuing.

Apply shared-queue policies to the following entities:

- [Epipe Services](#)
  - [IES Services](#)
  - [VPLS Services](#)
  - [VPRN Services](#)
- 

## Epipe Services

Use the following CLI syntax to apply QoS policies to ingress Epipe SAPs:

**CLI Syntax:** `config>service>epipe service-id [customer customer-id]  
sap sap-id  
ingress  
qos policy-id [shared-queuing]`

The following output displays an Epipe service configuration with SAP ingress policy 100 applied to the SAP with shared-queuing enabled.

```
A:SR>config>service# info
-----
epipe 6 customer 6 vpn 6 create
  description "Distributed Epipe to west coast"
  sap 1/1/10:0 create
    ingress
      qos 100 shared-queuing
    exit
  exit
  no shutdown
exit
-----
A:SR>config>service#
```

## IES Services

Use the following CLI syntax to apply the default policy to an IES service:

**CLI Syntax:** `config>service# ies service-id  
interface interface-name  
sap sap-id  
ingress  
qos policy-id [shared-queuing |multipoint-  
shared]`

The following output displays an IES service configuration with SAP ingress policy 100 applied to the SAP with shared-queuing enabled.

```
A:SR>config>service# info
-----
ies 88 customer 8 vpn 88 create
  interface "Sector A" create
    sap 1/1/1.2.2 create
      ingress
        qos 100 multipoint-shared
      exit
    exit
  exit
no shutdown
exit
-----
A:SR>config>service#
```

### VPLS Services

Use the following CLI syntax to apply the default shared-queue policy to an ingress VPLS SAP:

```
CLI Syntax: config>service# vpls service-id [customer customer-id]  
                sap sap-id  
                ingress  
                qos policy-id [shared-queuing | multipoint-shared]
```

The following output displays a VPLS service configuration with SAP ingress policy 100 with shared-queuing enabled.

```
A:SR>config>service# info  
-----  
vpls 700 customer 7 vpn 700 create  
  description "test"  
  sap 1/1/9:0 create  
    ingress  
      qos 100 multipoint-shared  
    exit  
  exit  
exit  
-----  
A:SR>config>service#
```

---

### VPRN Services

Use the following CLI syntax to apply QoS policies to ingress VPRN SAPs:

```
CLI Syntax: config>service# vprn service-id [customer customer-id]  
                interface ip-int-name  
                sap sap-id  
                ingress  
                qos policy-id [shared-queuing | multipoint-shared]
```

The following output displays a VPRN service configuration. The default SAP ingress policy was not modified but shared queuing was enabled.

```
A:SR7>config>service# info  
-----  
vprn 1 customer 1 create  
  interface "to-cel" create  
    address 11.1.0.1/24  
  sap 1/1/10:1 create  
    ingress  
      qos 1 multipoint-shared  
    exit
```

```
        exit
    exit
    no shutdown
    exit
```

```
-----
A:SR7>config>service#
```

## Default Shared Queue Policy Values

The only allowed shared queue policy is the default and cannot be deleted. The only configurable entities are the queue priority, attributes of individual queues and the description string. [Table 51](#) lists the default values.

**Table 51: Shared Queue Policy Defaults**

Field	Default
description	“Default Shared Queue Policy”
queue 1	auto-expedite
rate	100
cir	0
mbs	50
cbs	1
high-prio-only	10
queue 2	auto-expedite
rate	100
cir	25
mbs	50
cbs	3
high-prio-only	10
queue 3	auto-expedite
rate	100
cir	25
mbs	50
cbs	10
high-prio-only	10
queue 4	auto-expedite
rate	100
cir	25

**Table 51: Shared Queue Policy Defaults (Continued)**

<b>Field</b>	<b>Default</b>
mbs	25
cbs	3
high-prio-only	10
queue 5	auto-expedite
rate	100
cir	100
mbs	50
cbs	10

**Table 51: Shared Queue Policy Defaults (Continued)**

Field	Default
high-prio-only	10
queue 6	auto-expedite
rate	100
cir	100
mbs	50
cbs	10
high-prio-only	10
queue 7	auto-expedite
rate	100
cir	10
mbs	25
cbs	3
high-prio-only	10
queue 8	auto-expedite
rate	100
cir	10
mbs	25
cbs	3
high-prio-only	10

The fc-to-shared-queue mappings that cannot be modified are:

fc af	queue 3
fc be	queue 1
fc h1	queue 6
fc h2	queue 5
fc l1	queue 4

**Table 51: Shared Queue Policy Defaults (Continued)**

Field	Default
fc l2	queue 2
fc nc	queue 8

## Default Shared Queue Policy Values

The following output displays the default configuration:

```
ALA-7>config>qos>shared-queue# info detail
-----
description "Default Shared Queue Policy"
queue 1 auto-expedite create
  rate 100 cir 0
  mbs 50
  cbs 1
  high-prio-only 10
exit
queue 2 auto-expedite create
  rate 100 cir 25
  mbs 50
  cbs 3
  high-prio-only 10
exit
queue 3 auto-expedite create
  rate 100 cir 25
  mbs 50
  cbs 10
  high-prio-only 10
exit
queue 4 auto-expedite create
  rate 100 cir 25
  mbs 25
  cbs 3
  high-prio-only 10
exit
queue 5 auto-expedite create
  rate 100 cir 100
  mbs 50
  cbs 10
  high-prio-only 10
exit
queue 6 auto-expedite create
  rate 100 cir 100
  mbs 50
  cbs 10
  high-prio-only 10
exit
queue 7 auto-expedite create
  rate 100 cir 10
  mbs 25
  cbs 3
  high-prio-only 10
exit
queue 8 auto-expedite create
  rate 100 cir 10
  mbs 25
  cbs 3
  high-prio-only 10
exit
fc af create
  queue 3
exit
fc be create
  queue 1
```

```
exit
fc ef create
    queue 6
exit
fc h1 create
    queue 7
exit
fc h2 create
    queue 5
exit
fc l1 create
    queue 4
exit
fc l2 create
    queue 2
exit
fc nc create
    queue 8
exit
```

---

ALA-7>config>qos>shared-queue#

