

## Guidelines for Managing Buffers

---

Queue profiles enable you to manage queue thresholds and buffers to manage the following common problems:

- Queues that back up and consume too many buffers
- Queues that cannot obtain buffers when they need them (called *buffer starvation*)

You can set the buffer weight to ensure that some sets of queues get higher thresholds than others. Buffer weight is analogous to weight in a scheduler profile. It directs the router to set the queue thresholds proportionately.

This feature provides graceful buffer allocation as the global utilization goes higher; queues with more buffer weight always obtain more buffers, but they do not undergo a dramatic drop in threshold when the system moves from region to region.

JUNOS software uses 128-byte buffers. When setting very small queue thresholds, keep the following guidelines in mind:

- Specifying a maximum queue length of 0 bytes disables queuing of packets on the queue.
- Specifying a maximum queue length of 1–128 bytes creates a single 128-byte buffer for the queue.
- Specifying a maximum queue length of 129–256 bytes creates two 128-byte buffers for the queue.
- Packets and cells consume at least one buffer.

For example, a 64-byte packet consumes a single 128-byte buffer. If you specify a maximum queue length of 256 bytes, then either two packets of 64–128 bytes in length or a single packet of 129–256 bytes can be queued.

For example, suppose a line module with 4000 IP interfaces is configured with four queues per IP interface, corresponding to four traffic classes. Suppose that queues in two of the traffic classes are configured with a buffer weight of 24 to increase burst tolerance. The following example configures the video queue:

```
host1(config)#queue-profile video
host1(config-queue)#buffer-weight 24
host1(config-queue)#exit
host1(config)#
```

When the egress memory is fully loaded, dynamic oversubscription is 0 percent, and the 8000 queues with the default buffer weight strictly partition 25 percent of the 32-MB memory, leaving 75 percent of the memory for the queues weighted 24 (corresponding to the ratio 75 percent:25 percent, or 24:8). Therefore, these queues have committed thresholds of 1 KB each, and queues with the buffer weight of 24 have committed thresholds of 3 KB each. As the egress memory becomes progressively less loaded, all the queue thresholds increase proportionally, based on dynamic oversubscription, but the queues with buffer weight 24 are always set with thresholds three times larger than the default thresholds.

## Guidelines for Managing Buffer Starvation

Buffer starvation most commonly occurs when queues or nodes exist in a large round robin, usually in the default traffic-class group. When the round robin congests, the queues back up and require more buffers. The traffic in the round robin starts to burst based on a single node or queue. After a packet is dequeued, the node or queue can wait for thousands of other queues to dequeue a packet before it can dequeue again. During this time, the queue backs up.

If you configure different scheduler profile weights or assured rates for nodes in a large and congested round robin, the buffer starvation becomes apparent. The problem occurs when the heavy weighted nodes wait their turn in the round robin and thousands of other nodes dequeue. While the heavily weighted nodes wait, the system needs to buffer them. However, all queues receive the same buffer allocation by default. If the system goes to higher buffer regions, it starts dropping packets for all queues. When the heavy weight node finally transmits, it dequeues all buffers, but it cannot dequeue the packets that were dropped. You do not achieve the expected bandwidth based on scheduler profile weights.

To manage buffer starvation, configure buffer weights on queues so they are in the same ratio as the expected bandwidth for the queues. For example, if two queues have scheduler weight (or assured-rate) in the ratio of 2:1, then set the buffer weights to the same ratio.

To manage buffer starvation, set the **maximum-committed-threshold** on queues that do not need buffering, and increase the **buffer-weight** for the heavily weighted queues in the round robin.

The system calculates the correct ratio for you. Issue the **show egress queue rates** command to see the ratio:

```
host1# show egress-queue rates brief interface fastEthernet 9/0.2
```

interface	traffic class	forwarded rate	aggregate drop rate	minimum rate	maximum rate
ip FastEthernet9/0.2	best-effort	0	0	25000	1000000
	videoTrafficClass	0	0	375000	1000000
	multicastTrafficClass	0	0	925000	1000000
	internetTrafficClass	0	0	50000	1000000
Total:		0	0		

  

Queues reported:	4
Queues filtered (under threshold):	0
Queues disabled (no rate period):	0
Queues disabled (no resources):	0
Total queues:	4

The minimum rate for each queue is the approximate rate the queue achieves if all configured queues in the line module run infinite traffic. Configure the buffer weights in proportion to the minimum rate displayed by the system.

- Related Topics**
- Memory Requirements for Queue and Buffers
  - Configuring Queue Profiles to Manage Buffers and Thresholds

- Monitoring Forwarding and Drop Rates on the Egress Queue

