

# Output Drops Due to QoS on Access Layer Switches Troubleshooting TechNote



Document ID: 116089

Contributed by Shashank Singh, Cisco TAC Engineer.  
May 18, 2013

## Contents

### Introduction

### Prerequisites

- Requirements

- Components Used

### Background Information

### Feature Information

### Troubleshooting Methodology

### Common Problems

### Frequently Asked Questions

### Related Information

## Introduction

This document describes how to troubleshoot output drops due to quality of service (QoS) on Cisco Catalyst Switches Series 2960, 3750, 3750G, 3750X, 3560.

## Prerequisites

### Requirements

Cisco recommends that you have basic knowledge of QoS.

### Components Used

The information in this document is based on these platforms: Cisco Catalyst Switches Series 2960, 3750, 3750G, 3750X, 3560.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

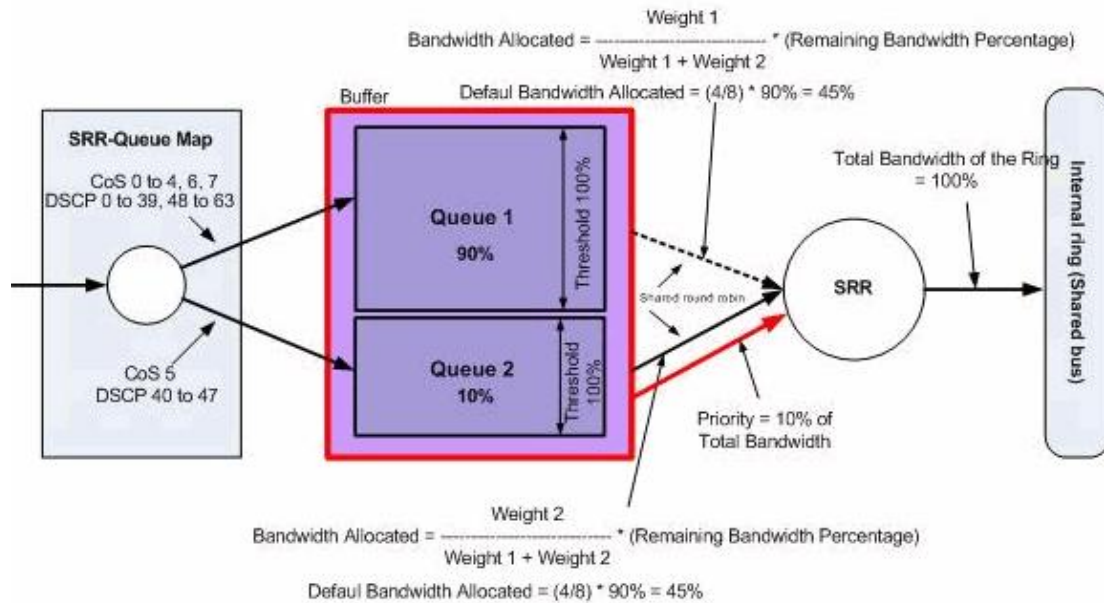
## Background Information

QoS is used to prioritize more important data during times of congestion. As a result, after QoS is enabled, less important bulk data might experience drops.

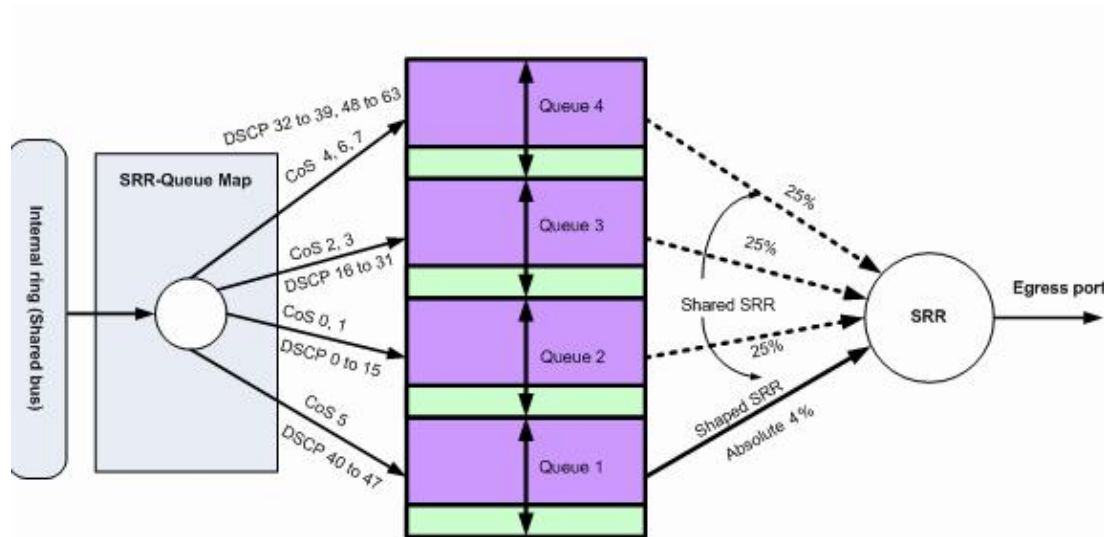
Cisco access layer switches implement QoS features in hardware. This document helps you determine if the drops are caused by QoS and describes various queueing and buffer tuning options to mitigate them.

# Feature Information

## Ingress Default Queue



## Egress Default Queue



# Troubleshooting Methodology

1. Identify the interfaces that carry outgoing data for the affected application or that experience output drops that increment. Compare the interface output rate and the interface speed and ensure that the drops are not due to over utilization of the link.

```
Switch#show int gi1/0/1
!-- Some output omitted.
GigabitEthernet0/1 is up, line protocol is up (connected)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  Full-duplex, 1000Mb/s, media type is 10/100/1000BaseTX
```

```
input flow-control is off, output flow-control is unsupported
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 1089
```

```
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 4000 bits/sec, 6 packets/sec
5 minute output rate 3009880 bits/sec, 963 packets/sec
```

2. Ensure that QoS is enabled on the switch. If it is not enabled, output drops are not related to QoS and hence further steps mentioned here are irrelevant.

```
Switch#show mls qos
QoS is enabled
QoS ip packet dscp rewrite is enabled
```

3. Identify the marking of the outgoing traffic that is dropped on the interface.

```
Switch#show mls qos int gi1/0/1 statistics
```

```
GigabitEthernet1/0/1 (All statistics are in packets)
```

```
dscp: incoming
```

```
-----
0 - 4 :      0      0      0      0      0
5 - 9 :      0      0      0      0      0
10 - 14 :    0      0      0      0      0
15 - 19 :    0      0      0      0      0
20 - 24 :    0      0      0      0      0
25 - 29 :    0      0      0      0      0
30 - 34 :    0      0      0      0      0
35 - 39 :    0      0      0      0      0
40 - 44 :    0      0      0      0      0
45 - 49 :    0 198910      0      0      0
50 - 54 :    0      0      0      0      0
55 - 59 :    0      0      0      0      0
60 - 64 :    0      0      0      0      0
```

```
dscp: outgoing
```

```
-----
0 - 4 :      0      0      0      0      0
5 - 9 :      0      0      0      0      0
10 - 14 :    0      0      0      0      0
15 - 19 :    0      0      0      0      0
20 - 24 :    0      0      0      0      0
25 - 29 :    0      0      0      0      0
30 - 34 :    0      0      0      0      0
35 - 39 :    0      0      0      0      0
40 - 44 :    0      0      0      0      0
45 - 49 :    0 248484      0      0      0
50 - 54 :    0      0      0      0      0
55 - 59 :    0      0      0      0      0
60 - 64 :    0      0      0      0      0
```

```
cos: incoming
```

```
-----
0 - 4 :      2      0      0      0      0
5 - 7 :      0      0      0      0      0
```

```
cos: outgoing
```

```
-----
0 - 4 :      0      0      0      0      0
5 - 7 :      0      0      0      0      0
```

```

output queues enqueued:
queue:      threshold1  threshold2  threshold3
-----
queue 0:    248484  0          0
queue 1:    0          0          0
queue 2:    0          0          0
queue 3:    0          0          0

output queues dropped:
queue:      threshold1  threshold2  threshold3
-----
queue 0:  1089        0          0
queue 1:    0          0          0
queue 2:    0          0          0
queue 3:    0          0          0

Policer: Inprofile:      0 OutofProfile:      0

```

**Note:** This example shows dropped packs on queue 0/threshold1 dropping packets. In other examples in the document, queue numbering is 1 – 4; therefore, this value will be queue 1.

4. Check the marking to output-q map on the switch in order to determine which queue-threshold pair maps to the marking that is dropped. In this scenario, queue1/threshold1 is mapped to dscp 46, which is dropped on the interface. This means that dscp 46 traffic is sent to queue1 and is dropped because that queue has insufficient buffer or lesser CPU cycles.

```
Switch#show mls qos maps dscp-output-q
```

```

Dscp-outputq-threshold map:
d1 :d2      0      1      2      3      4      5      6      7      8      9
-----
0 :    02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01 02-01
1 :    02-01 02-01 02-01 02-01 02-01 02-01 02-01 03-01 03-01 03-01 03-01
2 :    03-01 03-01 03-01 03-01 03-01 03-01 03-01 03-01 03-01 03-01 03-01
3 :    03-01 03-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01
4 :    01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 01-01 04-01 04-01
5 :    04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01 04-01
6 :    04-01 04-01 04-01 04-01

```

5. There are two methods to resolve these drops. The first method is to change the buffer and threshold values for the queue that drops packets. The second method is to configure the scheduler so that the queue that drops packets is serviced more often than the rest of the queues.

This steps shows how you can change the buffer and threshold for the affected queues and checks the buffer and threshold values associated with the queue identified in step 4.

**Note:** Each queue set has the option to configure the buffer size and threshold value for the four egress queues. Then, you can apply any one of the queue sets to any of the ports. By default, all interfaces use queue-set 1 for output queues unless explicitly configured to use queue-set 2.

In this scenario, queue 1 in queue-set 1 has 25% of the total buffer space and threshold 1 is set to 100%

```

Switch#show mls qos queue-set
Queueset: 1
Queue      :      1      2      3      4
-----
buffers    :      25      25      25      25
threshold1:     100     200     100     100
threshold2:     100     200     100     100
reserved   :      50      50      50      50

```

```

maximum : 400 400 400 400
Queueset: 2
Queue : 1 2 3 4
-----
buffers : 25 25 25 25
threshold1: 100 200 100 100
threshold2: 100 200 100 100
reserved : 50 50 50 50
maximum : 400 400 400 400

```

6. If you want to change the buffer and threshold values for the affected interface only, change queue-set 2 and configure the affected interface to use queue-set 2.

**Note:** You can change queue-set 1 as well; however, as all the interfaces by default use queue-set 1, the change is reflected to all interfaces.

In this example, queue-set 2 is changed so that queue 1 receives 70% of the total buffer.

```
Switch(config)#mls qos queue-set output 2 buffers 70 10 10 10
```

In this example, queue-set 2 and queue 1 thresholds are changed. Both threshold 1 and threshold 2 are mapped to 3100 so that they can pull buffer from the reserved pool if required.

```
Switch(config)#mls qos queue-set output 2 threshold 1 3100 3100 100 3200
```

7. Verify that the changes reflect under correct queue and queue-set.

```
Switch#show mls qos queue-set
Queueset: 1
Queue : 1 2 3 4
-----
buffers : 25 25 25 25
threshold1: 100 200 100 100
threshold2: 100 200 100 100
reserved : 50 50 50 50
maximum : 400 400 400 400
Queueset: 2
Queue : 1 2 3 4
-----
buffers : 70 10 10 10
threshold1: 3100 100 100 100
threshold2: 3100 100 100 100
reserved : 100 50 50 50
maximum : 3200 400 400 400

```

8. Make the affected interface use queue-set 2 so that the changes come into effect on this interface.

```
Switch(config)#int gi1/0/1
Switch(config-if)#queue-set 2
Switch(config-if)#end
```

Verify that the interface is mapped to queue-set 2.

```
Switch#show run int gi1/0/1
interface GigabitEthernet1/0/1
switchport mode access
mls qos trust dscp
queue-set 2
end
```

Check if the interface continues to drop packets.

9. You can also configure the scheduler to increase the rate at which queue 1 is serviced with the share and shape options. In this example, queue 1 alone receives 50% of the total CPU cycles and the other three queues collectively receive 50% of the CPU cycles.

```
Switch(config-if)#srr-queue bandwidth share 1 75 25 5
Switch(config-if)#srr-queue bandwidth shape 2 0 0 0
```

Check if the interface continues to drop packets.

10. Enable priority queue on this interface. This action ensures that all traffic in the priority queue is processed before any other queue.

*Note:* Priority queue is serviced until empty before the other queues are serviced. By default on 2960/3560/3750 switches, queue 1 is the priority queue.

```
Switch(config)#int gil/0/1
Switch(config-if)#priority-queue out
Switch(config-if)#end
```

Marking of the packet that is dropped on the interface can be mapped so that it goes to queue 1 (priority queue). This action ensure that traffic with this marking is always processed before anything else.

```
Switch(config)#mls qos srr-queue output dscp-map queue 1 threshold 1
```

## Common Problems

Here are some common problems:

- Output drops on interfaces after QoS is enabled.
- Choppy voice calls.
- Added delay causes suboptimal video traffic.
- Connection resets.

## Frequently Asked Questions

*Q: When do I alter the queue-set and when do I use sharing/shaping?*

*A:* The decision depends on the nature of drops. If the drops increment intermittently, this issue is most likely due to bursty traffic. On the contrary, if drops increment continuously at a constant rate, the queue that drops the packets most likely receives more data than it can send out.

For intermittent drops, the queue must have a large buffer that can accommodate occasional bursts. In order to implement this solution, you must alter the queue-set and allocate more buffer to the affected queue and increase the threshold values as well.

For continuous drops, you must configure the scheduler to service the affected queue more often and to take out more packets from the queue per CPU cycle. In order to implement this solution, you must configure the sharing/shaping on the egress queues.

*Q: What is the difference between shared mode and shaped mode?*

*A:* In shaped mode, the egress queues are guaranteed a percentage of the bandwidth, and they are rate-limited to that amount. Shaped traffic does not use more than the allocated bandwidth even if the link is idle. Shaped mode provides a more even flow of traffic over time and reduces the peaks and valleys of bursty traffic. With

shaping, the absolute value of each weight is used to compute the bandwidth available for the queues.

*srr-queue bandwidth shape weight1 weight2 weight3 weight4*

The inverse ratio ( $1/\textit{weight}$ ) controls the shaping bandwidth for this queue. In other words, queue1 is reserved  $1/\textit{weight1}$  percent of total bandwidth and so on. If you configure a weight of 0, the corresponding queue operates in shared mode. The weight specified with the *srr-queue bandwidth shape* command is ignored, and the weights specified with the *srr-queue bandwidth share interface* configuration command for a queue come into effect.

In shared mode, the queues share the bandwidth among them based on the configured weights. The bandwidth is guaranteed at this level but not limited to it. For example, if a queue is empty and no longer requires a share of the link, the remaining queues can expand into the unused bandwidth and share it among them.

*srr-queue bandwidth share weight1 weight2 weight3 weight4*

queue1 is guaranteed a minimum of  $\textit{weight1}/(\textit{weight1} + \textit{weight2} + \textit{weight3} + \textit{weight4})$  percent of the bandwidth but can also eat up into the bandwidth of other non-shaped queues if required.

## Related Information

- *Technical Support & Documentation – Cisco Systems*