

QoS Scheduling and Queueing on Catalyst 3550 Switches

Document ID: 24057

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Output Queueing Capability of Ports on Catalyst 3550 Switches

- Features That Both Gigabit and Non-Gigabit Ports Support
- Features That Only Gigabit Ports Support
- Features That Only Non-Gigabit Ports Support

CoS-to-Queue Mapping

Strict Priority Queue

Weighted Round-Robin on Catalyst 3550

WRED on Catalyst 3550 Switches

Tail Drop on Catalyst 3550 Switches

Queue Size Configuration on Gigabit Ports

Queue Management and Queue Size on Non-Gigabit Ports

Conclusion

Related Information

Introduction

Output scheduling ensures that important traffic is not dropped in the event of heavy oversubscription on the egress of an interface. This document discusses all the techniques and algorithms that are involved in output scheduling on the Cisco Catalyst 3550 switch. This document also focuses on how to configure and verify the operation of output scheduling on the Catalyst 3550 switches.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on the Catalyst 3550 that runs Cisco IOS® Software Release 12.1(12c)EA1.

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.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Output Queuing Capability of Ports on Catalyst 3550 Switches

There are two types of ports on 3550 switches:

- Gigabit ports
- Non-Gigabit ports (10/100-Mbps port)

These two ports have different capabilities. The remainder of this section summarizes these capabilities. The other sections of this document explain the capabilities in further detail.

Features That Both Gigabit and Non-Gigabit Ports Support

Each port on the 3550 has four different output queues. You can configure one of these queues as a strict priority queue. Each of the remaining queues are configured as nonstrict priority queues and are serviced with the use of weighted round-robin (WRR). On all ports, the packet is assigned to one of the four possible queues on the basis of the class of service (CoS).

Features That Only Gigabit Ports Support

Gigabit ports also support a queue management mechanism within each queue. You can configure each queue to use either weighted random early detection (WRED) or tail drop with two thresholds. Also, you can tune the size of each queue (the buffer that is assigned to each queue).

Features That Only Non-Gigabit Ports Support

Non-Gigabit ports do not have any queuing mechanism such as WRED or tail drop with two thresholds. Only FIFO queuing on a 10/100-Mbps port is supported. You cannot change the size of each of the four queues on these ports. However, you can assign a minimum (min) reserve size per queue.

CoS-to-Queue Mapping

This section discusses how the 3550 decides to place each packet in a queue. The packet is placed in the queue on the basis of the CoS. Each of the eight possible CoS values is mapped to one of the four possible queues with use of the CoS-to-queue map interface command that this example shows:

```
(config-if)#wrr-queue cos-map queue-id cos1... cos8
```

Here is an example:

```
3550(config-if)#wrr-queue cos-map 1 0 1
3550(config-if)#wrr-queue cos-map 2 2 3
3550(config-if)#wrr-queue cos-map 3 4 5
3550(config-if)#wrr-queue cos-map 4 6 7
```

This example places:

- CoS 0 and 1 in queue 1 (Q1)
- CoS 2 and 3 in Q2
- CoS 4 and 5 in Q3
- CoS 6 and 7 in Q4

You can issue this command in order to verify the CoS-to-queue mapping of a port:

```
cat3550#show mls qos interface gigabitethernet0/1 queueing
GigabitEthernet0/1
...Cos-queue map:
cos-qid
0 - 1
1 - 1
2 - 2
3 - 2
4 - 3
5 - 3
6 - 4
7 - 4...
```

Strict Priority Queue

A strict priority queue is always emptied first. So, as soon as there is a packet in the strict priority queue, the packet is forwarded. After each packet is forwarded from one of the WRR queues, the strict priority queue is checked and emptied if necessary.

A strict priority queue is specially designed for delay/jitter-sensitive traffic, such as voice. A strict priority queue can eventually cause starvation of the other queues. Packets that are placed in the three other WRR queues are never forwarded if a packet waits in the strict priority queue.

Tips

In order to avoid starvation of the other queues, pay special attention to what traffic is placed in the priority queue. This queue is typically used for voice traffic, the volume of which is typically not very high. However, if someone is able to send high-volume traffic with CoS priority to the strict priority queue (such as large file transfer or backup), the starvation of other traffic can result. In order to avoid this problem, special traffic needs to be placed in the classification/admission and marking of the traffic in the network. For example, you can take these precautions:

- Make use of the untrusted port QoS status for all untrusted source ports.
- Make use of the trusted boundary feature for the Cisco IP Phone port in order to ensure that it is not used in the trust state that is configured for an IP phone for another application.
- Police the traffic that goes to the strict priority queue. Set a limit for policing traffic with a CoS of 5 (differentiated services code point [DSCP] 46) to 100 MB on a Gigabit port.

For more information on these topics, refer to these documents:

- Understanding QoS Policing and Marking on the Catalyst 3550
- *Configuring a Trusted Boundary to Ensure Port Security* section of Configuring QoS (Catalyst 3500)

On the 3550, you can configure one queue to be the priority queue (which is always Q4). Use this command in interface mode:

```
3550(config-if)#priority-queue out
```

If the priority queue is not configured in an interface, Q4 is considered as a standard WRR queue. The Weighted Round–Robin on Catalyst 3550 section of this document provides more details. You can verify if the strict priority queue is configured on an interface if you issue the same Cisco IOS command:

```
NifNif#show mls qos interface gigabitethernet0/1 queueing
GigabitEthernet0/1
Egress expedite queue: ena
```

Weighted Round–Robin on Catalyst 3550

WRR is a mechanism that is used in output scheduling on the 3550. WRR works between three or four queues (if there is no strict priority queue). The queues that are used in the WRR are emptied in a round–robin fashion, and you can configure the weight for each queue.

For example, you can configure weights so that the queues are served differently, as this list shows:

- Serve WRR Q1: 10 percent of the time
- Serve WRR Q2: 20 percent of the time
- Serve WRR Q3: 60 percent of the time
- Serve WRR Q4: 10 percent of the time

For each queue, you can issue these commands in interface mode in order to configure the four weights (with one associated to each queue):

```
(config-f)#wrr-queue bandwidth weight1 weight2 weight3 weight4
```

Here is an example:

```
3550(config)#interface gigabitethernet 0/1
3550(config-if)#wrr-queue bandwidth 1 2 3 4
```

Note: The weights are relative. These values are used:

- $Q1 = \text{weight } 1 / (\text{weight1} + \text{weight2} + \text{weight3} + \text{weight4}) = 1 / (1+2+3+4) = 1/10$
- $Q2 = 2/10$
- $Q3 = 3/10$
- $Q4 = 4/10$

WRR can be implemented in these two ways:

- **WRR per bandwidth:** Each weight represents a specific bandwidth that is allowed to be sent. Weight Q1 is allowed to have roughly 10 percent of the bandwidth, Q2 gets 20 percent of the bandwidth, and so on. This scheme is only implemented in the Catalyst 6500/6000 series at this time.
- **WRR per packet:** This is the algorithm that is implemented in the 3550 switch. Each weight represents a certain number of packets that are to be sent, regardless of their size.

As the 3550 implements WRR per packet, this behavior applies to the configuration in this section:

- Q1 transmits 1 packet out of 10
- Q2 transmits 2 packets out of 10
- Q3 transmits 3 packets out of 10
- Q4 transmits 4 packets out of 10

Packets to be transmitted can all be the same size. You still reach an expectable sharing of bandwidth among the four queues. However, if the average packet size is different between the queues, there is a big impact on what is transmitted and dropped in the event of congestion.

For example, assume that you have only two flows present in the switch. Hypothetically, also assume that these conditions are in place:

- One Gbps of small interactive application traffic (80-byte [B] frames) with a CoS of 3 is placed in Q2.
- One Gbps of large-file transfer traffic (1518-B frames) with a CoS of 0 is placed in Q1.

Two queues in the switch are sent with 1 Gbps of data.

Both streams need to share the same output Gigabit port. Assume that equal weight is configured between Q1 and Q2. WRR is applied per packet, and the amount of data that are transmitted from each queue differs between the two queues. The same number of packets are forwarded out of each queue, but the switch actually sends this amount of data:

- 77700 packets per second (pps) out of Q2 = (77700 x 8 x 64) bits per second (bps) (approximately 52 Mbps)
- 77700 pps out of Q1 = (77700 x 8 x 1500) bps (approximately 948 Mbps)

Tips

- If you want to allow fair access for each queue to the network, take into account the average size of each packet. Each packet is expected to be placed in one queue and the weight modified accordingly. For example, if you want to give equal access to each of the four queues, such that every queue gets 1/4 of the bandwidth, the traffic is as follows:
 - ◆ In Q1: Best effort Internet traffic. Assume that traffic has an average packet size of 256 B.
 - ◆ In Q2: Backup composed of file transfer, with a packet mainly of 1500 B.
 - ◆ In Q3: Video streams, which are done on packets of 192 B.
 - ◆ In Q4: Interactive application that is composed of mainly a packet of 64 B.

This creates these conditions:

- ◆ Q1 consumes 4 times the bandwidth of Q4.
- ◆ Q2 consumes 24 times the bandwidth of Q4.
- ◆ Q3 consumes 3 times the bandwidth of Q4.
- In order to have equal bandwidth access to the network, configure:
 - ◆ Q1 with a weight of 6
 - ◆ Q2 with a weight of 1
 - ◆ Q3 with a weight of 8
 - ◆ Q4 with a weight of 24
- If you assign these weights, you achieve an equal bandwidth sharing among the four queues in the event of congestion.
- If the strict priority queue is enabled, the WRR weights are redistributed among the three remaining queues. If strict priority queue is enabled and Q4 is not configured, the first example with weights of 1, 2, 3, and 4 is:
 - ◆ $Q1 = 1 / (1+2+3) = 1$ packet out of 6
 - ◆ $Q2 = 2$ packets out of 6
 - ◆ $Q3 = 3$ packets out of 6

You can issue this Cisco IOS Software **show** command in order to verify the queue weight:

```
NifNif#show mls qos interface gigabitethernet0/1 queueing
GigabitEthernet0/1
QoS is disabled. Only one queue is used
When QoS is enabled, following settings will be applied
Egress expedite queue: dis
wrr bandwidth weights:
qid-weights
 1 - 25
 2 - 25
 3 - 25
 4 - 25
```

If the expedite priority queue is enabled, the Q4 weight is only used if the expedite queue gets disabled. Here is an example:

```
NifNif#show mls qos interface gigabitethernet0/1 queueing
GigabitEthernet0/1
Egress expedite queue: ena
wrr bandwidth weights:
qid-weights
 1 - 25
 2 - 25
 3 - 25
 4 - 25
```

!--- The expedite queue is disabled.

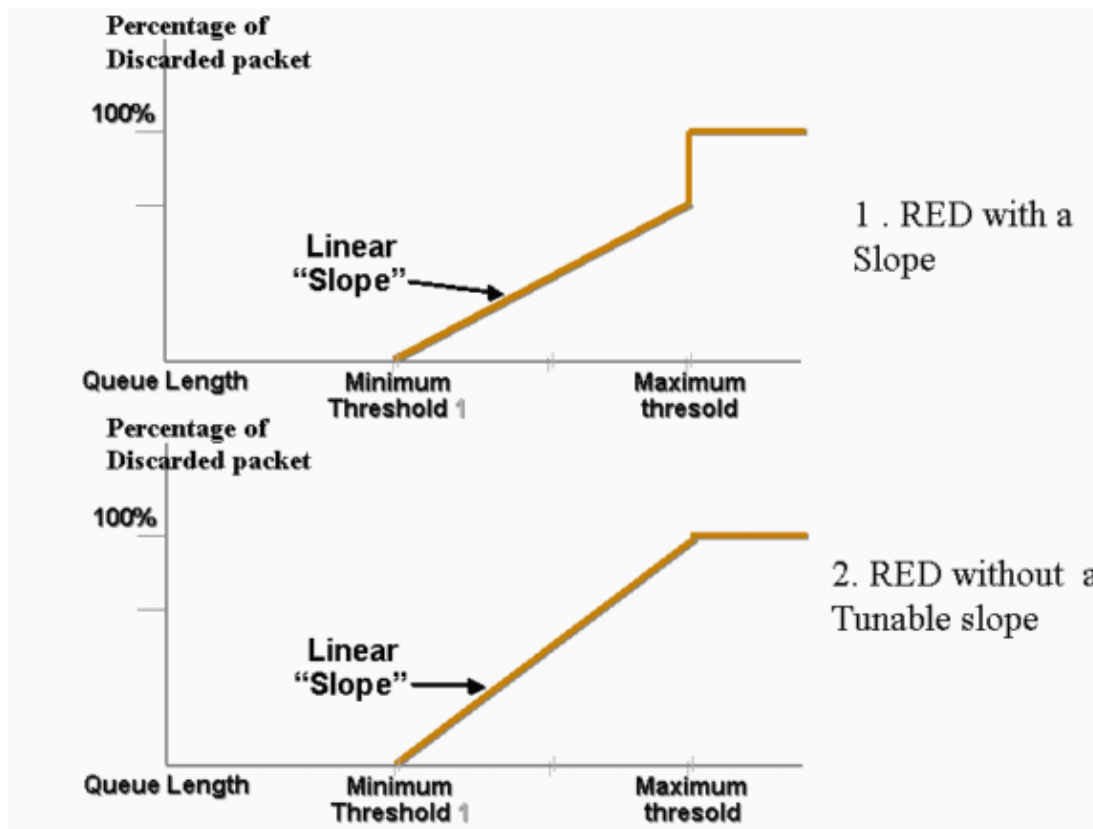
WRED on Catalyst 3550 Switches

WRED is only available on Gigabit ports on the 3550 series switches. WRED is a modification of random early detection (RED), which is used in congestion avoidance. RED has these parameters defined:

- **Min threshold:** Represents a threshold within a queue. No packets are dropped below this threshold.
- **Maximum (max) threshold:** Represents another threshold within a queue. All packets are dropped above the max threshold.
- **Slope:** Probability to drop the packet between the min and the max. The drop probability increases linearly (with a certain slope) with the queue size.

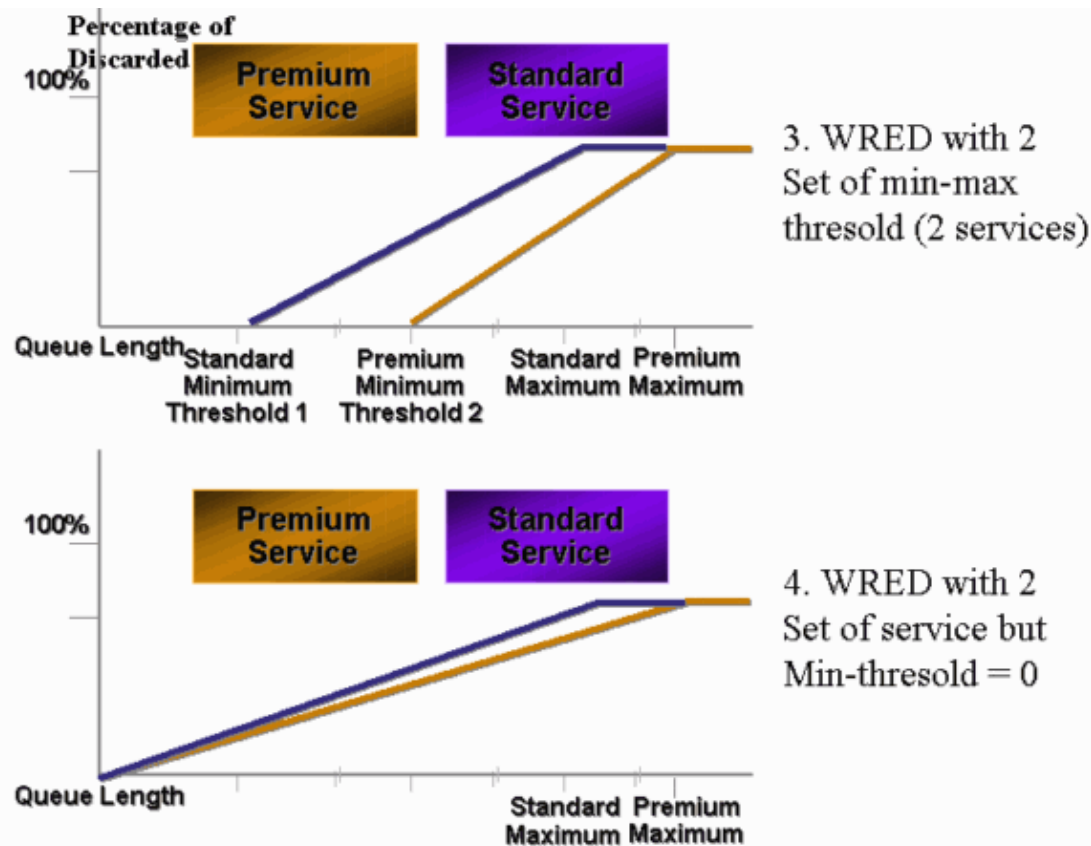
This graph shows the drop probability of a packet in the RED queue:

Note: All Catalyst switches that implement RED allow you to tune the slope.



In WRED, different services are weighted. You can define a standard service and a premium service. Each service receives a different set of thresholds. Only packets that are assigned to the standard service are dropped when min threshold 1 is reached. Only packets from premium services begin to be dropped when min threshold 2 is reached. If min threshold 2 is higher than min threshold 1, more packets from the standard service are dropped than are packets from the premium services. This graph shows an example of the drop probability for each service with WRED:

Note: The 3550 switch does not allow you to tune the min threshold, but only the max threshold. The min threshold is always hard set to 0. This gives a drop probability that represents what is currently implemented in the 3550.



Any queue that is enabled for WRED on the 3550 always has a nonzero drop probability and always drops packets. This is the case because the min threshold is always 0. If you need to avoid packet drop at max, use weighted tail drop, which the Tail Drop on Catalyst 3550 Switches section describes.

Tip: Cisco bug ID CSCdz73556 [\(registered customers only\)](#) documents an enhancement request for the configuration of min threshold.

For more information on RED and WRED, refer to the Congestion Avoidance Overview.

On the 3550, you can configure WRED with two different max thresholds in order to provide two different services. Different types of traffic are assigned to either threshold, which depends only on the internal DSCPs. This differs from the queue assignment, which only depends on the CoS of the packet. A DSCP-to-threshold table mapping decides to which threshold each of the 64 DSCPs goes. You can issue this command in order to see and modify this table:

```
(config-if)#wrr-queue dscp-map threshold_number DSCP_1 DSCP_2 DSCP_8
```

For example, this command assigns DSCP 26 to threshold 2:

```
NifNif(config-if)#wrr-queue dscp-map 2 26
NifNif#show mls qos interface gigabitEthernet0/1 queueing
GigabitEthernet0/1
Dscp-threshold map:
  d1 :  d2 0  1  2  3  4  5  6  7  8  9
-----
  0 :    01 01 01 01 01 01 01 01 01 01 01
  1 :    01 01 01 01 01 01 01 02 01 01 01
  2 :    01 01 01 01 01 02 01 02 01 01 01
  3 :    01 01 01 01 01 01 01 01 01 01 01
  4 :    02 01 01 01 01 01 02 01 01 01
  5 :    01 01 01 01 01 01 01 01 01 01 01
```


After definition of the DSCP-to-threshold map, WRED is enabled on the queue of your choice. Issue this command:

```
(config-if)#wrr-queue random-detect max-threshold queue_id threshold_1 threshold_2
```

This example configures:

- Q1 with threshold 1 = 50 percent and threshold 2 = 100 percent
- Q2 with threshold 1 = 70 percent and threshold 2 = 100 percent

```
3550(config)#interface gigabitEthernet 0/1
3550(config-if)#wrr-queue random-detect max-threshold 1 50 100
3550(config-if)#wrr-queue random-detect max-threshold 2 70 100
3550(config-if)#wrr-queue random-detect max-threshold 3 50 100
3550(config-if)#wrr-queue random-detect max-threshold 4 70 100
```

You can issue this command in order to verify the type of queuing (WRED or not) on each queue:

```
nifnif#show mls qos interface gigabitEthernet0/1 buffers
GigabitEthernet0/1
..
qid WRED thresh1 thresh2
1 dis 10 100
2 dis 10 100
3 ena 10 100
4 dis 100 100
```

The `ena` means enable, and the queue uses WRED. The `dis` means disable, and the queue uses tail drop.

You can also monitor the number of packets that are dropped for each threshold. Issue this command:

```
show mls qos interface gigabitEthernetx/x statistics
WRED drop counts:
qid thresh1 thresh2 FreeQ
1 : 327186552 8 1024
2 : 0 0 1024
3 : 37896030 0 1024
4 : 0 0 1024
```

Tail Drop on Catalyst 3550 Switches

Tail drop is the default mechanism on the 3550 on Gigabit ports. Each Gigabit port can have two tail drop thresholds. A set of DSCPs are assigned to each of the tail drop thresholds with use of the same DSCP threshold map that the WRED on Catalyst 3550 Switches section of this document defines. When a threshold is reached, all packets with a DSCP that is assigned to that threshold are dropped. You can issue this command in order to configure tail drop thresholds:

```
(config-if)#wrr-queue threshold queue-id threshold-percentage1 threshold-percentage2
```

This example configures:

- Q1 with tail drop threshold 1 = 50 percent and threshold 2 = 100 percent
- Q2 with threshold 1 = 70 percent and threshold 2 = 100 percent

```
Switch(config-if)#wrr-queue threshold 1 50 100
Switch(config-if)#wrr-queue threshold 2 70 100
Switch(config-if)#wrr-queue threshold 3 60 100
Switch(config-if)#wrr-queue threshold 4 80 100
```

Queue Size Configuration on Gigabit Ports

The 3550 switch uses central buffering. This means that there are no fixed buffer sizes per port. However, there is a fixed number of packets on a Gigabit port that can be queued. This fixed number is 4096. By default, each queue in a Gigabit port can have up to 1024 packets, regardless of the packet size. However, you can modify the way in which these 4096 packets are split among the four queues. Issue this command:

```
wrr-queue queue-limit Q_size1 Q_size2 Q_size3 Q_size4
```

Here is an example:

```
3550(config)#interface gigabitethernet 0/1
3550(config-if)#wrr-queue queue-limit 4 3 2 1
```

These queue size parameters are relative. This example shows that:

- Q1 is four times larger than Q4.
- Q2 is three times larger than Q4.
- Q3 is twice as large as Q4.

The 4096 packets are redistributed in this way:

- $Q1 = [4 / (1+2+3+4)] * 4096 = 1639$ packets
- $Q2 = 0.3 * 4096 = 1229$ packets
- $Q3 = 0.2 * 4096 = 819$ packets
- $Q4 = 0.1 * 4096 = 409$ packets

This command allows you to see the relative weights of split buffers among the four queues:

```
cat3550#show mls qos interface buffers
GigabitEthernet0/1
Notify Q depth:
qid-size
1 - 4
2 - 3
3 - 2
4 - 1
...
```

You can also issue this command in order to see how many free packets each queue can still hold:

```
(config-if)#show mls qos interface gigabitethernetx/x statistics
WRED drop counts:
qid  thresh1  thresh2  FreeQ
1 : 0      0      1639
2 : 0      0      1229
3 : 0      0      819
4 : 0      0      409
```

The FreeQ count parameter is dynamic. The FreeQ counter gives the maximum queue size minus the number of packets that are currently in the queue. For example, if there are currently 39 packets in Q1, 1600 packets are free in the the FreeQ count. Here is an example:

```
(config-if)#show mls qos interface gigabitethernetx/x statistics
WRED drop counts:
qid  thresh1  thresh2  FreeQ
1 : 0      0      1600
2 : 0      0      1229
3 : 0      0      819
4 : 0      0      409
```

Queue Management and Queue Size on Non-Gigabit Ports

There is no queue management scheme available on 10/100-Mbps ports (no WRED or tail drop with two thresholds). All four queues are FIFO queues. There is also no maximum queue size that reserves 4096 packets for each Gigabit port. 10/100-Mbps ports store packets in each queue until it is full because of a lack of resources. You can reserve a minimum number of packets per queue. This minimum is set to 100 packets per queue by default. You can modify this min reserve value for each queue if you define different min reserve values and assign one of the values to each queue.

Complete these steps in order to make this modification:

1. Assign a buffer size for each global min reserve value.

You can configure a maximum of eight different min reserve values. Issue this command:

```
(Config)# mls qos min-reserve min-reserve-level min-reserve-buffersize
```

These min reserve values are global to the switch. By default, all min reserve values are set to 100 packets.

For example, in order to configure a min reserve level 1 of 150 packets and a min reserve level 2 of 50 packets, issue these commands:

```
nifnif(config)#mls qos min-reserve ?
<1-8>  Configure min-reserve level
nifnif(config)#mls qos min-reserve 1 ?
<10-170>  Configure min-reserve buffers
nifnif(config)#mls qos min-reserve 1 150
nifnif(config)#mls qos min-reserve 2 50
```

2. Assign one of the min reserve values to each of the queues.

You must assign each of the queues to one of the min reserve values in order to know how many buffers are guaranteed for this queue. By default, these conditions apply:

- ◆ Q1 is assigned to min reserve level 1.
- ◆ Q2 is assigned to min reserve level 2.
- ◆ Q3 is assigned to min reserve level 3.
- ◆ Q4 is assigned to min reserve level 4.

By default, all min reserve values are 100.

You can issue this interface command in order to assign a different min reserve value per queue:

```
(config-if)#wrr-queue min-reserve queue-id min-reserve-level
```

For example, in order to assign to Q1 a min reserve of 2 and to Q2 a min reserve of 1, issue this command:

```
nifnif(config)#interface fastethernet 0/1
nifnif(config-if)#wrr-queue min-reserve ?
<1-4> queue id
nifnif(config-if)#wrr-queue min-reserve 1 ?
<1-8> min-reserve level
nifnif(config-if)#wrr-queue min-reserve 1 2
nifnif(config-if)#wrr-queue min-reserve 2 1
```

You can issue this command in order to verify the min reserve assignment that results:

```
nifnif#show mls qos interface fastethernet0/1 buffers
FastEthernet0/1
Minimum reserve buffer size:
150 50 100 100 100 100 100 100

!--- This shows the value of all eight min reserve levels.

Minimum reserve buffer level select:
2 1 3 4

!--- This shows the min reserve level that is assigned to
!--- each queue (from Q1 to Q4).
```

Conclusion

Queuing and scheduling on a port on the 3550 involves these steps:

1. Assign each CoS to one of the queues.
2. Enable strict priority queues, if needed.
3. Assign the WRR weight, and take into account the expected packet size within the queue.
4. Modify the queue size (Gigabit ports only).
5. Enable a queue management mechanism (tail drop or WRED, on Gigabit ports only).

Proper queuing and scheduling can reduce delay/jitter for voice/video traffic and avoid loss for mission-critical traffic. Be sure to adhere to these guidelines for maximum scheduling performance:

- Classify the traffic that is present in the network in different classes, with either trusting or specific marking.
- Police traffic in excess.

Related Information

- [Understanding QoS Policing and Marking on the Catalyst 3550](#)
- [Configuring QoS – Product Documentation](#)
- [LAN Product Support Pages](#)
- [LAN Switching Support Page](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2013 – 2014 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)