Table of Contents

QoS Scheduling and Queueing on Catalyst 3550 Switches

Document ID: 24057

Please provide your feedback on this document.

Introduction

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

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

There are no specific prerequisites for this document.

Components Used

This document was written using a Catalyst 3550 running software 12.1(12c)EA1.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Output Queueing Capability of Ports on Catalyst 3550 Switches

There are two types of ports on 3550 switches: Gigabit ports and non−Gigabit ports (10/100M port). These two ports have different capabilities. These capabilities are summarized below, and explained in further detail throughout this document.

Features Supported by Both Gigabit and Non−Gigabit Ports

Each port on the 3550 has four different output queues. One of these queues can be configured as a strict priority queue. Each of the remaining ports are configured as non−strict priority queues, and are serviced using Weighted Round Robin (WRR). On all ports, the packet is assigned to one of the four possible queues based on their Class of Service (CoS).

Features Only Supported by Gigabit Ports

Gigabit ports also support a queue management mechanism within each queue. Each queue can be configured to use either Weighted Random Early Discard (WRED) or tail drop with two thresholds. The size of each queue can also be tuned (buffer assigned to each queue).

Features Only Supported by Non−Gigabit Ports

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 port is supported. You cannot change the size of each of the four queues on these ports. You can however, assign a minimum 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 based on the CoS. Each of the eight possible CoS values will be mapped to one of the four possible queues using the CoS to queue map interface commands shown below.

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

An example is shown below.

```
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 Q1, CoS 2 and 3 in Q2, CoS 4 and 5 in Q3, and CoS 6 and 7 in Q4.

The CoS to queue mapping of a port can be verified by issuing the following command, as shown below.

```
cat3550# sh mls qos int gig 0/1 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. This means that 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 placed in the three other WRR queues will never be forwarded if a packet is waiting in the strict priority queue.

Tips

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, which is usually not very high volume traffic. If someone is able however, to send high volume traffic with CoS priority going to the strict priority queue (such as huge file transfer or back up), this could lead to starvation of other traffic. 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 may want to take the following precautions:

- Making use of untrusted port qos status for all untrusted source ports.
- Making use of the trusted boundary feature for the Cisco IP phone port to make sure it is not used in the trust state 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 (DSCP 46) to 100M on a Gigabit port.

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

- Understanding QoS Policing and Marking on the Catalyst 3550
- Trusted Boundary on the Catalyst 3550

On the 3550, you can configure one queue to be the priority queue (always Q4) using the following 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 (this is described in more detail in the next section). You can verify whether the strict priority queue is configured on an interface by issuing the same IOS command, as shown below.

```
NifNif#sh mls qos interface gig 0/1 queueing
GigabitEthernet0/1
Egress expedite queue: ena
```

Weighted Round Robin on Catalyst 3550

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

For example, weights can be configured so that the queues are served differently, as shown below.

- Serving WRR Q1 : 10% of time
- Serving WRR Q2 : 20% of time
- Serving WRR Q3 : 60% of time
- Serving WRR Q4 : 10% of time

For each queue, you can configure the four weights (one associated to each queue) by issuing the following commands in interface mode:

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

An example is shown below.

```
3550(config)# interface gigabitethernet0/1
3550(config−if)# wrr−queue bandwidth 1 2 3 4
```

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

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

WRR can be implemented in the following two ways:

1. **WRR per bandwidth:** Each weight represents a specific bandwidth allowed to be sent. Weight Q1 would be allowed to have roughly 10% of the bandwidth, Q2 would get 20% of the bandwidth, and so on. This scheme is only implemented in the Catalyst 6000 family at this time.
2. **WRR per packet:** This is the algorithm implemented in the 3550 switch. This means that each weight represents a certain number of packets to be sent, regardless of their size.

As the 3550 implements WRR per packet, the following behavior applies to the configuration shown above:

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

It is fine if packets to be transmitted are all the same size. You will still reach an expectable sharing of bandwidth among the four queues. If the average packet size is different between the queues however, this will have a huge 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. Also assume, hypothetically, that the following conditions are in place:

- One Gigabit per second of small interactive application traffic (80 bytes frames) with a CoS of 3 placed in Q2.
- One Gigabit per second of large− file transfer traffic (1518 bytes frames) with a CoS of 0 placed in Q1.

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

Both streams need to share the same output Gigabit port. Assume that We also equal weight is configured between Q1 and Q2. WRR is applied per packet, and the amount of data transmitted from each queue will be different between the two queues. The same amount of packets are forwarded out of each queue, yet the switch actually sends the following amounts of data:

- 77700 packets per second out of Q2 = (77700 x 8 x 64) bits/sec (around 52 Mbps)
- 77700 packets per second out of Q1 = (77700 x 8 x 1500) bits/sec (around 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 (every queue should get 1/4 of the bandwidth), the traffic will be as follows:
    - ♦ In Q1: Best effort Internet traffic. Assume that traffic has an average packet size of 256 bytes.
    - ♦ In Q2 : Backup composed of file transfer with a packet mainly of 1500 bytes.
    - ♦ In Q3 : Video streams, which are done on packets of 192 bytes.
    - ♦ In Q4 : Interactive application composed of mainly a packet of 64 bytes,

  This creates the following 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.
- To have equal bandwidth access to the network, configure the following:
    - ♦ 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 the above weights are assigned, an equal bandwidth sharing among the four queues will be achieved in the event of congestion.
- If the strict priority queue is enabled, the WRR weights are redistributed among the three remaining queues. The first example with weights of 1, 2, 3, and 4 would be as follows if strict priority queue is enabled, and Q4 is not configured:
    - ♦ Q1 = 1 / (1+2+3) = 1 packet out of 6
    - ♦ Q2 = 2 packets out of 6
    - ♦ Q3 = 3 packets out of 6

  The queue weight can be verified by issuing the following IOS **show** command:

```
NifNif#sh mls qos interface gig 0/1 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 in the event that the expedite queue gets disabled . An example is shown below.

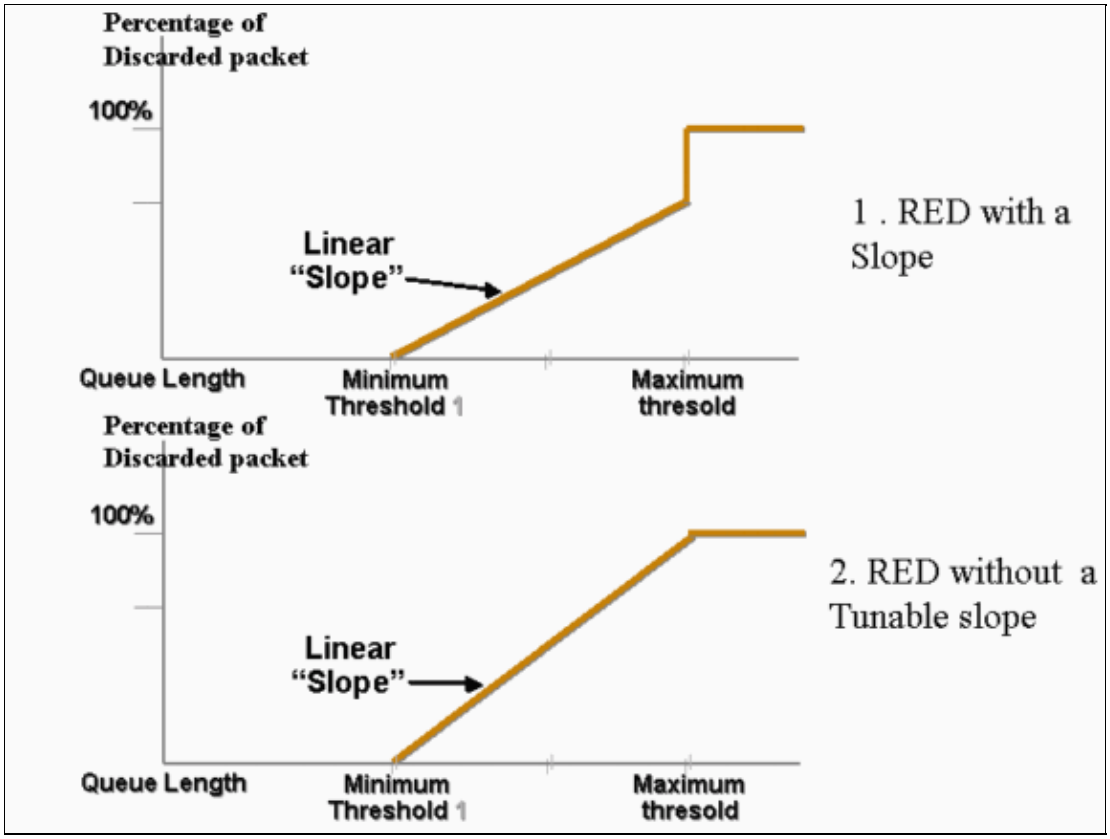
```
NifNif#sh mls qos interface gig 0/1 queueing
GigabitEthernet0/1
Egress expedite queue: ena
wrr bandwidth weights:
qid−weights
 1 − 25
 2 − 25
 3 − 25
 4 − 25


!−−− The expedite queue is disabled.
```

WRED on Catalyst 3550 Switches

WRED is only available on Gigabit ports on the 3550 family switches. WRED is a modification of RED , which is used in congestion avoidance. RED has the following parameters defined:

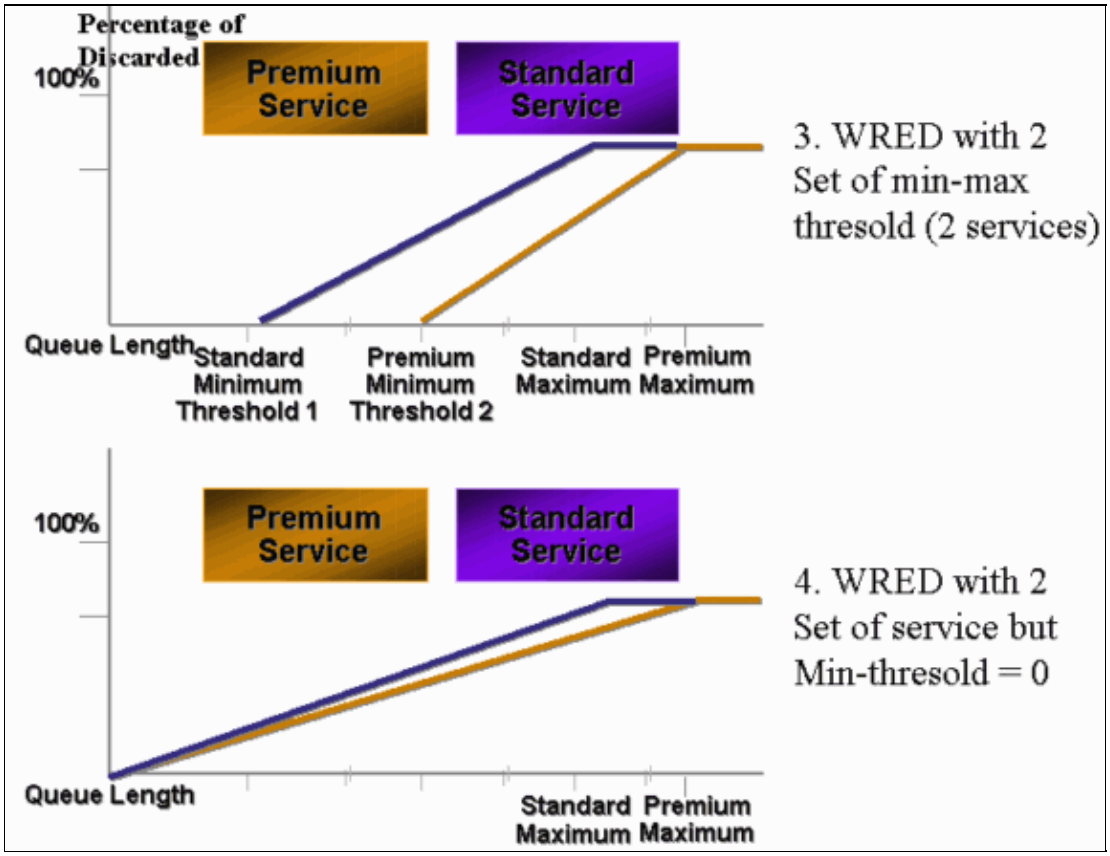
- **min−threshold:** represents a threshold within a queue. No packets are dropped below this threshold.
- **max−threshold:** represents another threshold within a queue. All packets are dropped above the max−threshold.
- **slope:** probability to drop the packet between min and max. The drop probability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

The drop probability of a packet in the RED queue is shown in the graph below. Note that all Catalyst switches that do implement RED allow for tuning 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 assigned to the standard service are dropped when min−thresold 1 is reached. Only packets from premium services will begin to be dropped when min−thresold 2 is reached. If min−thresold 2 is higher than min−thresold 1, more packets from the standard service are dropped than from the premium services. The graph below shows an example of the drop probability for each service with WRED.

The 3550 switch does not allow tuning the min−thresold, only the max−threshold. Min−threshold is always hard set to 0. This gives a drop probability representing what is currently implemented in the 3550.



Any queue enabled for WRED on the 3550 will always have a non−zero drop probability, and will always drop packets. This is a result of the min−thresold always being 0. If packet drop needs to be avoided at maximum, it is best to use weighted tail drop, as described in the next section.

Tip: An enhancement request for configuring min−thresold is documented in Bug ID CSCdz73556. To learn more details about this bug, use the Bug Toolkit (registered customers only) .

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

On the 3550, WRED can be configured with two different max−thresholds to provide two different services. Different types of traffic will be assigned to either threshold depends only on their internal Differentiated Services Code Points (DSCPs). This is different than the queue assignment, which only depends on the CoS of the packet. A DSCP to threshold table mapping decides which threshold each of the 64 DSCPs will go to. This table can be seen and modified by issuing the following commands:

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

For example, the command shown below assigns DSCP 23 to threshold 2.

```
NifNif(config−if)#wrr−queue dscp−map 2 26
NifNif#sh mls qos int gig 0/1 queue
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
      1 :    01 01 01 01 01 01 02 01 01 01
      2 :    01 01 01 01 02 01 02 01 01 01
      3 :    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
6 :    01 01 01 01
```

Once the DSCP to threshold map has been defined, WRED is enabled on the queue of your choice. This is done by issuing the following command:

```
(config-if)# wrr-queue random-detect max-thresold Queue_id Threshold_1 Threshold_2
```

The example shown below configures one Q1 with threshold 1 = 50% and threshold 2 = 100%, and Q2 with threshold 1 = 70% and threshold 2 = 100%.

```
3550(config)# interface gigabitethernet0/1
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 verify the type of queuing (WRED or not) on each queue by issuing the following command:

```
nifnif#sh mls qos int gi 0/1 buffers
GigabitEthernet0/1
..
qid WRED thresh1 thresh2
1   dis  10      100
2   dis  10      100
3   ena  10      100
4   dis  100     100
```

`ena` means enable, and the queue is using WRED. `dis` means disable, and the queue is using tail drop.

You can also monitor the number of packets dropped for each threshold by issuing the following command:

```
sh mls qos int gig x/x stat
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 using the same DSCP threshold map defined in the WRED section of this document. When a threshold is reached, all packets with a DSCP assigned to that threshold are dropped. Tail drop thresholds can be configured by issuing the following command:

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

The example below configures Q1 with tail drop threshold 1 = 50% and threshold 2 = 100%, and Q2 with threshold 1 = 70% and threshold 2 = 100%.

```
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. There are however, fixed amounts of packets on a Gigabit port that can be queued. This amount is fixed to 4096. By default, each queue in a Gigabit port can have up to 1024 packets regardless of the packet size. You can however, modify the way these 4096 packet are split among the four queues by issuing the following command :

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

An example is shown below.

```
3550(config)# interface gigabitethernet0/1
3550(config-if)# wrr-queue queue-limit 4 3 2 1
```

These Q size parameters are relative. The example above shows that Q1 will be four times larger than Q4, Q2 three times larger than Q4, and Q3 twice as large as Q4. The 4096 packet are redistributed as follows:

- 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

The following commands allow you to see the relative weights of split buffers among the four queues:

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

You can also see how many free packets each queue can still hold by issuing the following command:

```
(config-if)# sh mls qos int gig x/x stat
 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 currently in the queue. For example, if there are currently 39 packets in Q1 the FreeQ count, 1600 packets are free, as shown below.

```
(config-if)# sh mls qos int gig x/x stat
 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 ports (no WRED or tail drop with two thresholds). All four queues are FIFO queues. There is also no maximum queue size that reserve 4096 packets for each Gigabit port. 10/100 ports store packets in each queue until full due to a lack of resources. You can reserve a minimum amount of packets per queue. This minimum is set to 100 packets per queue by default. You can modify this minimum−reserve value for each queue by defining a different minimum reserve value, and by assigning one of these values to each queue.

This can be done by performing the following steps:

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

   You can configure a maximum of eight different min−reserve values by issuing the following 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, to configure a min−reserve level 1 of 150 packets and a min reserve level 2 of 50 packets, issue the following 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.

   Each of the queues needs to be assigned to one of the min−reserve values to know how many buffers are guaranteed for this queue. By default, the following 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.

   A different min−reserve value can be assigned per queue by issuing the following interface command:

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

   For example, to assign to Q1 a min−reserve of 2, and to Q2 a min−reserve of 1, issue the following command:

```
nifnif(config)#int fas 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
```

The resulting min−reserve assignment can be verified by issuing the following command:

```
nifnif#sh mls qos int fas 0/1 buffers
FastEthernet0/1
Minimum reserve buffer size:
150 50 100 100 100 100 100 100


!--- Showing the value of all eight min-reserve level.

Minimum reserve buffer level select:
2 1 3 4


!--- Showing the min-reserve level assigned to
!--- each queue (from Q1 to Q4).
```

Conclusion

Queuing and scheduling on a port on the 3550 involves performing the following steps:

1. Assigning each CoS to one of the queues.
2. Enabling strict priority queues, if needed.
3. Assigning the WRR weight, taking into account the expected packet size within the queue.
4. Modifying the Q size (Gigabit ports only).
5. Enabling 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. Make sure to adhere to the following guidelines for maximum scheduling performance:

- Classify the traffic present in the network in different classes by either trusting or specific marking.
- Policing traffic in excess.

Related Information

- **Understanding QoS Policing and Marking on the Catalyst 3550**
- **Configuring QoS − Product Documentation**
- **Technical Support − Cisco Systems**

Updated: Oct 17, 2005 Document ID: 24057