



# A Bit More 6500 QoS

Peter J. Welcher and Carole Warner Reece

## Introduction

This article continues our discussion of QoS from last month. There are a few things that just didn't fit into last month's article.

Pete has written several prior articles and presentations about QoS. See his web page at <http://www.netcraftsmen.net/welcher/>. Some previous QoS articles:

QoS in the Campus	<a href="http://www.netcraftsmen.net/welcher/papers/qoscampus.html">http://www.netcraftsmen.net/welcher/papers/qoscampus.html</a>
Configuring Campus QoS	<a href="http://www.netcraftsmen.net/welcher/papers/qoscampuscfg.html">http://www.netcraftsmen.net/welcher/papers/qoscampuscfg.html</a>
New Quality of Service Features in Cisco IOS 12.1	<a href="http://www.netcraftsmen.net/welcher/papers/newqos121.html">http://www.netcraftsmen.net/welcher/papers/newqos121.html</a>
QoS for the Cisco 6500 (Revisited)	<a href="http://www.netcraftsmen.net/welcher/papers/qosfor6500.html">http://www.netcraftsmen.net/welcher/papers/qosfor6500.html</a>

The last of the above articles is last month's article, co-authored with Carole.

## Trust and End-to-End

QoS has to be end to end. That means you really should not neglect QoS anywhere along the traffic path. If you have to prioritize, WAN should come first. But just because campus links are high speed doesn't mean they are free from occasional congestion and dropped packets. And in the campus, you may want to support voice or video mixed in with your data. Traffic averages are just that -- averages -- and do not reflect "micro-bursts" that can saturate queues. For those who have heard me on this subject before, well, I thought it bore repeating.

Example: doing QoS then sending traffic over an IPSec tunnel. You have no control over QoS in the Internet. Sure, you can do what you can within your network, and I won't quite say it is wasted effort. But my gut says this is fine-tuning 1% of the problem and ignoring the other 99%.

Example2: doing QoS in your network and WAN, but using a WAN provider (FR, ATM, Metro Ethernet) with no QoS and no stated Service Level Agreement.

If there is part of your traffic path that is out of your control, e.g. WAN or Metro links, what can you do? You might try characterizing it. The first tool everybody looks at is ping, ICMP, perhaps something like Ping Plotter for small-scale testing. But ICMP is not greatly accurate, as responding may be low priority for the device being pinged. In addition, sites may well rate limit ICMP, meaning you may not get replies when the site is also being pinged by others. Using Cisco IP SLA (former SAA or RTR) functionality provides more sophisticated testing. With CiscoWorks IPM or Brix Networks, Concord, or InfoVista you can even get sophisticated reporting. The key thing here is that these tests can report on lost packets, delay, and jitter, and do so with sensitivity to DSCP (ToS byte) values.

One of the key ideas in QoS is to mark traffic at the Trust Boundary, and then use the markings downstream for simpler classification and specification of per-hop behaviors. One subtle aspect of this is being aware when a midstream device might be remarking the traffic. In particular, if you enable QoS on the 6500 or other switches, the traffic is remarked to 0

(BE, Best Effort) unless you have some trust policy in place on the port or VLAN. And you won't even notice unless you're monitoring the received mix, by examining the number of inbound matches downstream in the network.

Another potential gotcha: if you have a Metro Ethernet service, does it remark your CoS and/or DSCP value while your frames transit the carrier's network? One thought: this might happen if the Ethernet Provider has enabled QoS on their switches, but doesn't trust your traffic because you didn't purchase the QoS service. Pete recently talked to someone whose provider turned on ingress policing per contract, but in doing so inadvertently managed to invoke rate-limiting on multicast traffic, meaning STP, CDP, and OSPF mostly stopped working. One's expectation is that Ethernet service is transparent. But it isn't, particularly if the SP is doing QoS or any form of policing or rate limiting. You might be doing the rate limiting, as the 6500 has some hardware rate-limits enabled by default. (Try the `show mls rate-limit command`.)

Several of us have done QoS work in large networks now. One network mixes Cisco L2 switches, QinQ, optical gear with RPR technology, and MPLS VPN. Each technology has its own approach to QoS with incumbent strengths and weaknesses. When there is a lowest common denominator, typically 8 classes due to the markings available, you have to decide whether to go with that many classes, or to re-mark traffic after transiting the technology with limited markings. (Our inclination on this is that 5 to 8 classes is plenty for most purposes, and you really might not want to have to manage more).

One very useful practice when there are technology mixes like this is to create a QoS diagram, showing a typical end-to-end path within the network. (Or, where there are dual-technology WAN links, show both types of path). This diagram should be annotated to indicate which marking(s) are in use on each link. Indicate the trust boundary and any fine points affecting markings. For example, if you trust inbound CoS on a 6500, you are thereby setting the internal DSCP value. You may also wish to mark up the diagram with other QoS-related information. The point here is not the diagram per se, it is that the diagram is a useful tool for thinking through, documenting, revisiting, and troubleshooting your QoS strategy, especially making sure that the strategy is effective end to end.

## Weighted Round Robin

One of the interesting features in the 6500 series is the ability of the LAN cards to do Weighted Round Robin (WRR) on hardware output queues. Other models like the 3750 do variants resembling this, e.g. Shaped Round Robin (SRR). The 3750 code allows per-port selection of shared mode SRR for WRR-like behavior, or SRR shaping mode, which does per-queue shaping / policing. These features allow us to assign bandwidth to hardware queues. The 6500 Sup-32 uplinks can do SRR as well.

The 6500 series documentation contains a warning that if you change the default settings you should know what you are doing. So please bear that in mind throughout the rest of this section. The following material reflects our interpretation of a feature where the internals are not heavily documented -- so there may be aspects of this we are not aware of.

Now that we've got the warning out of the way, let's go ahead and mess around with the hardware! First you need to enable QoS with the

**mls qos**

global configuration command. Or you won't find anything below having any effect!

The two things you can tune are the WRR queue bandwidth weights and queue depths. Queue depths are just how many frames each queue can hold. I've been thinking of this lately as how much of a burst of traffic can be stored. This is roughly what the Frame Relay Bc and Be parameters measure. The weights are numbers from 1 to 255. Transmission bandwidth is pro-rated between the hardware queues according to the weights.

A concrete example may help with this. Suppose the switch port has 4 queues, and none of them are absolute priority queues. Suppose the weights are assigned according to the following table:

Queue	Weight
1	80
2	60
3	40
4	20

Note the weights sum to 200 (100 would have been too easy). Then queue 1 gets  $80/200 = 40\%$  of the bandwidth, queue 2

gets 60/200 = 30% of the bandwidth, and so on. As with custom-queuing, the switch cannot transmit a fraction of a frame, so we would expect that the observed percentages might vary mildly from the target percentages.

The 6500 ports use either WRR or DWRR (Deficit Weighted Round Robin). See the 6500 documentation for which ports do which form of WRR. Think of DWRR as "rollover bytes" (to borrow a concept from cell phones). That is, if a queue doesn't get to use its byte transmission quota in one round, it gets a credit for the unused bytes the next time round -- a slightly bigger transmission quota. This tends to average out frame size effects and make the actual transmission percentages closer to the configured percentages.

## Translating MQC to 6500 MLS QoS

Pete finds himself thinking about QoS in terms of the MQC commands, since they so nicely express what he is trying to do. Does this happen to you? Or does Pete have QoS on the brain? Here's our point: suppose you wish to allocate bandwidth on a LAN port that does not support the full set MQC (Modular QoS CLI) commands. The "missing" commands might be "priority", "bandwidth", and "shape". What alternatives do we have on the 6500?

If you wish to emulate the MQC LLQ feature (the "priority" command), you may be able to enable or use an absolute priority queue. Note: 6500 LAN ports do not support LLQ. If you enable absolute priority queuing, it generally uses the highest-numbered queue on Cisco switches (except the 3750 which uses queue 1 for this). Usually CoS 5 (DSCP values 40-47) are mapped to this queue by default.

```
PE12#sh queueing int gig 1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin
Port QoS is enabled
Port is untrusted
Extend trust state: not trusted [COS = 0]
Default COS is 0
Queueing Mode In Tx direction: mode-cos
Transmit queues [type = lp3q8t]:
Queue Id      Scheduling  Num of thresholds
-----
      01          WRR             08
      02          WRR             08
      03          WRR             08
      04      Priority             01

WRR bandwidth ratios:  100[queue 1] 150[queue 2] 200[queue 3]
queue-limit ratios:    50[queue 1]  20[queue 2]  15[queue 3]
<snip>
```

So all your voice or other DSCP EF traffic uses the priority queue. Note that this is not quite the same behavior as with MQC however, because the **priority** command polices the LLQ queue to make sure other classes of traffic are not starved. You could add policing to the 6500 configuration if you were concerned about this, and wished to take protective (defensive) measures. Or you could trust the VoIP or IPT Call Admission Control (which is needed for VoIP to work well in conjunction with QoS anyway).

You can also adjust what CoS is mapped to the priority queue with the

```
priority-queue cos-map 1 cos1 [... cosn]
```

interface configuration command.

For example,

```
PE12(config-if)# priority-queue cos-map 1 5 6 7
Propagating cos-map configuration to:  Gi1/1 Gi1/2 Gi1/3 Gi1/4 Gi1/5 Gi1/6 Gi1/7
Gi1/8 Gi1/9 Gi1/10 Gi1/11 Gi1/12
PE12(config-if)#
```

Note that this command is applied to all ports supported on the ASIC.

Suppose you want to emulate the bandwidth command on the 6500. You can't do it per class on a LAN port. But you can look at the class to queue mappings, and you can use the wrq-queue bandwidth and the rcv-queue bandwidth commands to

allocate minimum bandwidth for the transmit and and receive queues, very similar to the way the MQC bandwidth command works.

Suppose you have emulated the **priority** command as above using either the default or your manually configured CoS to priority queue mappings. And you know that VoIP traffic is using up to say 10% of the bandwidth on some port. Say that leaves you 3 egress queues you can divide up the other 90% of bandwidth on, based on the LAN card and Sup engine you are using. Say you configure

```
wrr-queue bandwidth 60 30 10
```

for a port with 1 priority and 3 standard transmit queues. (The **wrr-queue bandwidth** command will allocate bandwidth between the standard transmit queues.)

The 60 30 10 weights happen to add up to 100 by pre-arrangement, so queue 1 is getting 60% of the available bandwidth, queue 2 30%, and queue 3 10%. Ah, but only 90% is available for these non-priority queues to divvy up. So queue 1 gets 60% of 90% of the link bandwidth, or 54%. Queue 2 gets 30% of 90%, or 27% of the link bandwidth. And queue 3 gets 10% of 90% or 9%. The end effect is queue 1 gets 54%, queue 2 27%, queue 3 9%, and priority queue 10% of the link bandwidth.

That shows how to take a WRR configuration and figure out what it's doing for you. Usually we want to work this the other way around, and take some MQC QoS plan and convert it to WRR weights. Let's work through an example. Say your MQC plan is shown in the following table:

Class	DSCP Value	Cos Value	Percent
Voice	EF	5	5
Class Realtime (videoconferencing)	AF41 and AF43	4	10
Priority Data	AF31	3	20
Bulk Data	AF21	2	20
Business Data	AF11	1	20
Best Effort	0	0	25

Suppose your CoS maps put Cos 0 and 1 into queue 1, CoS 2 and 3 into queue 2, CoS 5 into the priority queue, and the rest into queue 3.

You can also adjust what CoS is mapped to the standard transmit queues with the

```
wrr-queue cos-map queue-id threshold-id cos1 [... cosn]
```

interface command. Note that this command is applied to all ports supported on the ASIC.

```
PE12(config-if)#wrr-queue cos-map 3 2 2 4
Propagating cos-map configuration to: Gi1/1 Gi1/2 Gi1/3 Gi1/4 Gi1/5 Gi1/6 Gi1/7
Gi1/8 Gi1/9 Gi1/10 Gi1/11 Gi1/12
PE12(config-if)#
```

(We're ignoring queue thresholds for now, which are really more about selective dropping behavior as the queue fills). Then the above table translates into the following bandwidth percentage needs, adding percentages where multiple classes go into the same queue:

Queue	CoS Values	Percent
Priority	5	5
3	4, 6, 7	10
2	2, 3	20+20 = 40
1	0, 1	25+20 = 45

What weights should we use? We're planning that the priority queue will consume 5% off the top. Say we're even going to

police the traffic to cap it at 5%. Note that you have to do the math to figure out how many bits per second 5% is, and perhaps round that a little. The 6500 policing doesn't accept configuration with percentages.

Let's figure out queue 3 first. Let's use weights that add up to 100, since that's simplest. We want weight W chosen so that  $W / 100 \times 95 = 10$ . Then  $W = 10/95 \times 100$  or about 11. Doing the same for queue 2, we get  $40/95 \times 100 = 42$ . For queue 3,  $45/95 \times 100 = 47$ . Check:  $11 + 42 + 47 = 100$ . (If the rounding off didn't lead to numbers adding to 100, either leave them as is or adjust some them up or down by 1 until they do add to 100). So we'd configure WRR weights 11, 42, and 47.

## Trusting CoS

Pete has a big hangup with trust. As far as inbound CoS that is! If you trust CoS inbound, historically that sets the internal DSCP value per the CoS to DSCP map, rather than from the DSCP bits you so laboriously set at the trust boundary. If it weren't for that issue, Pete would love to have the benefits of differentiated behavior in inbound queues.

Carole supplied a pair of answers to this concern. First, DSCP Transparency uses CoS for queueing, and allows CoS remarking without altering the DSCP value, and while leaving the PFC active. The `no mls qos rewrite ip dscp` command is how you configure this. See <http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/qos.html> for details. A second partial answer is the `mls qos queueing-only` command, which disables PFC classification and marking. All queueing is based on inbound CoS. This keeps DSCP intact but is a bit drastic (roughly equivalent to disabling major parts of QoS).

## Monitoring QoS

Another question we've been debating internally is monitoring ASIC QoS. Ideally one could keep an eye on queue depths, number of packets processed for each DSCP level, drops, etc. The 3550 supports a command such as `mls qos monitor dscp 8 16 24 32`. You can subsequently do the command `show mls qos interface g0/3 statistics` and get a listing of statistics for each of those DSCP values observed on the interface, inbound and outbound. We omit sample output since it is in the copyrighted Cisco documentation, see in particular the link [http://www.cisco.com/en/US/products/hw/switches/ps646/products\\_tech\\_note09186a00800feff5.shtml](http://www.cisco.com/en/US/products/hw/switches/ps646/products_tech_note09186a00800feff5.shtml).

The following sample shows what you can get from a 6500:

```
P5#sh mls qos ip
QoS Summary [IP]:          (* - shared aggregates, Mod - switch module)

  Int Mod Dir  Class-map DSCP  Agg  Trust Fl  AgForward-By  AgPoliced-By
  -----
  PO1/1  5  In   cust-A   32    2   No  0           0           0
  PO1/1  5  In   cust-B   24    1   No  0           0           0
  GE4/2  5  In   IPP-5   40    6   No  0       2891812       0
  GE4/2  5  In   IPP-4   32    5   No  0       4339092       0
  GE4/2  5  In   IPP-3   24    7   No  0       8676810       0
  GE4/2  5  In   IPP-2   16    8   No  0       4338634       0
  GE4/2  5  In   IPP-1    8    9   No  0       8676352       0
  GE4/2  5  In   IPP-0    0   10   No  0       17353620      0
  GE4/4  5  In   cust-A   32    4   No  0           0           0
  GE4/4  5  In   cust-B   24    3   No  0           0           0
  All    5  -    Default  0     0*  No  0       442570626327  0
```

## Summary

Your comments, questions, and suggestions for future articles are of course welcome! See below to decipher Pete or Carole's email address.

---

Dr. Peter J. Welcher (CCIE #1773, CCSI #94014, CCIP) is a Senior Consultant with Chesapeake NetCraftsmen. NetCraftsmen is a high-end consulting firm and Cisco Premier Partner dedicated to quality consulting and knowledge transfer. NetCraftsmen has ten CCIE's, with expertise including large network high-availability routing/switching and design, VoIP, QoS, MPLS, IPSec VPN, wireless LAN and bridging, network management, security, IP multicast, and other areas.

See <http://www.netcraftsmen.net> for more information about NetCraftsmen. Pete's links start at <http://www.netcraftsmen.net/welcher>. New articles will be posted under the Articles link. Questions, suggestions for articles, etc. can be sent to [pjw@netcraftsmen.net](mailto:pjw@netcraftsmen.net) (formatted this way to fool email harvesting software).

Carole Warner Reece (CCIE #5168) is also a Senior Consultant with Chesapeake NetCraftsmen. She has worked with a wide variety of products and technologies in complex environments, and has broad experience in network design, consulting, and project implementation, as well as in developing network training materials. She can be reached at [cwr@netcraftsmen.net](mailto:cwr@netcraftsmen.net).

12/10/2005. Updated 9/18/2008.

Copyright (C) 2005 Peter J. Welcher