Cisco Systems

White Paper

Version 4.04

Understanding Quality of Service on the Catalyst 6500 and Cisco 7600 Router

Carl Solder

Technical Marketing Engineer
Internetwork Systems Business Unit

June 2006
Version 4.04

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED ‘‘AS IS’’ WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents

Table of Figures

Tables

Terminology

Aggregate	A type of Policer (see Policing) applied to **ALL** traffic matching a particular traffic class
ASIC	Application Specific Integrated Circuit is a purpose built chipset designed to perform a specific function in hardware
CatOS	Catalyst Operating System is one of two operating system options available for the Catalyst 6500 platform. Originally derived from the Catalyst 5000 code base.
Cisco IOS	Cisco IOS is the second of two Operating Systems available for the Cat6500. Code Base derived from router IOS and provides single OS image for L2 and L3 subsystems and single configuration file.
CEF	Cisco Express Forwarding is a technique whereby a forwarding information base is computed from the routing tables and used by the PFC to switch all L3 traffic
Classification	Process of identifying a frame based on specific Layer 2, 3 and 4 fields in the header
CoS	Class of Service – This is a Cisco term used to refer to the 802.1p specification – if refers to the 3 bits in an Ethernet header that indicate the priority of an Ethernet frame
dCEF	Distributed version of Cisco Express Forwarding used by the DFC or DFC3x in combination with the PFC2 or PFC3x
DFC	Distributed Forwarding Card is a combo daughter card comprising a MSFC and PFC used by a fabric enabled Cat6500 line card to perform distributed switching
DSCP	Differentiated Services Code Point is the six most significant bits in the ToS byte, and is used to indicate the priority of an IP Packet. Not to be confused with the internal DSCP (see below).
Frame	Term used to define a piece of data with a layer 2 header. An example would be an Ethernet frame.
Flow	A unidirectional flow of traffic from a source host to a destination host. Most TCP/IP connections use 2 flows, one for sending data and one for receiving data.
Internal DSCP	An internal priority assigned to the frame by the PFC as it transits the switch. It is derived from an existing CoS or ToS setting and is used to reset CoS or ToS as the frame exits the switch.
IP Precedence	The three most significant bits in the ToS field in the IPv4 header used to indicate the priority of an IP Packet
Microflow	A type of Policer (see policing) applied on a per flow-basis for traffic coming into a port or a VLAN
MSFC	Multilayer Switch Feature Card is the Layer 3 switching engine that sites on the Catalyst Supervisor as a daughter card.
MSFC1	Multilayer Switch Feature Card found on a Supervisor 1 or Supervisor 1A
MSFC2	Multilayer Switch Feature Card found on a Supervisor 1A or Supervisor 2
MSFC3	Multilayer Switch Feature Card found on all Supervisor 720 models
Packet	Term used to define a piece of data with a layer 3 header. An example would be an IP packet.
PFC	Policy Feature Card is a daughter card that sits on the Supervisor providing hardware accelerated L3/L4 switching, QoS and Security
PFC1	Policy Feature Card supported on a Supervisor 1 or Supervisor 1A

PFC2	Policy Feature Card supported on a Supervisor 2
PFC3a	Policy Feature Card supported on a Supervisor 720
PFC3B	Policy Feature Card supported on a Supervisor 720-3B
PFC3BXL	Policy Feature Card supported on a Supervisor 720-3BXL
Policing	Process of ensuring identified traffic flow is limited to a predefined flow rate; excess traffic above this level is either dropped or has its QoS value marked down.
QoS	Quality of Service
TCAM	Specialized memory (Table) for storing ACL's used on the PFC1, PFC2, and PFC3 as well as the DFC and DFC3.
Threshold	A utilization level in a buffer used by a congestion management algorithm to determine when to start dropping identified frames
ToS	Type of Service field is a one-byte field in the IP V4 header used in part to denote the priority of that IP packet.
VoIP	Voice over IP
WRED	Weighted Random Early Discard – a buffer management algorithm that uses CoS bits to identify which frames to drop at times of congestion
WRR	Weighted Round Robin is a scheduling algorithm that uses weights assigned to queues to determine how much data will be emptied from a queue before moving to the next queue.

ABSTRACT

This document explains the Quality of Service (QoS) capabilities available in the Catalyst 6500 and Cisco 7600 Router as it applies to PFC QoS for IPv4 data. This document does not cover QoS features on the Cisco 7600 WAN linecards like the FlexWAN, OSM and SIP modules, nor does it cover QoS for MPLS. This document covers QoS configuration capabilities and provides some examples of QoS implementation. It focuses on shipping hardware as of the date of this publication.

This paper is not meant to be a configuration guide. Configuration examples are used throughout this paper to assist in the explanation of QoS features of the Catalyst 6500 and Cisco 7600 hardware and software. This paper uses Cisco IOS configuration examples when required. For syntax reference for QoS command structures, please refer to the configuration and command guides for the Catalyst 6500 on Cisco Connection Online at http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/index.htm

1. What is Layer 2 and Layer 3 QoS

While some may think of QoS in a switch is only about prioritising Ethernet frames, it is in fact much more. Layer 2 and Layer 3 QoS in the Catalyst 6500 entails the following:

1. **Input Trust and Queue Scheduling** – When the frame enters the switch port, it can be assigned to an ingress port-based queue prior to being switched to an egress port. Different queues can be used for different traffic that require different service levels, or where switch latency must be kept to a minimum. For instance, IP based video and voice data requires low latency, so there may be a need to switch this data prior to switching other data like FTP, web, email, telnet, etc. The Trust element of QoS determines whether to honor any existing priority settings incorporated in the incoming frame.
2. **Classification** – the process of classification is one of inspecting different fields in the Ethernet Layer 2 header, along with fields in the IP header (Layer 3) and the TCP/UDP header (Layer 4) to determine the level of service that should be applied to the frame as it transits the switch.
3. **Policing** – Policing is the process of inspecting whether traffic on a given port or within a VLAN has exceeded a pre defined rate. Should that traffic be out of profile (i.e. the traffic stream is in excess of the pre defined rate limit), excess data can be either dropped or its priority value can be marked down.
4. **Marking** – The process of rewriting is the ability of the switch to modify the priority of the packet via the CoS in the Ethernet header or the Type of Service bits in the IPv4 header
5. **Output Queue Congestion Management and Scheduling** – When the packet is ready to be switched to its next hop destination, the switch will place the Ethernet frame into an appropriate outbound (egress) queue for switching. The switch will perform buffer (congestion) management on this queue by monitoring the utilization. Congestion management utilizes a Random Early Discard (RED) algorithm whereby random frames are refused admission to the queue once a threshold has been exceeded. Weighted RED (WRED) is a derivative of RED whereby the frames priority values are inspected to determine which frames will be dropped. When the buffers reach set thresholds, then (typically) lower priority frames are dropped allowing the higher priority frames to enter the queue. Scheduling of the packet is performed by a round robin (and possibly strict priority) scheduling technique that moves between the queues to forward the packets.

This paper will explain in more detail each of the mechanisms above and how they relate to the Catalyst 6500.

2. Why The Need for QoS in a Switch

References to large capacity switching backplanes, millions of switched packets per second and non-blocking switches are synonymous terms used with many switches today. Why then do we need QoS? The answer is simply because of congestion.

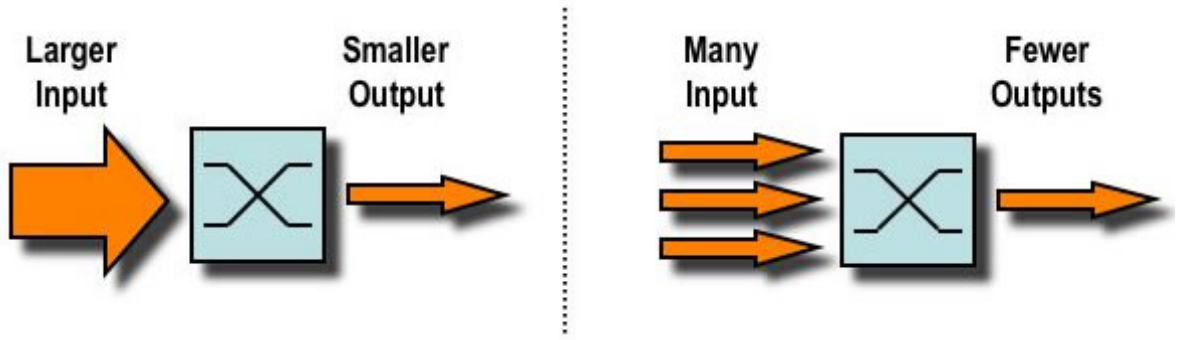

Figure 1 – Congestion causes in switches

A switch may be the fastest and largest non-blocking switch in the world, but if you have either of the two scenarios shown in the above figure (as in just about every network in the world), then that switch can experience congestion. At times of congestion, packets will be dropped if the congestion management features are not up to par. When packets are dropped, retransmissions occur. When retransmissions occur, the network load can increase. In already congested networks this can add to existing performance issues and potentially further degrade overall network performance.

With converging networks, congestion management is even more critical. Latency sensitive traffic like voice and video can be severely impacted if delays are incurred. Simply **adding more buffers to a switch** will also **not necessarily alleviate congestion** problems either. Latency sensitive traffic needs to be switched as fast as possible. First we need to identify this important traffic through classification techniques. Then we need to implement buffer management techniques to avoid the higher priority traffic from being dropped during congestion. Finally we need to incorporate scheduling techniques to forward important packets from queues as quickly as possible. As you will read in the document, the Catalyst 6500 implements all of these techniques making its QoS subsystem one of the most comprehensive in the industry today.

All of the QoS techniques described in the previous section will be explored in more detail throughout this paper.

3. Hardware Support for QOS in the Catalyst 6500

QoS support in the Catalyst 6500 is performed by Application Specific Integrated Circuits (ASIC's) in hardware. This is one of the major differentiators between the QoS capabilities on this platform and QoS capabilities on other Cisco Router platforms where those router platforms execute most QoS functionality in software. The specific hardware supporting QoS includes the Multi-layer Switch Feature Card (MSFC), the Policy Feature Card (PFC) and various port ASIC's on the line cards. The focus of this document will be on the QoS capabilities of the PFC and ASIC's on the line cards.

3.1 Policy Feature Card (PFC)

There are five versions of the Policy Feature Card in use today. The PFC1 is an optional daughter card for the Supervisor 1 and Supervisor 1A, the PFC2 is an integrated daughter card found on the Supervisor 2, the PFC3a is integrated into the Supervisor 720 and the PFC3B, and PFC3BXL are integrated into the Supervisor 720-3B and Supervisor 720-3BXL respectively.

There is a set of base QoS functions common to all versions of the PFC. Each generation of PFC supports the following:

- Traffic policing using a single leaky bucket algorithm
- Traffic classification of ingress traffic using Access Control Lists to inspect information in the Layer 2, 3 and 4 headers
- Re-marking of DSCP priority bits
- Ability to trust incoming Class of Service and Type of Service priority bits

The PFC2 is a 2nd generation forwarding engine and adds the following features over the PFC1:

- The ability to push down local QoS policies to a Distributed Forwarding Card (DFC)
- Support for dual leaky bucket algorithm when policing traffic.
- A dedicated QoS TCAM (separate from the TCAM used to store Security ACL's) containing 32K entries and 4K masks

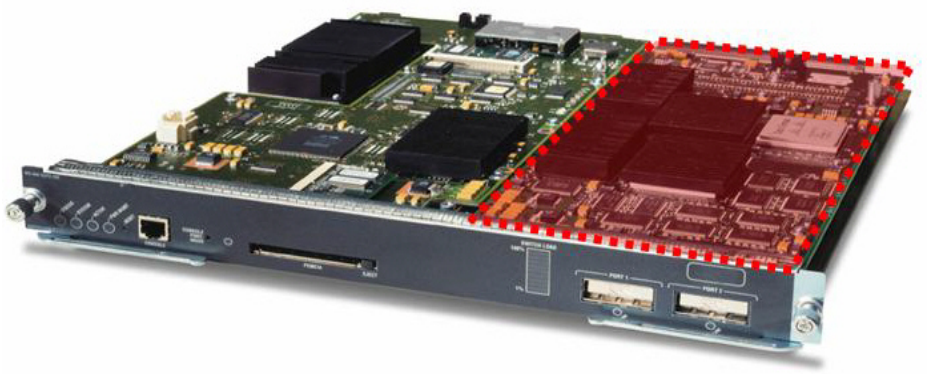

Figure 2 - PFC2 on the Supervisor 2

The PFC3a adds the following QoS features over the PFC2

- Egress Policing is now supported for aggregate policing on both routed ports and VLAN switched virtual interface (SVI) ports
- Egress marking and egress classification
- User Based Rate Limiting (UBRL)
- DSCP Transparency

Figure 3 - Policy Feature Card 3 on the Supervisor 720

Figure 4 - Policy Feature Card 3 (PFC3)

Both the PFC3B and PFC3BXL add the following QoS features over the PFC3a

- Apply QoS policies on Tunnel interfaces
- Apply Layer 2 ACL's for IPv4 traffic (supported on a per VLAN basis)
- Ability to match on CoS or VLAN within a class map

All versions of the PFC support the processing of QoS in hardware. This means that forwarding rates are not impacted when QoS is enabled in the switch. The PFC1 can support centralised forwarding rates up to 15Mpps, while the PFC2 and all versions of the PFC3 can support centralised forwarding rates up to 30Mpps.

3.2 Distributed Forwarding Card (DFC)

The Catalyst 6500 architecture supports the use of Distributed Forwarding Cards (DFC) on CEF256, dCEF256, and CEF720 based linecards. A DFC is used to hold a local copy of the forwarding tables along with Security and QoS policies to facilitate local switching on the linecard. The original DFC is supported in CEF256 and dCEF256 based linecards when used with the Sup2. A CEF256 linecards support an optional DFC while dCEF256 linecards support an integrated DFC. The DFC3a is available as

an option on CEF256 and CEF720 based linecards. More recently, the DFC3B and DFC3BXL were introduced for linecards to operate with the new PFC3B and PFC3BXL.

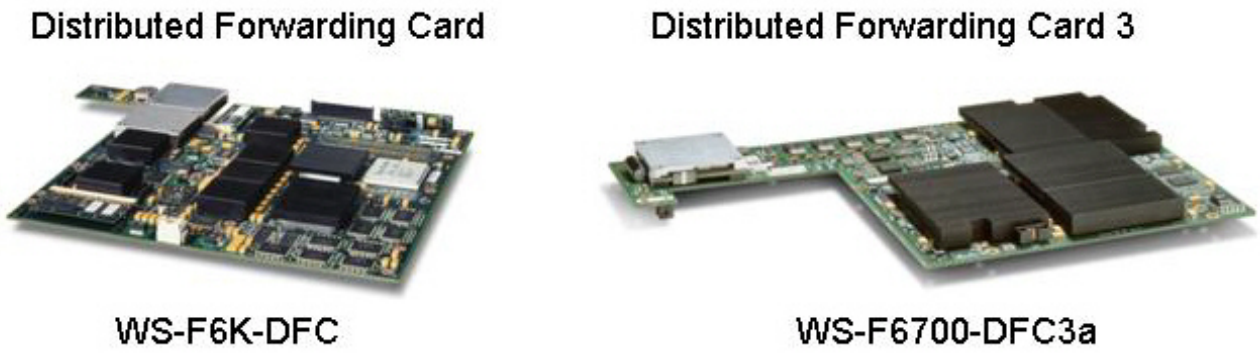

Table 1 - DFC and the DFC3a

It is important to note that there are some operational considerations that can impact the ability of the Catalyst 6500 system to provide specific QoS features. This can happen when you mix different generations of PFC's and DFC's together. The rule is that the system will operate at the lowest common feature denominator. The table below sets out the operational level of the system based on a mix of DFC and PFC interaction.

	PFC2	PFC3a	PFC3B	PFC3BXL
DFC	Normal Operation	Combination not allowed	Combination not allowed	Combination not allowed
DFC3a	Combination not allowed	Normal Operation	PFC3B operates as a PFC3a	PFC3BXL operates as a PFC3a
DFC3B	Combination not allowed	DFC3B operates as a DFC3a	Normal Operation	PFC3BXL operates as a PFC3B
DFC3BXL	Combination not allowed	DFC3BXL operates as a DFC3a	DFC3BXL operates as a DFC3B	Normal Operation

Table 2 - PFC/DFC Interaction Matrix

All DFC options allow fabric (crossbar connected) line cards to perform local switching. In order facilitate local switching, QoS policies initially programmed into the PFC are pushed down and stored on the DFC/DFC3x. This process negates the need to access QoS policies on the PFC when a local switching operation is in progress. Storing the QoS policies on the DFC allows maximum local switching performance speeds to be maintained. The administrator cannot directly configure the DFC or DFC3x; rather, the master MSFC/PFC on the active supervisor controls and programs the DFC. The original DFC will only work with the PFC2 on the Supervisor 2, while the new DFC3x will work with the PFC3x on the Supervisor 720.

The primary MSFC (MSFC2 or MSFC3) will calculate, then push down a FIB table (Forwarding Information Base) giving the DFC3x its layer 3 forwarding tables. The MSFC will also push down a copy of the QoS policies so that they are also local to the line card. Subsequent to this, local switching decisions can reference the local copy of any QoS policies providing hardware QoS processing speeds and yielding higher levels of performance though distributed switching.

3.3 Linecard Port ASIC Queue Structure

The final aspect of QoS support in the Catalyst 6500 is implemented by ASIC's (Application Specific Integrated Circuits) on the linecard itself. Those ASIC's implement the queues, buffering and threshold management used at ingress and egress for the temporary storage of frames as they transit the switch.

Each of the ports on the linecards detailed below uses one of a number of specific queue structures. The queue structures use nomenclature to indicate how many normal and strict priority queues are supported along with how many thresholds per queue are available. The keywords in the queue structure type include the following:

- "Q" – indicates the number of normal queues in the queue structure
- "P" – indicates the number of Strict Priority (Low Latency) queues in the queue structure
- "T" – indicates the number of thresholds per normal queue – the Strict Priority does not implement any form of threshold.

Each of the queue structures in use on linecards today are listed in the table below and are further referenced in each of the linecard sections that follow.

Queue Structure	Receive or Transmit Queue Type	Support Strict Priority Queue	Number of Queues	Number of Thresholds per Queue
1Q2T	Receive	No	1	2
1Q4T	Receive	No	1	4
1Q8T	Receive	No	1	8
1P1Q0T	Receive	Yes	2	0
1P1Q4T	Receive	Yes	2	4
1P1Q8T	Receive	Yes	2	8
2Q8T	Receive	No	2	8
8Q8T	Receive	No	8	8
1P2Q1T	Transmit	Yes	3	1
1P2Q2T	Transmit	Yes	3	2
1P3Q1T	Transmit	Yes	4	1
1P3Q8T	Transmit	Yes	4	8
1P7Q8T	Transmit	Yes	8	8
2Q2T	Transmit	No	2	2

Figure 5 - Linecard Queue Structures

Each of the major linecard groups are detailed below with a summary of the QoS features available for that linecard.

3.3.1. 10/100 Line Cards (WS-X6148, WS-X6324 and WS-X6348 Series)

These linecards are classic based linecards (support a 32Gb bus connection only). Linecards included in this group include the following:

- WS-X6148 group includes WS-X6148-RJ45, WS-X6148-RJ45V, WS-X6148-RJ21, WS-X6148-RJ21V, WS-X6148-45AF, WS-X6148-21AF

- WS-X6348 group includes WS-X6348-RJ45, WS-X6348-RJ45V, WS-X6348-RJ21V
- WS-X6324 group includes WS-X6324-100FX-MM, WS-X6324-100FX-SM

Details for these linecards are listed in the following table.

	WS-X6148	WS-X6324	WS-X6348
Number of Ports	48	24	48
# Port ASIC's per linecard	4	2	4
# Physical Ports per Port ASIC	12	12	12
Receive Queue Per Port	Yes	Yes	Yes
Transmit Queue Per Port	Yes	Yes	Yes
Per Port Buffering	Yes	Yes	Yes
Buffer available per port	128K	128K	128K
Buffer on Receive side	16K	16K	16K
Buffer on Transmit Side	112K	112K	112K
Transmit Queue Structure per port	2Q2T	2Q2T	2Q2T
Receive Queue Structure per port	1Q2T	1Q2T	1Q2T
Receive (RX) Strict Priority Queue	No	No	No
Transmit (TX) Strict Priority Queue	No	No	No

Table 3 – QoS on 10/100 Line Cards (WS-X6148, WS-X6324 and WS-X6348 Series)

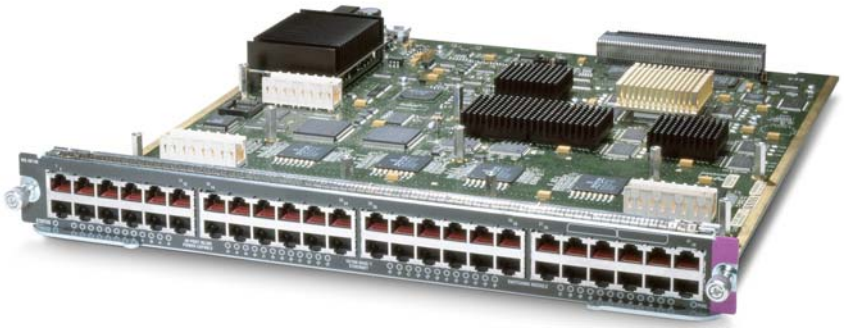

Figure 6 - WS-X6148-RJ45 Linecard

3.3.2. 10/100 Line Cards (WS-X6148A Series)

Linecards in this group include the following

- WS-X6148A-RJ45
- WS-X6148-FE-SFP (100FX Linecard)

	WS-X6148A-RJ45	WS-X6148-FE-SFP
Number of 10/100 Ports	48	-
Number of 100FX ports	-	48
# Port ASIC's on the linecard	6	6
# Physical Ports per Port ASIC	8	8
Per Port Buffering	Yes	Yes

Buffer on Receive Side	64K	64K
Buffer on Transmit Side	5.2MB	5.2MB
Transmit Queue Structure per port	1P3Q8T	1P3Q8T
Receive Queue Structure per port	1P1Q4T	1P1Q4T
Receive (RX) Strict Priority Queue	Yes	Yes
Transmit (TX) Strict Priority Queue	Yes	Yes

Table 4 - QoS on 10/100 Line Cards (WS-X6148A Series)

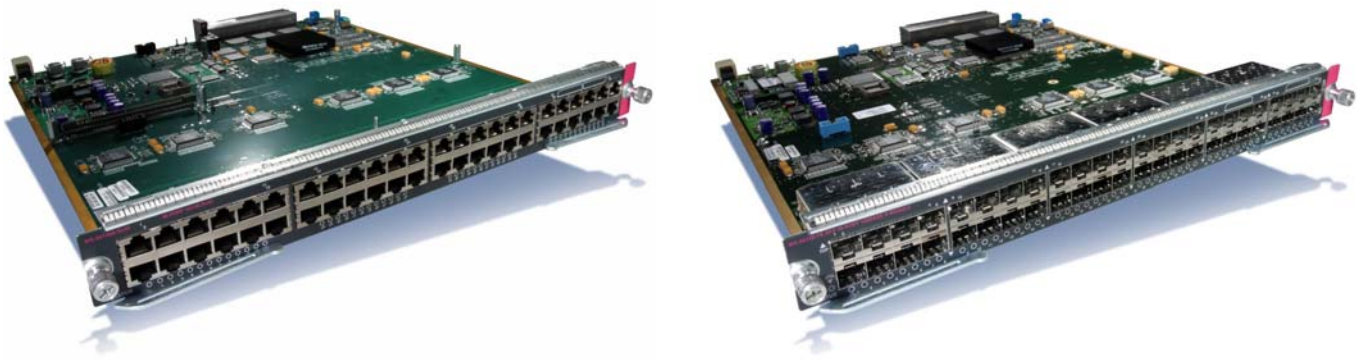

Figure 7 - WS-X6148A-RJ45 and WS-X6148-SFP Linecard

3.3.3. 10/100 Line Cards (WS-X6148X2, WS-X6196-RJ21)

This group includes the following linecards:

- WS-X6148X2-RJ45
- WS-X6196-RJ21

The WS-X6148X2-RJ45 linecard is unique in the Catalyst 6500 family line-up due to it having 48 physical ports on the linecard, but with the addition of a separate patch panel can be set-up to provide 96 ports of 10/100. The WS-X6196-RJ21 provides the full 96 ports via a set of front panel RJ-21 interfaces.

	WS-X6148X2-RJ45	WS-X6196-RJ21
Number of 10/100 Ports	96	96
Port Type	RJ-45	RJ-21
# Port ASIC's per linecard	2	2
# Physical Ports per Port ASIC	48	48
Per Port Buffering	Yes	Yes
Buffer on Receive Side	28K	28K
Buffer on Transmit Side	1.1MB	1.1MB
Transmit Queue Structure per port	1P3Q1T	1P3Q1T
Receive Queue Structure per port	1P1Q0T	1P1Q0T

Table 5 - QoS on 10/100 Line Cards (WS-X6148X2, WS-X6196-RJ21)

Both of these modules are pictured below.

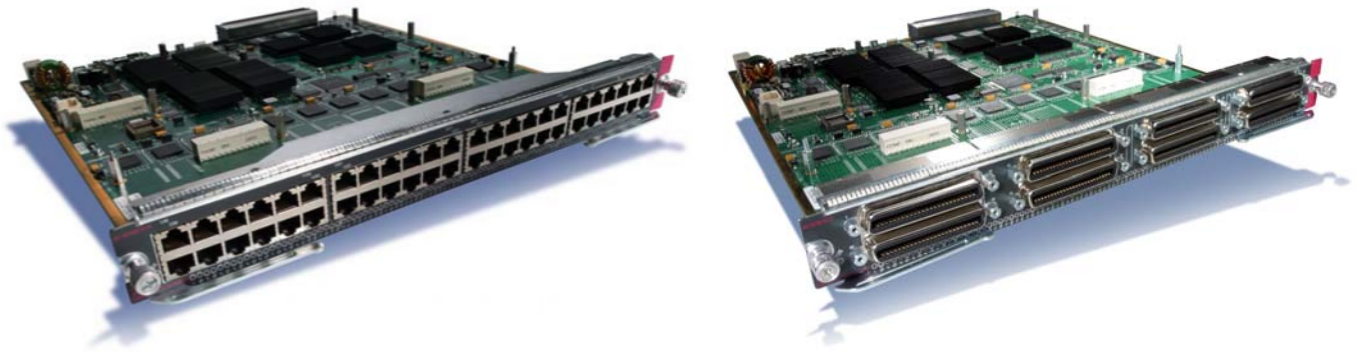

Figure 8 - WS-X6148X2-RJ45 and WS-X6196-RJ21 Linecards

3.3.4. 10/100 Line Cards (WS-X6524, WS-X6548)

This group includes the following linecards:

- WS-X6524-100FX-MM
- WS-X6548-RJ-45
- WS-X6548-RJ-21

These 10/100 Ethernet linecards utilize a different port ASIC implementation to the one used in the WS-X6148 and WS-X6348 series linecards mentioned above. Both queue structure and port buffering has been enhanced significantly and use the queue structures 1P1Q4T and 1P3Q1T. More importantly a strict priority queue now exists on both the ingress and egress side.

	WS-X6524	WS-X6548
Number of 10/100 Ports	-	48
Number of 100FX ports	24	-
Port Type	100FX	RJ-45
# Port ASIC's per linecard	1	1
# Physical Ports per Port ASIC	24	48
Per Port Buffering	Yes	Yes
Buffer on Receive Side	28K	28K
Buffer on Transmit Side	1.1MB	1.1MB
Transmit Queue Structure per port	1P3Q1T	1P3Q1T
Receive Queue Structure per port	1P1Q4T	1P1Q4T

Table 6 - QoS on 10/100 Line Cards (WS-X6524, WS-X6548)

Table 7 - WS-X6548-RJ45 Linecard

3.3.5. Gigabit Ethernet Line Cards (WS-X6408, WS-X6408A)

Linecards in this group include the following

- WS-X6408-GBIC
- WS-X6408A-GBIC

	WS-X6408	WS-X6408A
Number of GE Ports	8	8
# Port ASIC's on the linecard	2	2
# Physical Ports per Port ASIC	4	4
Per Port Buffering	Yes	Yes
Buffer on Receive Side	73K	73K
Buffer on Transmit Side	439K	439K
Transmit Queue Structure per port	2Q2T	1P2Q2T
Receive Queue Structure per port	1Q4T	1P1Q4T
Receive (RX) Strict Priority Queue	No	Yes
Transmit (TX) Strict Priority Queue	No	Yes

Table 8 - QoS on Gigabit Ethernet Line Cards (WS-X6408, WS-X6408A)

3.3.6. Gigabit Ethernet Line Cards (WS-X6316, WS-X6416, WS-X6516, WS-X6816)

Linecards in this group include the following

- WS-X6316-GBIC
- WS-X6416-GBIC
- WS-X6516-GBIC
- WS-X6816-GBIC

Details on each of the linecard groups are detailed in the following table.

	WS-X6316	WS-X6416	WS-X6516	WS-X6816
Number of GE Ports	16	16	16	16
Port Type	GBIC	GBIC	GBIC	GBIC
# Port ASIC's per linecard	4	4	4	4
# Physical Ports per Port ASIC	4	4	4	4

Per Port Buffering	Yes	Yes	Yes	Yes
Buffer on Receive Side	73K	73K	73K	73K
Buffer on Transmit Side	439K	439K	439K	439K
Transmit Queue Structure per port	1P2Q2T	1P2Q2T	1P2Q2T	1P2Q2T
Receive Queue Structure per port	1P1Q4T	1P1Q4T	1P1Q4T	1P1Q4T
Receive Strict Priority Queue	Yes	Yes	Yes	Yes
Transmit Strict Priority Queue	Yes	Yes	Yes	Yes

Table 9 - QoS on Gigabit Ethernet Line Cards (WS-X6316, WS-X6416, WS-X6516, WS-X6816)

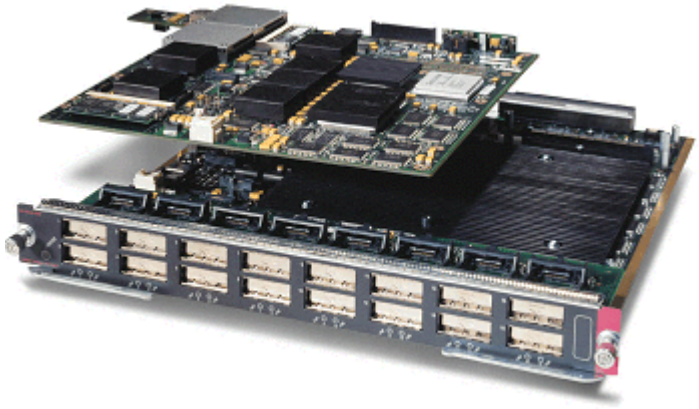

Table 10 - WS-X6516-GBIC with optional DFC

3.3.7. Gigabit Ethernet Line Cards (WS-X6516a)

The WS-X6516a-GBIC is almost identical to the WS-X6516-GBIC linecard, except, from a QoS perspective, it supports a later version of the GE ASIC. It supports the same queue structures of 1P1Q4T and 1P2Q2T. This new GE ASIC provides more port buffering than that provided on the WS-X6516-GBIC. Now, a full 1Mb of port buffering is available to each GE port on the linecard (946KB on the transmit side and 135KB on the receive side). Aside from the increased buffering, all other QoS capabilities are the same as the earlier WS-X6516-GBIC linecard version.

3.3.8. Gigabit Ethernet Line Cards (WS-X6148-GE-TX, WS-X6548-GE-TX)

Linecards in this group include the following

- WS-X6148-GE-TX
- WS-X6548-GE-TX

	WS-X6148	WS-X6548
Number of GE Ports	48	48
# Port ASIC's on the linecard	6	6
# Physical Ports per Port ASIC	8	8
Per Port Buffering	Yes	Yes
Buffer on Receive Side	185KB between 8 ports	185KB between 8 ports

Buffer on Transmit Side	1.2MB between 8 ports	1.2MB between 8 ports
Transmit Queue Structure per port	1P2Q2T	1P2Q2T
Receive Queue Structure per port	1Q2T	1Q2T
Receive (RX) Strict Priority Queue	No	No
Transmit (TX) Strict Priority Queue	Yes	Yes

Table 11 - QoS on Gigabit Ethernet Line Cards (WS-X6148-GE-TX, WS-X6548-GE-TX)

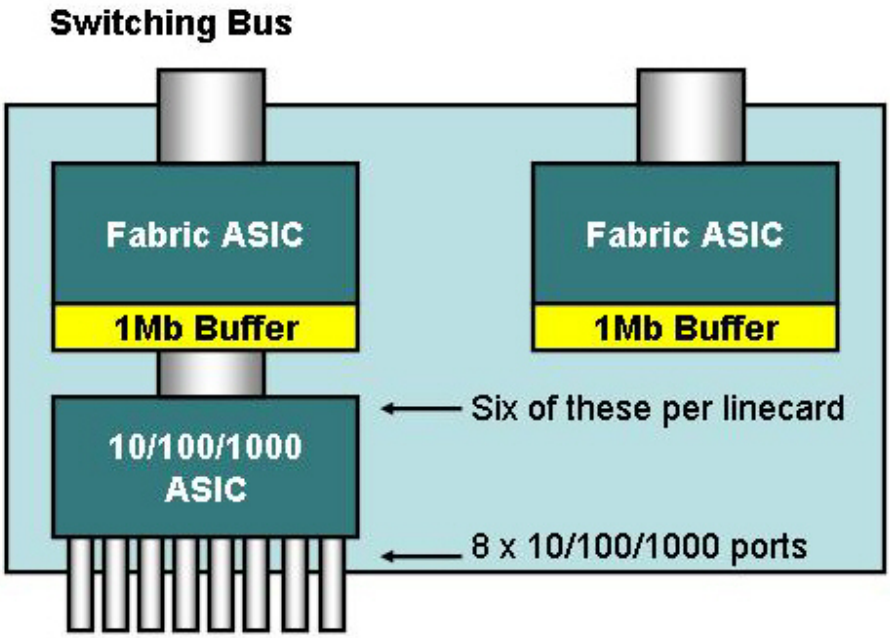

Figure 9 - WS-X6148-GETX and WS-X6548-GETX architecture

Buffering for these linecards is implemented on a shared basis. There is 1.2Mb of buffering available for each 10/100/1000 ASIC. There are six 10/100/1000 ASIC's on the linecard. Each of these ASIC's support 8 physical 10/100/1000 ports. Each of these ports has access into the local ASIC's shared memory pool.

3.3.9. Gigabit Ethernet Line Cards (WS-X6724, WS-X6748)

Linecards in this group include the following:

- WS-X6724-SFP
- WS-X6748-SFP
- WS-X6748-GE-TX

	WS-X6724	WS-X6748
Number of GE Ports	24	48
# Port ASIC's on the linecard	2	4
# Physical Ports per Port ASIC	24	24
Per Port Buffering	Yes	Yes
Buffer on Receive Side	166KB	166KB
Buffer on Transmit Side	1.2MB	1.2MB
Transmit Queue Structure per port	1P7Q8T	1P7Q8T
Receive Queue Structure per port	1Q8T	1Q8T
Receive Queue Structure per port with DFC	2Q8T	2Q8T
Receive (RX) Strict Priority Queue	No	No
Transmit (TX) Strict Priority Queue	Yes	Yes

Table 12 - QoS on Gigabit Ethernet Line Cards (WS-X6724, WS-X6748)

Figure 10 - WS-X6748-GETX Linecard

These new linecards supports a strict priority queue on the transmit side for each port. A total of 1.33Mb of buffering is available for each port. This is 166Kb is set aside for the receive queues, and 1.2Mb is set aside for the transmit queues.

3.3.10. 10-Gigabit Ethernet Line Cards (WS-X6502-10GE)

Late in 2001, Cisco introduced a 10GE line card providing 1 port of 10GE per line card. This module uses one slot in the 6500 chassis. For the 10GE port, the queue structure of 1P1Q8T and 1P2Q2T is used providing two receive queues and three transmit queues. One of the receive queues and one transmit queue is designated as a strict priority queue. Buffering is also allocated for the port providing 256K of receive buffering and 64Mb of transmit buffering. This port implements a 1p1q8t queue structure for the receive side and a 1p2q1t queue structure for the transmit side. Queue structures are detailed later in the document.

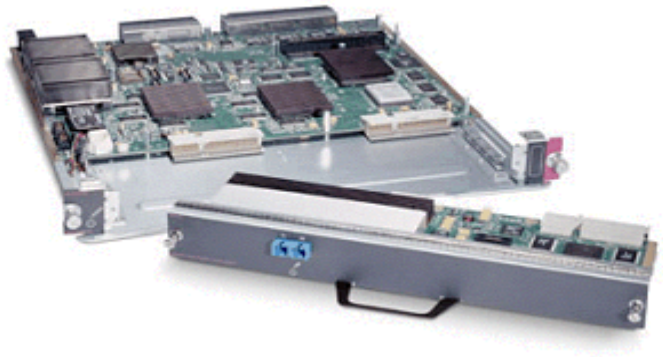

Figure 11 - WS-X6502-10GE Linecard

3.3.11. 10-Gigabit Ethernet Line Cards (WS-X6704-10GE)

The most recent addition to the 10GE family is the 4-port 10GE linecard (WS-X6704-10GE). Details for the QoS implementation on this linecard are detailed below:

	WS-X6704
Number of 10 GE Ports	4
# Port ASIC's on the linecard	4
# Physical Ports per Port ASIC	1
Per Port Buffering	Yes
Buffer on Receive Side	2MB
Buffer on Transmit Side	14MB
Transmit Queue Structure per port	1P7Q8T
Receive Queue Structure per port	1Q8T
Receive Queue Structure per port with DFC	8Q8T
Receive (RX) Strict Priority Queue	No
Transmit (TX) Strict Priority Queue	Yes

Table 13 - QoS on 10-Gigabit Ethernet Line Cards (WS-X6704-10GE)

This linecard requires a Supervisor 720 to be installed. It supports the 1Q8T and 1P7Q8T queue structures. A strict priority queue is supported on the transmit side only. When a DFC3 is present, the input queue structure changes to 8Q8T. The WS-X6704-10GE module is pictured below:

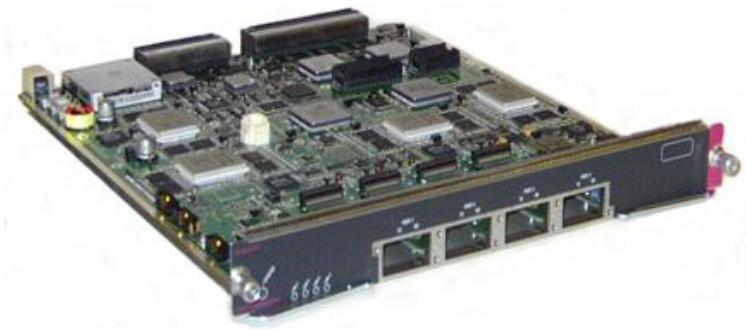

Figure 12 - WS-X6704.10GE Linecard

More details on queue structures and buffer assignments for each linecard types are detailed in Appendix One.

3.4 Catalyst 6500 QoS Hardware Summary

The hardware components that perform the above QoS functions in the Catalyst 6500 are detailed in the table below:

QoS Process	Catalyst 6500 Component that performs function
Input Scheduling	Performed by port ASIC's on linecard L2 only with or without the PFC
Ingress Trust	Performed by port ASIC's on linecard
Ingress Congestion Management	Performed by port ASIC's on linecard
Ingress Classification	Performed by PFC
Egress Classification	Performed by PFC
Ingress Policing	Performed by PFC via L3 Forwarding Engine
Egress Policing	Performed by PFC via L3 Forwarding Engine
Packet Re-write	Performed by port ASIC's on linecard
Output Congestion Management	Performed by port ASIC's on linecard
Output Scheduling	Performed by port ASIC's on linecard

Figure 13 - QoS functions in a Catalyst 6500

4. Catalyst 6500 Software support for QoS

The Catalyst 6500 supports two operating systems. The original software platform that shipped with the Catalyst 6500 was Catalyst OS (or more commonly referred to as CatOS) and this was derived from the code base used on the Catalyst 5000 platform. Cisco introduced Cisco IOS (previously known as Native IOS) at a later date, and this uses a code base derived from the Cisco Router IOS. Both OS platforms (CatOS and Cisco IOS) implement software support to enable the QoS hardware features on the Catalyst 6500 platform. There is no difference in the QoS capabilities of the Catalyst 6500 when running either OS platform. The QoS differences between the OS platforms lie in the configuration options used to activate various QoS features.

In subsequent sections when explaining the configuration of QoS, this paper will use configuration examples from IOS platform only.

4.1 Priority Mechanisms in IP and Ethernet

For any QoS services to be applied to data, there must be a way to "tag" (or prioritize) an IP Packet or an Ethernet frame. The Type of Service (ToS) in the IPv4 header and the Class of Service (CoS) fields in the Ethernet header are used to achieve this. These are described in more detail below.

4.1.1. Type of Service (ToS)

Type of Service (ToS) is a one-byte field that exists in an IPv4 header. The ToS field consists of 8 bits of which the first 3 bits were used to indicate the priority of the IP Packet. These first 3 bits are referred to as

the IP Precedence bits. These bits can be set from 0 to 7, (0 being the lowest priority and 7 being the highest priority). Support has been around for setting IP Precedence in IOS for many years. Support for resetting IP Precedence can be done by the MSFC or by the PFC (independent of the MSFC). A Trust setting of "untrusted" (discussed later in this paper) can also wipe out any IP Precedence settings on an incoming frame (please refer to the section on "Trust" later in this paper).

The values that can be set for IP Precedence are:

IP Precedence bits	IP Precedence Value
000	Routine
001	Priority
010	Intermediate
011	Flash
100	Flash Override
101	Critical
110	Internetwork Control
111	Network Control

Figure 14 - IP Precedence Priority Bit Settings

The diagram below is a representation of the IP Precedence bits in the ToS header. The three Most Significant Bits (MSB) are interpreted as the IP Precedence bits.

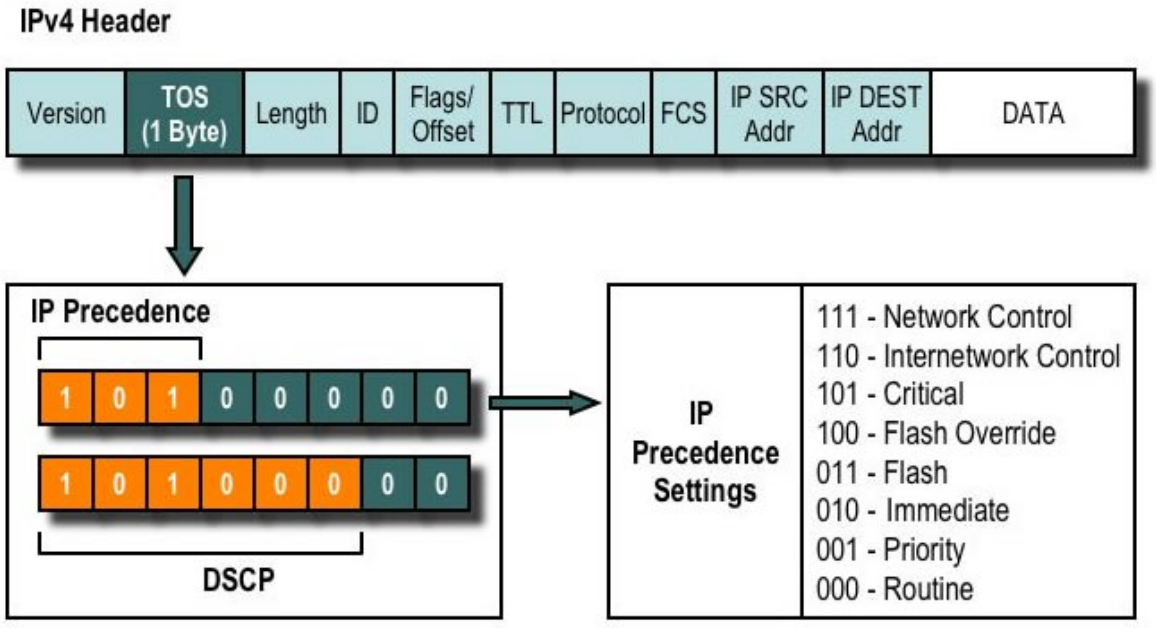

Figure 15 - ToS Byte Settings

More recently, the use of the ToS field has been expanded to encompass the Six Most Significant Bits, referred to as DSCP. DSCP (Differentiated Services Code Point) results in 64 priority values (2 to the power of 6) that can be assigned to the IPv4 Packet. DSCP (also referred to as DiffServ) is defined in RFC 2474 (Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers).

The Catalyst 6500 can manipulate (modify) the ToS priority bits and this can be achieved using both the PFC and/or the MSFC. When a data frame comes into the switch, it will be assigned a DSCP value, which is derived from a predefined default or from existing priority settings. This DSCP value is used internally in the switch to assign levels of service (QoS policies) defined by the administrator. The DSCP can already exist in a frame and be used, or the DSCP can be derived from the existing CoS, IP Precedence or DSCP in the frame (should the port be trusted). A map is used internally in the switch to derive the DSCP (explained later). With 8 possible CoS/IP Precedence values and 64 possible DSCP values, the default map will map CoS/IPPrec 0 to DSCP 0, CoS/IPPrec 1 to DSCP 8, CoS/IPPrec 2 to DSCP 16, and so on. These default mappings can be overridden by the administrator. When the frame is scheduled to an outbound port, the CoS can be re-written and the DSCP value is used to derive the new CoS. A new feature of the PFC3 allows the DSCP to be left alone while at the same time rewriting CoS. This is a new configurable option recently released for the PFC3 only.

4.1.2. Class of Service (CoS)

CoS refers to three bits in either an ISL header or an 802.1Q header that is used to indicate the priority of the Ethernet frame as it passes through a switch network.

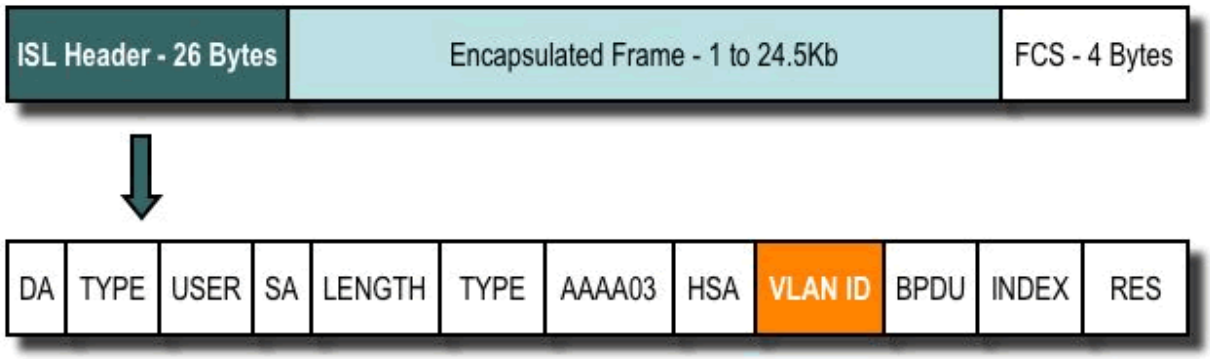

Figure 16 - Inter Switch Link (ISL) Frame Format

In the ISL header, a VLAN tag (seen in red above) includes 3 bits, which are used to assign a priority value to this frame.

For the purposes of this document, I will only be referring to the use of the 802.1Q header. The Class of Service bits in the 802.1Q header are officially referred to as the 802.1p bits. Not surprisingly there are three CoS bits which match the number of bits used for IP Precedence. In many networks, a packet may traverse both layer 2 and layer 3 domains, so to maintain QoS, the ToS can be mapped to CoS and vice versa.

Shown below is an Ethernet frame tagged with an 802.1Q field, which consists of a 2-byte Ether-type and a 2-byte Tag. Within the 2-byte tag are the user priority bits (known as 802.1p).

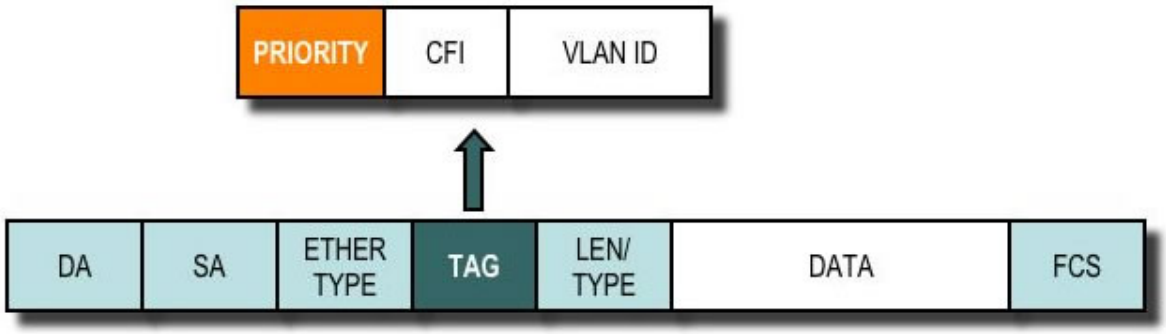

Figure 17 – Class of Service User Priority (802.1p) in an Ethernet frame

5. QoS Flow in the Catalyst 6500

QoS in the Catalyst 6500 is the most comprehensive implementation of QoS found in all of the current Cisco Catalyst Switches. The following sections will describe how the various QoS processes are applied to a frame as it transits the switch.

Before continuing further, the flow of QoS though the Catalyst 6500 switch will be reviewed. Earlier it was noted that there are a number of QOS elements that many layer 2 and layer 3 switches can offer, those being Classification, Input Queue Scheduling, Policing, Rewriting and Output Queue Scheduling. The following summarizes how the Catalyst 6500 implements these elements.

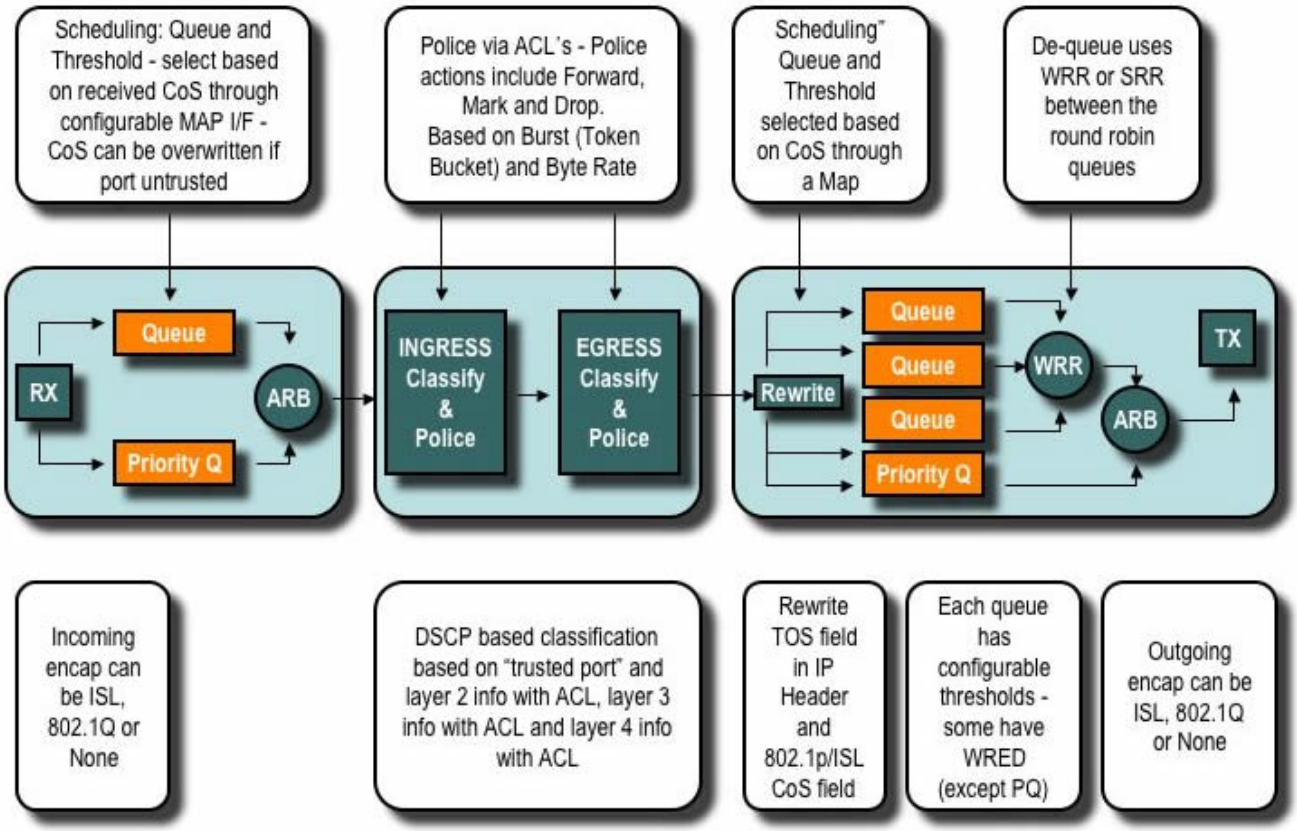

Figure 18 - QoS Flow on the Catalyst 6500

A frame enters the switch and is initially processed by the port ASIC that received the frame. It will place the frame into a receive queue.

The port ASIC will use the CoS bits as an indicator of which queue to place the frame into (if more than one input queue is present). If the port is classified as un-trusted, then the port ASIC can overwrite the existing CoS bits based on a predefined value.

The frame is then passed to the L2/L3 Forwarding Engine (PFC), which will classify and optionally police (rate limit) the frame. Classification is the process of assigning the frame a DSCP value (refer to earlier explanation of DSCP), which **is used internally** by switch for processing the frame and assigning it a service level. The DSCP will be derived from one of the following:

1. An existing DSCP value set prior to the frame entering the switch
2. Derived from the received IP Precedence bits already set in the IPv4 header. As there are 64 DSCP values and only 8 IP Precedence values, the administrator will configure a mapping that is used by the switch to derive the DSCP. Default mappings are in place should the administrator not configure the maps. Mappings are explained later in this document.
3. Derived from the received CoS bits already set prior to the frame entering the switch. Like IP Precedence, there are a maximum of 8 CoS values each of which must be mapped to one of 64.DSCP values. This map can be configured or the switch can use the default map in place.
4. The DSCP can be set for the frame using a DSCP default value typically assigned though an Access Control List (ACL) entry.

After a DSCP value is assigned to the frame, policing (rate limiting) is applied should a policing configuration exist. Policing will limit the flow of data through the PFC by dropping or marking down traffic that is out of profile. Out of Profile is a term used to indicate that traffic has exceeded a limit defined by the administrator as the amount of bits per second the PFC will send. Out of profile traffic can be dropped, or, the data can still be sent but CoS value is marked down. The PFC1 and PFC2 currently only support input policing (rate limiting). The PFC3 supports input and output policing. The output-policing feature of the PFC3 applies to routed (layer 3) ports or VLAN interfaces (switch SVI interface – this is discussed in more detail later in the paper).

The PFC will then pass the frame to the egress port for processing. At this point, a rewrite process is invoked to modify the CoS values in the frame and the ToS value in the IPv4 header. Prior to passing the frame to the port ASIC, the PFC will derive the CoS based on internal DSCP. The port ASIC will then use the CoS value passed to it to place the frame into the appropriate queue. While the frame is in the queue, the port ASIC will monitor the buffers and implement WRED to avoid the buffers from overflowing. A Round Robin scheduling algorithm is then used to schedule and transmit frames from the egress port

Each of the sections below will explore this flow in more detail giving configuration examples for each of the steps described above.

6. Queues, Buffers, Thresholds and Mappings

Before QoS configuration is described in detail, certain terms must be explained further to ensure the reader fully understands the QoS configuration capabilities of the switch.

6.1 Queues

Each port in the switch has a series of input and output queues that are used as temporary storage areas for data. Catalyst 6500 line cards implement different number of queues for each port. The queues are usually implemented in hardware ASIC's for each port. This is different to routers where the queues are virtualised by the software. On the first generation Catalyst 6500 line cards, the typical queue configuration included one input queue and two output queues. Later line cards use enhanced ASIC's which incorporated additional queues. One innovation included support for a special strict priority (SP) queue, which is ideal for storing latency sensitive traffic like Voice over IP (VoIP). Data in this queue is serviced in a strict priority fashion. That is, if a frame arrives in the SP queue, scheduling and transmission of frames from the lower queues is ceased in order to process the frame in the strict priority queue. Only when the SP queue is empty will scheduling of packets from the lower queue(s) recommence.

When a frame arrives at an ingress port and congestion is present, it will be placed into a queue. The decision behind which queue the frame is placed in is determined by the CoS value in the Ethernet header of the incoming frame.

On egress, a scheduling algorithm will be employed to empty the transmit (output) queue. There are a number of Round Robin techniques available (dependent on the hardware and software in use) to perform this function. These Round Robin techniques include Weighted Round Robin (WRR), Deficit Weighted Round Robin (DWRR) and Shaped Round Robin (SRR). While each of these are explored later, in the case of WRR and DWRR, each queue uses a weighting to dictate how much data will be emptied from the queue before moving onto the next queue. For SRR, a rate (or limit) is applied on a per queue basis and this dictates the upper limit of bandwidth that can be used by this queue.

6.2 Buffers

Each queue is assigned a certain amount of buffer space to store transit data. Resident on the port ASIC is buffer memory, which is split up and allocated on a per port basis. Per port buffering for each of the linecards is detailed in Appendix One.

6.3 Thresholds

One aspect of normal data transmission is that if a packet is dropped, it will (if we are talking TCP flows) result in that packet being retransmitted by the TCP endpoint. At times of congestion, this can add to the load on the network and potentially cause buffers to overload even more. As a means of ensuring the buffers do not overflow, the Catalyst 6500 switch employs a number of techniques to avoid this happening.

Thresholds are arbitrary internal levels assigned by the switch port ASIC (and configurable by the administrator) that define utilization points in the queue at which the congestion management algorithm can start dropping data. On the Catalyst 6500 ports, each queue structure defines a number of thresholds that are associated with *input* queues. The same applies to the number of thresholds associated with *output* queues.

What data is eligible to be dropped when a threshold is breached is determined by the priority of the packet. Different CoS priorities are assigned to each threshold. In this way, when the threshold is

exceeded, the congestion management algorithm immediately knows which packets with which priority value are eligible to be dropped.

6.4 Mappings

In the queues and threshold section above, it was mentioned that the CoS value in the Ethernet frame is used to determine which queue to place the frame into and at what point of the buffer filling up is a frame eligible to be dropped. A map is the vehicle used to assign a priority value to a threshold in a queue.

When QoS is configured on the Catalyst 6500, default mappings are enabled that define the following:

- At what thresholds frames with specific CoS values are eligible to be dropped
- Which queue a frame is placed into (based on its CoS value)

While the default mappings exist, these default mappings can be overridden by the administrator. Mapping exist for the following:

- Map a CoS value on an incoming frame to a DSCP value
- Map an IP Precedence value on an incoming frame to a DSCP value
- Map a DSCP value to a CoS value for an outgoing frame
- Map CoS values to drop thresholds on receive queues
- Map CoS values to drop thresholds on transmit queues
- Mapping DSCP markdown values for frames that exceed policing statements
- Mapping a CoS value to a frame with a specific destination MAC address

An example of a map can be shown in the following diagram:

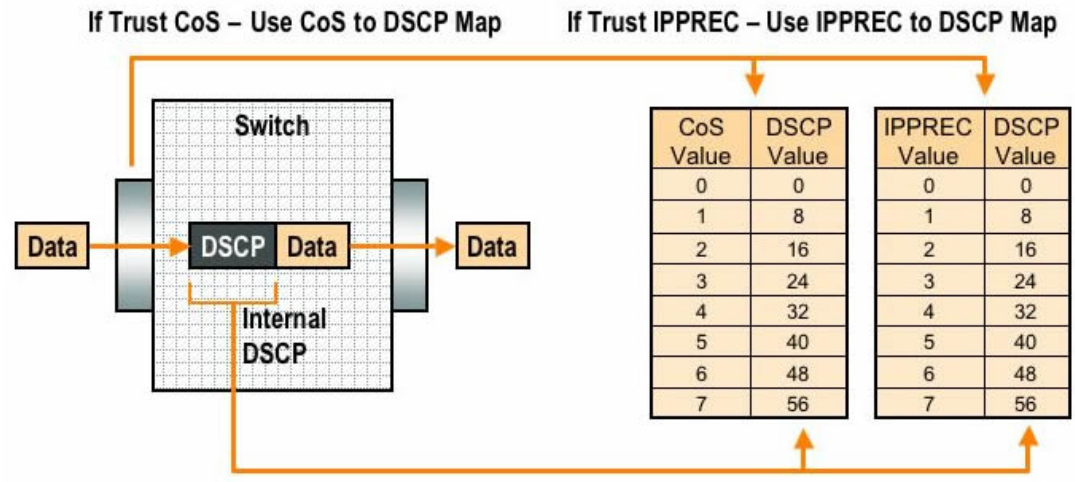

Figure 19 - Mapping Priority to Internal DSCP

This example map is an ingress map used to take either the CoS or IP Precedence value of an incoming frame and map it to an internal DSCP value. The mapping used for a port can be viewed using the following command:

```
Cat6500# show queueing int f4/2
```

```
<snip>
Packets dropped on Transmit:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
    1     1               0  [0 1 ]
    1     2               0  [2 3 ]
    2     1               0  [4 6 ]
    2     2               0* [7 ]
    3     1               0* [5 ]
                                * - shared transmit counter

  Packets dropped on Receive:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
    1     1               0  [0 1 ]
    1     2               0  [2 3 ]
    1     3               0  [4 6 ]
    1     4               0* [7 ]
    2     1               0* [5 ]
<snip>
```

6.5 Weighted Random Early Discard and Round Robin Scheduling

The Weighted Random Early Discard (WRED) congestion management mechanism and the Round Robin scheduling mechanisms are extremely powerful algorithms resident on the Catalyst 6500 port ASIC's. There are three implementations of Round Robin scheduling on the Catalyst 6500 and they include Weighted Round Robin (WRR), Deficit Weighted Round Robin (DWRR) and Shaped Round Robin (SRR). WRED and all the Round Robin scheduling options use the priority tag (CoS) inside an Ethernet frame to provide enhanced buffer management and outbound scheduling. All of these are explained further later in this document.

6.5.1. WRED

WRED is a buffer management algorithm employed by the Catalyst 6500 to minimize the impact of dropping high priority traffic at times of congestion. WRED is based on the RED (Random Early Discard) algorithm used in some of our competitors switching products.

Before we look at RED and WRED, let's quickly revisit TCP flow management. Flow management ensures that the TCP sender does not overwhelm the network. The "TCP slow-start" algorithm (defined in RFC2001) is part of the solution to address this. It dictates that when a flow starts, a single packet is sent before it waits for an acknowledgement (ACK). When the ACK is received, the TCP endpoint will send two packets and wait for the next ACK before sending more data. This process gradually increases the number of packets sent before each ACK is received. This will continue until the flow reaches a transmission level (i.e. send "x" number of packets) that the network can handle without the load incurring congestion. Should congestion occur, the slow-start algorithm will throttle back the window size (i.e. number of packets sent before waiting for an acknowledgement). This will normalize transmission to a set number of frames that the network can handle without dropping them.

Back to RED and WRED. RED will monitor a queue as it starts to fill up. Once a certain threshold has been exceeded, packets will start to be dropped randomly. No regard is given to specific flows; rather random packets will be dropped. These packets could be from high or low priority flows. Dropped packets can be part of a single flow or multiple TCP flows. If multiple flows are impacted, as described above, this can have a considerable impact on each flows window size.

Unlike RED, WRED is not so random when dropping frames. WRED takes into consideration the priority of the frames (in the Cat6500 case it uses to CoS value). With WRED the administrator assigns frames with certain CoS values to specific thresholds. Once these thresholds are exceeded, frames with CoS values that are mapped to these thresholds are eligible to be dropped. Other frames with CoS values assigned to the higher thresholds are en-queued. This process allows for higher priority flows to be kept intact keeping their larger window sizes intact and minimizing the latency involved in getting the packets from the sender to the receiver.

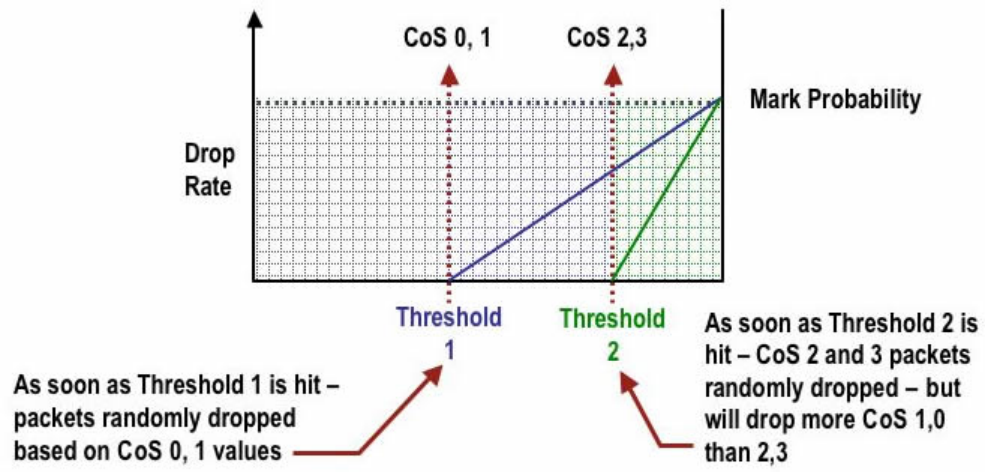

Figure 20 - WRED high and low thresholds

WRED also supports high and low threshold settings for a given threshold. In the diagram above, you can see there are two thresholds (1 and 2). Threshold #1 is a level under which no traffic mapped to that threshold is dropped. When that threshold is exceeded, traffic mapped to that threshold (CoS 0 and 1) is eligible to be dropped. The more the threshold is exceeded, the greater rate at which those packets are dropped. When Threshold 2 is exceeded, traffic marked with CoS 2 and 3 will start to be dropped, but at a rate less than traffic marked CoS 0 and 1.

How do you know if your line card supports WRED? Simply run the following command and in the output is a section indicating support or no support for WRED on that port. The first example below shows a port that does not support WRED. This is a show output of a 10/100 port on the WS-X6148-RJ45:

```
Cat6500# show queueing int f4/1

<snip>

queue tail-drop-thresholds
    --------------------------
    1     80[1] 100[2]
```

```
    2     80[1] 100[2] <snip>
```

Note that in the output above it refers to the Tail Drop thresholds. The next example below shows a port that does support WRED. This is the same CLI output from a 10/100/1000 port on a WS-X6516-GETX:

```
Cat6500# show queueing int f4/1
<snip>

queue random-detect-min-thresholds
    ----------------------------------
      1    40[1] 70[2]
      2    40[1] 70[2]

    queue random-detect-max-thresholds
    ----------------------------------
      1    70[1] 100[2]
      2    70[1] 100[2]
<snip>
```

Should WRED not be available on a port, the port will use a Tail Drop method of buffer management. Tail Drop, as its name implies, simply drops incoming frames once the buffers have been fully utilized

6.5.2. WRR

WRR is used to schedule egress traffic from transmit queues. A normal round robin algorithm will alternate between transmit queues sending an equal number of packets from each queue before moving to the next queue. The **weighted** aspect of WRR allows the scheduling algorithm to inspect a weighting that has been assigned to each queue. This defines how much bandwidth each queue has access too. The WRR scheduling algorithm will empty out more data from queues with a higher weighting (each interval) than other queues, thus providing a bias for designated queues.

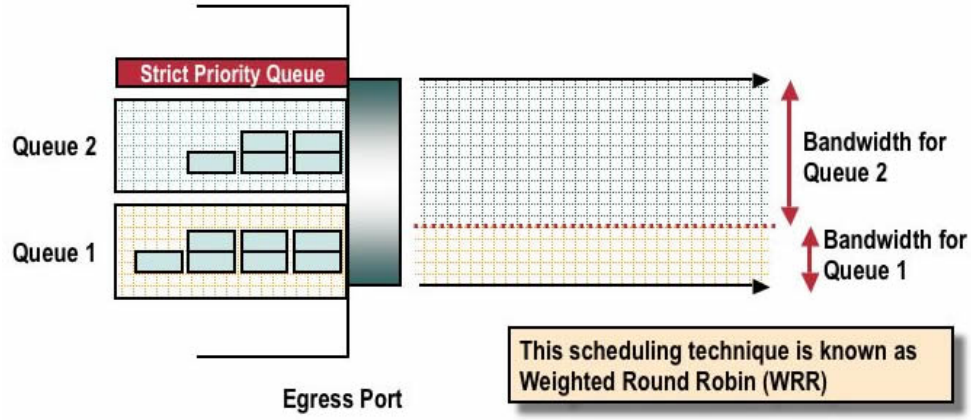

Figure 21 - Weighted Round Robin

When configuring WRR, a weight is assigned to the queue. The weight number is a number from 1 to 255. It is important to point out that this number does not represent a percentage. If you had two queues, and you wanted queue 1 to have 4 times as much bandwidth as queue2, then you could configure the weights as 4 and 1, 8 and 2, 80 and 20, 100 and 25, etc. As you can see, the weights do not need to add up to 100, rather they represent a ratio that is used by the WRR algorithm to assign bandwidth to individual queues.

WRR configurable settings can be seen in the following show output:

```
Cat6500# show qos info config 9/1
<snip>
Transmit queues [type = 1p2q2t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR low             2
       2         WRR high            2
       3         Priority            1

    WRR bandwidth ratios:  100[queue 1] 255[queue 2]
    queue-limit ratios:     80[queue 1]  20[queue 2]
```

Configuration for WRR and the other aspects of what have been described above are explained in the following section.

6.5.3. Deficit Weighted Round Robin

DWRR is a feature that is used on egress (transmit) queues. Explaining DWRR is best served by using an example. Let's assume a switch port has 3 queues and we are using the normal WRR algorithm. Queue 1 has been given access to 50% of the bandwidth, Queue 2 has 30% and Queue 3 has 20%. If the WRR algorithm is servicing Queue 2 and on this service pass it has used 99.9% of its allotted bandwidth, the WRR algorithm will still send out the next packet in the queue as the queues allotted bandwidth allocation has not yet been used up. When it sends the next packet, it will exceed the amount of bandwidth that was configured for this queue. Statically over time, Queue 2 may end up using a lot more bandwidth than it was initially configured for when using WRR.

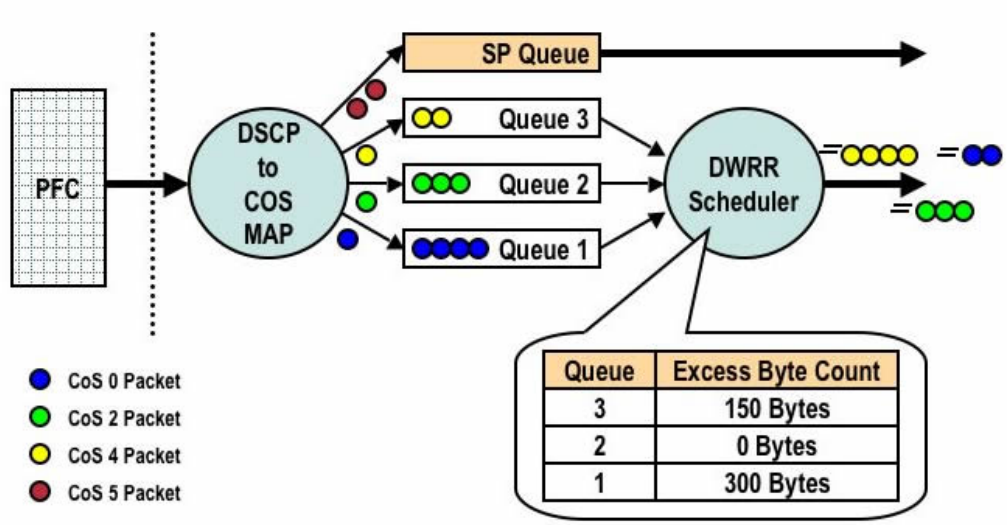

Figure 22 - Deficit Weighted Round Robin

DWRR alleviates this problem from occurring. If the queue uses more bandwidth than it was allotted, DWRR keeps a tally of the extra bandwidth used on that pass. On the next pass through the queues, the DWRR algorithm will subtract the extra bandwidth used on the last pass for this turn. This means

statistically over a period of time that each queue will use bandwidth that is much closer to the configured amount for that queue.

6.5.4. Shaped Round Robin

SRR is a recent addition to the scheduling capabilities of the Catalyst 6500 family. At the time of writing this paper, support for SRR is only available on the uplink ports of the Supervisor 32. SRR is different to WRR in that the SRR algorithm provides a way to shape outbound traffic to a stated rate. In some respects, it is similar to a policer except that traffic in excess of the rate will be buffered rather than dropped as with a policer.

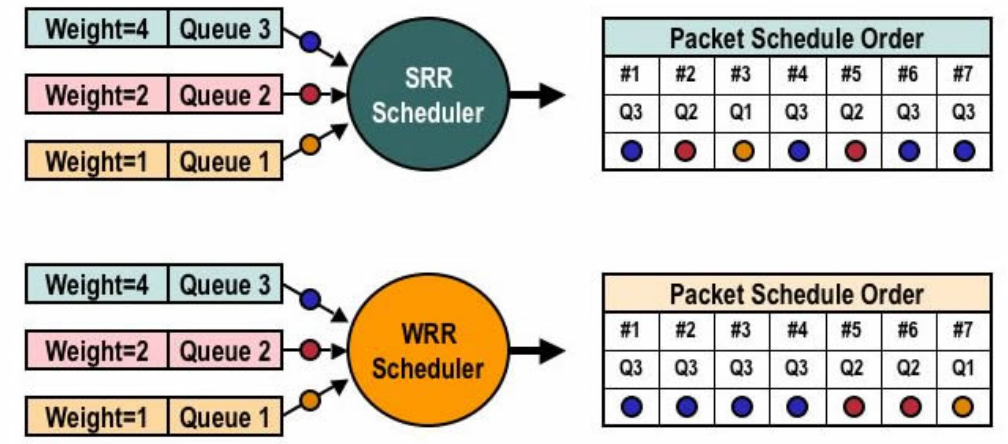

Figure 23 - SRR compared to WRR

The shaper is implemented on a per queue basis and has the effect of smoothing transient bursts of data that pass through that port. SRR will limit traffic output for that queue to the stated rate. Even if bandwidth is available, the rate will never exceed what is configured. SRR also modifies the way in which it schedules data when compared to the WRR algorithm. This can be seen in the diagram above which shows a representation of the packet scheduling order for both algorithms.

6.5.5. Strict Priority Queuing

Selected linecards in the Catalyst 6500 family support a strict priority queue on a per port basis. This queue operates outside of the bounds set by the Round Robin scheduling algorithm described above. The purpose of a strict priority queue is to facilitate support for latency sensitive traffic that gets queued on the linecard. When a packet is placed into a strict priority queue, scheduling of packets from WRR queues will cease and the packet(s) in the strict priority queue will be transmitted. Only when the strict priority queue is empty will the scheduling process recommence sending packets from WRR (DWRR/SRR) queues.

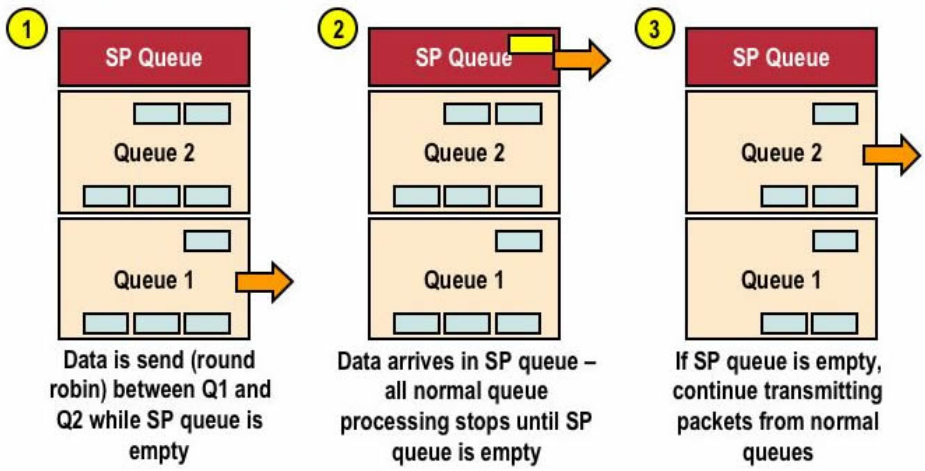

Figure 24 - Strict Priority Queuing process

7. Configuring (Port ASIC based) QoS on the Catalyst 6500

QoS configuration instructs either the port ASIC or the PFC to perform a QoS action. The following sections will look at QoS configuration for both these processes. On the port ASIC, QoS configuration affects both inbound and outbound traffic flows.

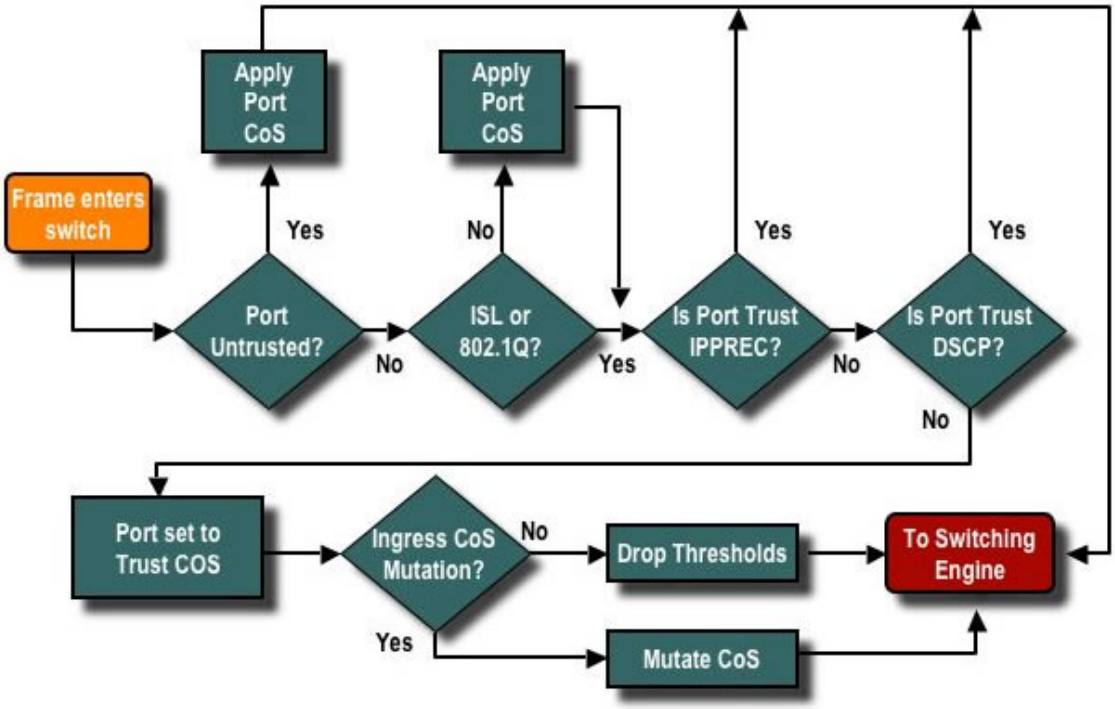

Figure 25 - Ingress QoS Processing Flow

From the above diagram it can be seen that the following QoS configuration processes apply

1. Trust states of ports
2. Applying port based CoS
3. Determine CoS Mutation

4. Receive Drop Threshold assignment
5. CoS to Receive Drop threshold maps

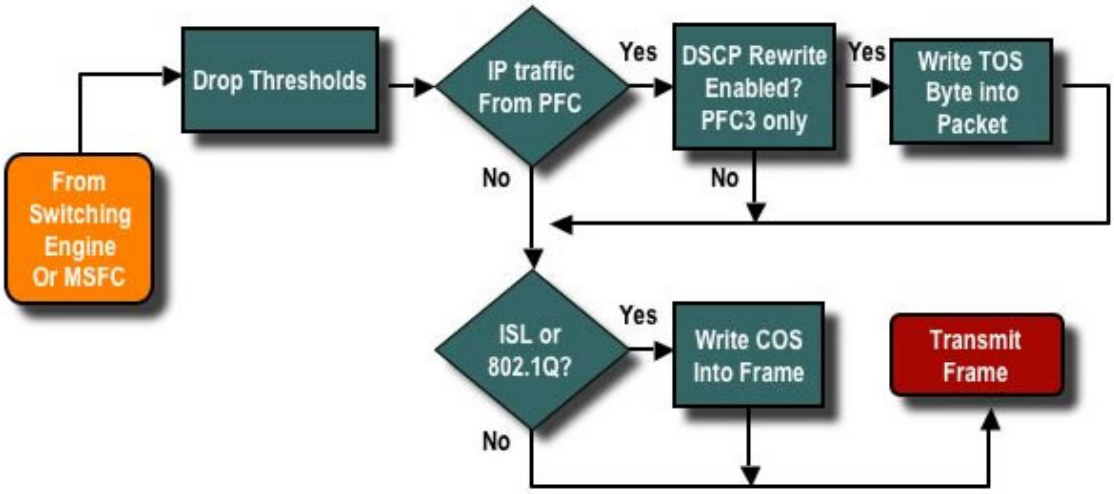

Figure 26 - Egress QoS Processing Flow

When a frame is processed by either the MSFC or the PFC, it is passed to the outbound port ASIC for further processing. Any frames processed by the MSFC will have their CoS values reset to zero. This needs to be taken into consideration for QoS processing on outbound ports.

The above diagram shows QoS processing performed by the Port ASIC for outbound traffic. Some of the processes invoked on outbound QoS processing includes:

1. Transmit Tail Drop and WRED threshold assignments
2. CoS to Transmit Tail Drop and WRED maps
3. DSCP Rewrite
4. CoS Rewrite for ISL/802.1Q frames

Also, not shown on the diagram above, is the process of reassigning the CoS to the outbound frame using a DSCP to CoS Map; this would occur prior to the egress port performing congestion management.

The following sections will look in more detail at the QoS configuration capabilities of the port based ASIC's.

7.1 Enabling QoS

Before any QOS configuration can take place on the Catalyst 6500, QoS must first be enabled on the switch. This is achieved using the following command:

```
Cat6500(config)# mls qos
```

The state of QoS on the given switch can be inspected using the following command

```
Cat6500# show mls qos
  QoS is enabled globally
  QoS ip packet dscp rewrite enabled globally
```

```
 Input mode for GRE Tunnel is Pipe mode
 Input mode for MPLS is Pipe mode
 Vlan or Portchannel(Multi-Earl) policies supported: Yes
 Egress policies supported: Yes


----- Module [6] -----
 QoS global counters:
   Total packets: 20
   IP shortcut packets: 0
   Packets dropped by policing: 0
   IP packets with TOS changed by policing: 0
   IP packets with COS changed by policing: 0
   Non-IP packets with COS changed by policing: 0
   MPLS packets with EXP changed by policing: 0
```

When QoS is enabled in the Catalyst 6500, the switch will set a series of QoS defaults for the switch. The global defaults include the following settings:

QoS Feature	Default setting
Trust state of each port	Un-trusted
Port CoS Default value	Zero (0)
Microflow Policing	Enabled
Intra VLAN Microflow Policing	Disabled
QoS is Port based or VLAN based	Port Based
CoS to DSCP Mapping (DSCP set from CoS value)	CoS 0 = DSCP 0 CoS 1 = DSCP 8 CoS 2 = DSCP 16 CoS 3 = DSCP 24 CoS 4 = DSCP 32 CoS 5 = DSCP 40 CoS 6 = DSCP 48 CoS 7 = DSCP 56
IP Precedence to DSCP Map (DSCP set from IP Precedence value)	IP precedence 0 = DSCP 0 IP precedence 1 = DSCP 8 IP precedence 2 = DSCP 16 IP precedence 3 = DSCP 24 IP precedence 4 = DSCP 32 IP precedence 5 = DSCP 40 IP precedence 6 = DSCP 48 IP precedence 7 = DSCP 56

DSCP to CoS map (CoS set from DSCP values)	DSCP 0-7 = CoS 0 DSCP 8-15 = CoS 1 DSCP 16-23 = CoS 2 DSCP 24-31 = CoS 3 DSCP 32-39 = CoS 4 DSCP 40-47 = CoS 5 DSCP 48-55 = CoS 6 DSCP 56-63 = CoS 7
Policers	None Enabled
Policy Maps	None Enabled
Protocol Independent MAC Filtering	Disabled
VLAN Based MAC ACL QoS filtering	Disabled

Table 14 – Summary of default states set when QoS is enabled

There are also a number of default settings that are applied to receive and transmit queues and these include the following:

QoS Feature	Default setting for Queue Size Percentage
Queue 2Q8T	Low priority queue – 80% High priority queue – 20%
Queue 8Q8T	Lowest priority queue – 80% Intermediate queues – 0% Highest priority queue – 20%
Queue 2Q2T	Low priority queue – 80% High priority queue – 20%
Queue 1P2Q2T	Low priority queue – 70% High priority queue – 15% Strict Priority queue – 15%
Queue 1P2Q1T	Low priority queue – 70% High priority queue – 15% Strict Priority queue – 15%
Queue 1P3Q8T	Low priority queue – 50% Medium priority queue – 20% High priority queue – 15% Strict Priority queue – 15%
Queue 1P7Q8T	Standard queue 1 – 50% Standard queue 2 – 20% Standard queue 3 – 15% Standard queue 4 thru 7 – 0% Strict Priority queue – 15%

Table 15 - Default Transmit and Receive Queue Settings

Additional defaults can be found in the QoS configuration guides on CCO for a given software release.

7.2 Trusted and Un-trusted Ports

Any given port on the Catalyst 6500 can be configured as trusted or un-trusted. The trust state of the port dictates how it marks, classifies and schedules the frame as it transits the switch. By default, all ports are in the un-trusted state. This means that any packet entering that port will have its ToS and CoS rewritten with a zero value. Consideration must thus be given when QoS is enabled for devices that tag their packets with a priority. Devices such as IP phones, Call managers, key servers, etc should have their trust setting reviewed.

7.2.1. Un-trusted Ports (Default setting for ports)

Should a port be configured as an un-trusted port, a frame upon initially entering the port will have its CoS and ToS value reset by the port ASIC to zero. This means the frame will be given the lowest priority service on its path through the switch. Alternatively, the administrator can reset the CoS value of any Ethernet frame that enters an un-trusted port to a pre-determined value. Configuring this will be discussed in a later section.

Setting the port as un-trusted will instruct the switch to **NOT** perform any congestion avoidance. Congestion avoidance is the method used to drop frames based on their CoS values once they exceed thresholds defined for that queue. All frames entering this port will equally be eligible to be dropped once the buffers reach 100%. This process is the same when trust-ipprec or trust-dscp on ingress is used

NOTE: For **Cisco IOS**, the software, setting trust is not supported on 1Q4T ports except Gigabit Ethernet ports.

```
Cat6500(config)# interface gigabitethernet 1/1
Cat6500(config-if)# no mls qos trust
```

In the example above, you enter the interface configuration and then apply the **no** form of the command to set the port as un-trusted.

7.2.2. Trusted Ports

Sometimes Ethernet frames entering a switch will have either a CoS or ToS settings that the administrator wants the switch to maintain as the frame transits the switch. For this traffic, the administrator can set the trust state of a port where that traffic comes into the switch as trusted.

As mentioned earlier, the switch uses a DSCP value internally to assign a predetermined level of service to that frame. As a frame enters a trusted port, the administrator can configure the port to look at either the existing CoS, IP Precedence or DSCP value to set the internal DSCP value. Alternatively, the administrator can set a predefined DSCP to every packet that enters the port This is achieved by attaching an ACL with a predefined DSCP.

Setting the trust state of a port to trusted can be achieved using the following command
With **Cisco IOS** the setting of trust can be performed on a Gigabit Ethernet interface, 10GE interface, 10/100 ports on the WS-X6548-RJ45/21 line card and 100Mb ports on the WS-X6524 100FX ports. as well as the 6148/6548-GE-TX linecards)

```
Cat6500(config)# interface gigabitethernet 5/4
Cat6500(config-if)# mls qos trust ip-precedence
```

This example sets the trust state of Gigabit Ethernet port 5/4 to trusted. The frames IP precedence value will be used to derive the DSCP value.

7.3 Preserving the Received ToS Byte (DSCP Transparency)

When a packet enters the switch, the switch will derive an internal DSCP value from the incoming priority bits (based on the trust setting). This internal DSCP is used to write the ToS byte when the packet egresses the switch. This action can thus change the ingress DSCP setting.

Those customers who would like to preserve the integrity of their DSCP can use this feature to avoid the PFC rewriting the DSCP on egress. This feature is supported on the PFC3, PFC3B and PFC3BXL. It is worth noting that only the PFC3B and PFC3BXL support this for MPLS frames and tunnelled traffic (i.e. GRE and IP-in-IP). This can be achieved using the following command:

```
Cat6500(config)# mls qos rewrite ip ?
  dscp  packet ip dscp rewrite enable/disable
Cat6500(config)# mls qos rewrite ip dscp
```

7.4 Port based vs. VLAN Based QoS

When a QoS policy is applied to a port, the PFC needs to know whether this policy is applicable to this port only or whether this policy is part of a global policy that applies to all ports in a given VLAN. By default, policies applied to a port are viewed as only being applicable for that port. If, however, it is required to apply a policy on a VLAN (which implicitly means the policy should be applied to all ports in that VLAN), then the port needs to be marked indicating its part of a VLAN QoS policy. Should this be a requirement, then the following command needs to be applied to all ports in that VLAN.

```
Cat6500(config)# mls qos vlan-based
```

7.5 Setting the Switch to Queuing-Only Mode

Configuring trust requires QoS to be enabled on the Catalyst 6500. Enabling QoS also activates all of the QoS features (Classification and Policing) on the PFC. It is possible a scenario might arise where trust is required, but classification and policing are not. When this situation arises, the switch can be configured to operate in a mode where PFC QoS functions are disabled while leaving other QoS functions untouched. This mode is referred to as Queuing-only mode and can be enabled as follows:

```
Cat6500(config)# mls qos queueing-only
```

7.6 Input Classification and Setting Port Based CoS

On ingress to a switch port, an Ethernet frame can have its CoS changed if it meets one of the following two criteria:

1. The port is configured as untrusted, or
2. The Ethernet frame does not have an existing CoS value already set

Should you wish to re-configure the CoS of an incoming Ethernet frame, you should use the following command

```
Cat6500(config)# interface fastethernet 5/13
```

```
Cat6500(config-if)# mls qos cos 4
```

This command sets the CoS of incoming Ethernet frames on port 13 on module 5 to a value of 4 when an unmarked frame arrives or if the port is set to un-trusted.

7.7 Applying CoS Mutation on 802.1Q Tunnel Ports

CoS mutation is a feature available on the PFC3x that allows a "mutated" CoS value to be used on 802.1Q trunk ports for ingress queue threshold management. This feature maintains the CoS value of the incoming packet and does not change it in the same way a CoS rewrite action might take. Enabling this feature creates a map that defines what "mutated" CoS value each ingress CoS value will take. An example of a map is shown below:

Original CoS Value	Mutated CoS Value for Threshold Management
0	3
1	0
2	2
3	4
4	1
5	5
6	7
7	6

Table 16 - Example CoS Mutation Mapping

In the example map above, an ingress frame with a CoS value of 4 will take on the mutated value of 1 when placed in the queue. Should the threshold where CoS value of 1 is mapped to be exceeded, then that frame (with the original CoS value of 4) will be dropped.

This feature can only be applied to an 802.1Q trunk port whose trust setting is set to **trust-cos**. This feature is also only applicable to select linecards including the following: WS-X6724-SFP, WS-X6748-GE-TXm WS-X6748-SFP and WS-X6704-10GE. When CoS mutation is applied to a port on these modules, all ports in that ASIC port grouping also take on this feature. Port groupings are defined as follows:

Module	Number of Ports	Number of Port Groups	Port Grouping
WS-X6724-SFP	24	2	1-12, 13-24
WS-X6748-SFP	48	4	1-12, 13-24, 25-26, 37-48
WS-X6748-GE-TX	48	4	1-12, 13-24, 25-26, 37-48
WS-X6704-10GE	4	4	1, 2, 3, 4

Table 17 - CoS Mutation Port Groupings on WS-X67XX linecards

To configure CoS Mutation, the following commands must be used:

```
Cat6500(config)# mls qos map cos-mutation testmap 3 0 2 4 1 5 7 6
Cat6500(config)# interface G1/5
Cat6500(config-if)# mls qos cos-mutation testmap
```

This also assumes the following commands have been assigned:

```
Cat6500(config)# interface G1/5
Cat6500(config-if)# switchport
Cat6500(config-if)# switchport trunk encapsulation dot1q
Cat6500(config-if)# switchport mode trunk
Cat6500(config-if)# mls qos trust trust-cos
```

7.8 Configure Receive Drop Thresholds

On ingress to the switch port, the frame will be placed into a receive queue. To avoid buffer overflows, the port ASIC can implement thresholds on the receive queue and uses these thresholds to identify frames that can be dropped once those thresholds have been exceeded. The port ASIC will use the frames set CoS value to identify which frames can be dropped when a threshold is exceeded. This capability allows higher priority frames to remain in the buffer for longer when congestion occurs.

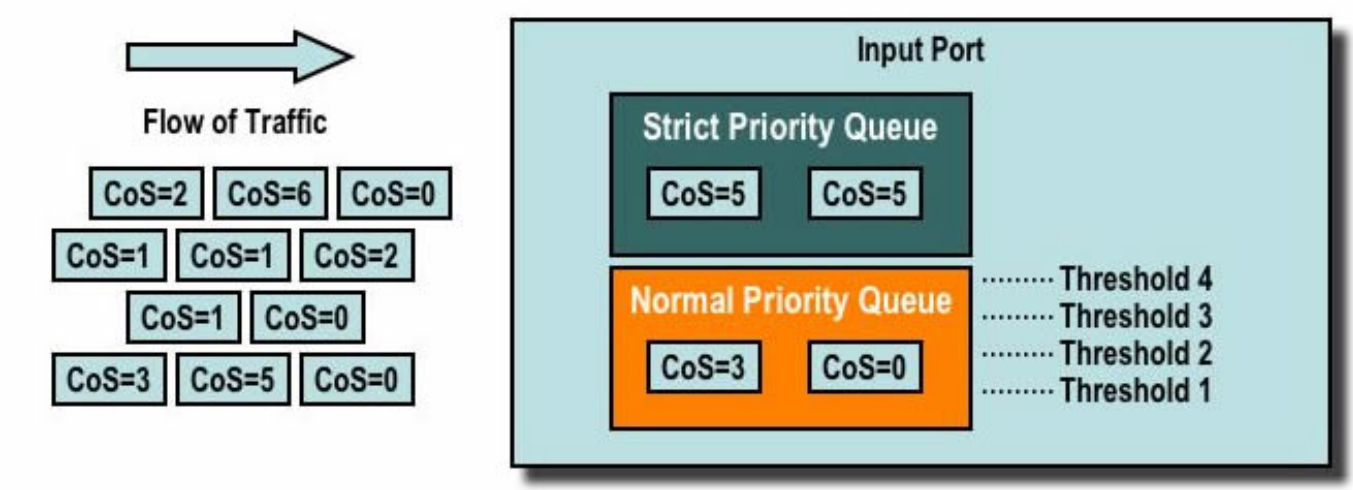

Figure 27 - Receive Drop Thresholds – Example uses 1P1Q4T threshold (on GE ports)

As shown in the above diagram, frames arrive and are placed in the queue. As the queue starts to fill, the thresholds are monitored by the port ASIC. When a threshold is breached, frames with CoS values identified by the administrator are dropped randomly from the queue. The default threshold mappings for the different queue structures are shown below.

Feature			Default Value
1Q2T Receive Queue	Threshold 1	CoS	0, 1, 2, 3, 4
		Tail Drop	80%
		WRED	Not Supported
	Threshold 2	CoS	5, 6, 7
		Tail Drop	100%
		WRED	Not Supported
1Q4T Receive Queue	Threshold 1	CoS	0, 1
		Tail Drop	50%
		WRED	Not Supported
	Threshold 2	CoS	2,3
		Tail Drop	60%
		WRED	Not Supported

	Threshold 3	CoS	4,5
		Tail Drop	80%
		WRED	Not Supported
	Threshold 4	CoS	6,7
		Tail Drop	100%
		WRED	Not Supported
1P1Q4T Receive Queue	Threshold 1	CoS	0, 1
		Tail Drop	50%
		WRED	Not Supported
	Threshold 2	CoS	2,3
		Tail Drop	60%
		WRED	Not Supported
	Threshold 3	CoS	4
		Tail Drop	80%
		WRED	Not Supported
	Threshold 4	CoS	6,7
		Tail Drop	100%
		WRED	Not Supported
	Strict Priority Queue	CoS	5
		Tail Drop	100%
1P1Q0T Receive Queue	Standard Queue	CoS	0, 1, 2, 3, 4, 6, 7
		Tail Drop	100%
	Strict Priority Queue	CoS	5
		Tail Drop	100%
1P1Q8T Receive Queue	Threshold 1	CoS	0
		Tail Drop	Disabled
		WRED	40% Low; 70% High
	Threshold 2	CoS	1
		Tail Drop	Disabled
		WRED	40% Low; 70% High
	Threshold 3	CoS	2
		Tail Drop	Disabled
		WRED	50% Low; 80% High
	Threshold 4	CoS	3
		Tail Drop	Disabled
		WRED	50% Low; 80% High
	Threshold 5	CoS	4
		Tail Drop	Disabled
		WRED	60% Low; 90% High
	Threshold 6	CoS	6
		Tail Drop	Disabled
		WRED	60% Low; 90% High
	Threshold 7	CoS	7
		Tail Drop	Disabled
		WRED	70% Low; 100% High

	Strict Priority Queue	CoS	5
		Tail Drop	100%
1Q8T Receive Queue	Threshold 1	CoS	0
		Tail Drop	50%
		WRED	Not Supported
	Threshold 2	CoS	None
		Tail Drop	50%
		WRED	Not Supported
	Threshold 3	CoS	1, 2, 3, 4
		Tail Drop	60%
		WRED	Not Supported
	Threshold 4	CoS	None
		Tail Drop	60%
		WRED	Not Supported
	Threshold 5	CoS	6, 7
		Tail Drop	80%
		WRED	Not Supported
	Threshold 6	CoS	None
		Tail Drop	90%
		WRED	Not Supported
	Threshold 7	CoS	5
		Tail Drop	100%
		WRED	Not Supported
	Threshold 8	CoS	None
		Tail Drop	100%
		WRED	Not Supported
2Q8T Receive Queue	Queue 1 Threshold 1	CoS	0,1
		Tail Drop	70%
		WRED	Not Supported
	Queue 1 Threshold 2	CoS	2,3
		Tail Drop	80%
		WRED	Not Supported
	Queue 1 Threshold 3	CoS	4
		Tail Drop	90%
		WRED	Not Supported
	Queue 1 Threshold 4	CoS	6,7
		Tail Drop	100%
		WRED	Not Supported
	Queue 1 Threshold 5-8	CoS	None
		Tail Drop	100%
		WRED	Not Supported
	Queue 2 Threshold 1	CoS	5
		Tail Drop	100%
		WRED	Not Supported
	Queue 2 Threshold 2-8	CoS	None

		Tail Drop	100%
		WRED	Not Supported
8Q8T Receive Queue	Queue 1 Threshold 1	CoS	0,1
		Tail Drop	Disabled
		WRED	40% Low; 70% High
	Queue 1 Threshold 2	CoS	2,3
		Tail Drop	Disabled
		WRED	40% Low; 80% High
	Queue 1 Threshold 3	CoS	4
		Tail Drop	Disabled
		WRED	50% Low; 90% High
	Queue 1 Threshold 4	CoS	6,7
		Tail Drop	Disabled
		WRED	50% Low; 100% High
	Queue 1 Threshold 5-8	CoS	None
		Tail Drop	Disabled
		WRED	50% Low; 100% High
	Queue 2-7 Threshold 1-8	CoS	None
		Tail Drop	Disabled
		WRED	Disabled
	Queue 8 Threshold 1	CoS	5
		Tail Drop	Disabled
		WRED	100% Low; 100% High
	Queue 8 Threshold 2-8	CoS	None
		Tail Drop	Disabled
		WRED	100% Low; 100% High

Table 18 - Per Queue Structure Default Threshold Mappings

These drop thresholds can be changed by the administrator. Also, the default CoS values that are mapped each threshold can also be changed. Different line cards implement different receive queue implementations. A summary of the queue types is shown below:

An example of this configuration capability is shown below:

```
Cat6500(config-if)# wrr-queue threshold 1 40 50
Cat6500(config-if)# wrr-queue threshold 2 60 100
```

These commands configure the 4 thresholds for a port with a queue structure of 1Q4T RX queue.

```
Cat6500(config-if)# rcv-queue threshold 1 60 75 85 100
```

This command configures for a 1P1Q4T receive queue (which applies to the new WS-X6548-RJ45 10/100 line card).

Receive drop thresholds must be enabled by the administrator. Currently the "**mls qos trust trust trust-cos"** command should be used to activate the receive drop thresholds .

7.9 Configuring Transmit Drop Thresholds

On an egress port, the port will have one of a number of transmit thresholds that are used as part of the congestion avoidance mechanism. Queue 1 is always denoted as the standard low priority queue and the higher numbered queues are denoted as the standard higher priority queue. Depending on the line cards used, they will employ either a tail drop or a WRED threshold management algorithm. Both algorithms employ two thresholds for each transmit queue.

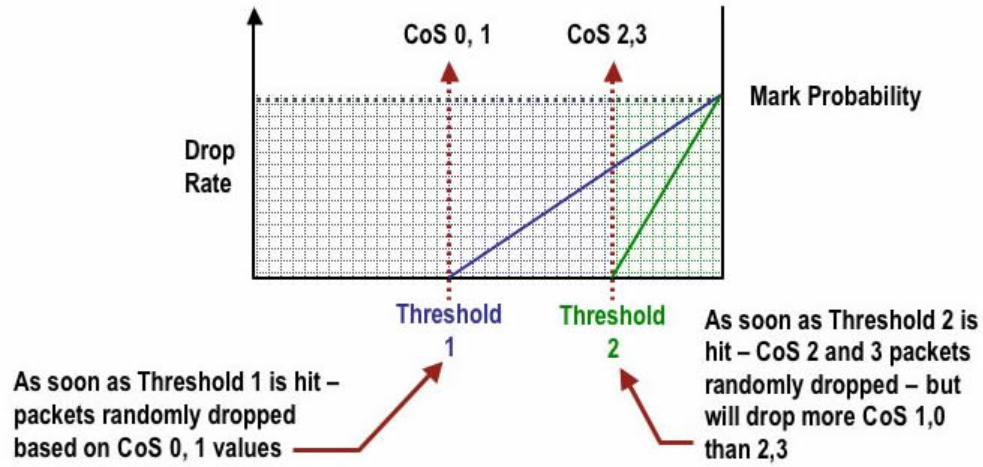

Figure 28 - Output Thresholds

The administrator can manually set these thresholds as follows:

```
Cat6500(config-if)# wrr-queue random-detect max-threshold  1 40 100
```

This sets the WRED drop thresholds for a port with a 1p2q2t queue structure 1 to 40% for threshold 1 (Tx) and 100% for threshold 2 (Tx).

WRED can also be disabled if required in Cisco IOS. The method used to do this is to use the "**no"** form of the command. An example of disabling WRED is shown as follows:

```
Cat6500(config-if)# no wrr-queue random-detect queue_id
```

7.10 Mapping CoS to Thresholds

After thresholds have been configured, the administrator can then assign CoS values to these thresholds, so that when the threshold has been exceeded, frames with specific CoS values can be dropped. Usually, the administrator will assign lower priority frames to the lower thresholds, thus maintaining higher priority traffic in the queue should congestion occur.

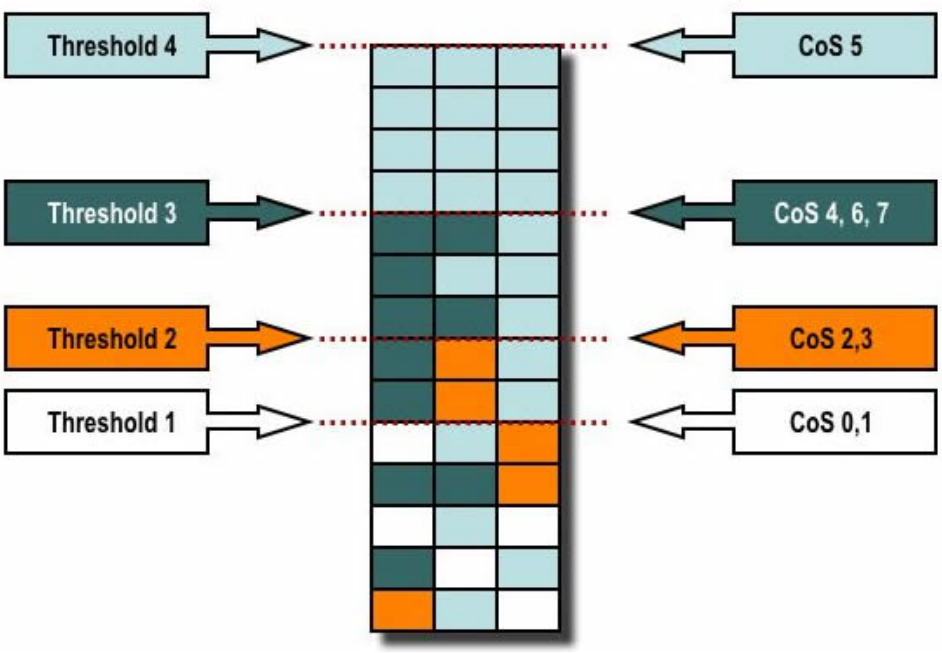

Figure 29 – Mapping CoS to Thresholds

The above figure shows an input queue with 4 thresholds, and how CoS values have been assigned to each threshold. The following shows how CoS values can be mapped to thresholds

```
Cat6500(config-if)# wrr-queue cos-map 1 1 0 1
```

7.11 Configure Bandwidth on Transmit Queues

When a frame is placed in an output queue, it will be transmitted using an output-scheduling algorithm. The output scheduler process uses Weighted Round Robin (WRR - a scheduling algorithm) to transmit frames from the output queues. Depending on the line card hardware being used, there are two; three, four or eight transmit queues per port.

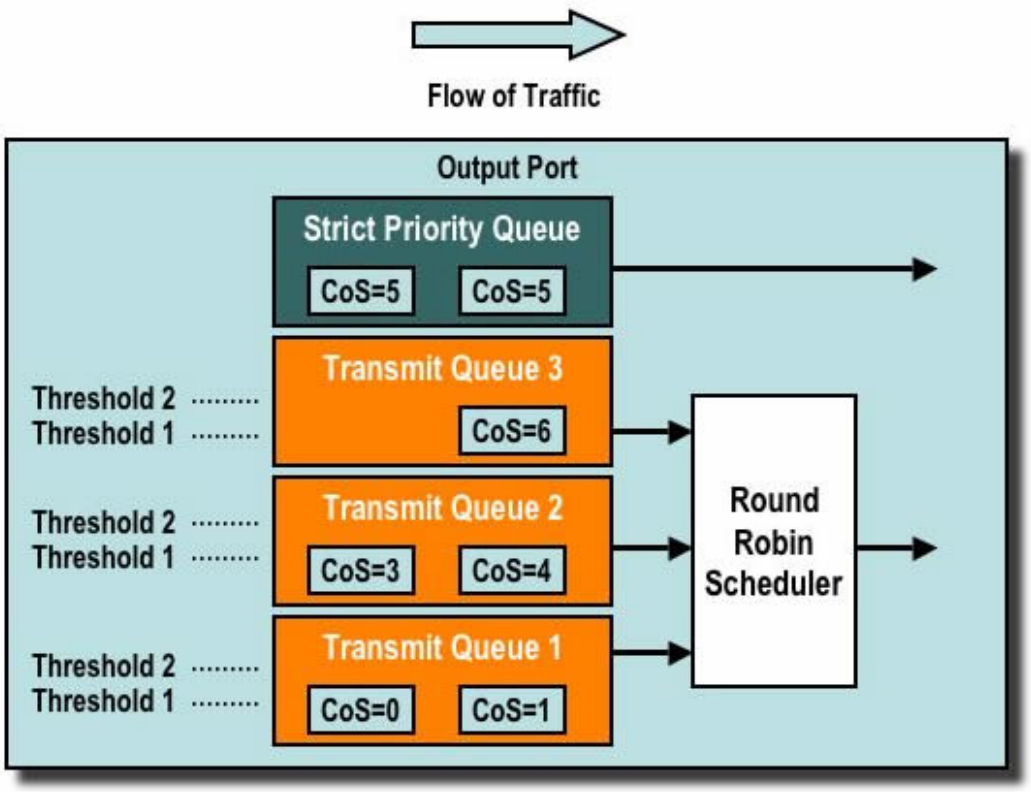

Figure 30 – Weighted Round Robin (WRR)

On the WS-X6248, 6148 and WS-X6348 line cards (with 2q2t queue structures), two transmit queues are used by the WRR mechanism for scheduling. On the WS-X6548 line cards (with a 1p3q1t queue structure) there are 4 Tx queues. Of these 4 Tx queues, 3 Tx queues are serviced by the WRR algorithm (the last Tx queue is a strict priority queue). On GE line cards (except the 67xx cards), there are 3 Tx queues however (using a 1p2q2t queue structure); one of these queues is a strict priority queue so the WRR algorithm only services 2 Tx queues. The WS-X67xx series of GE linecards differ in that they offer 8 transmit queues of which one is a strict priority queue

Typically the administrator will assign a weight to the transmit queue. WRR works by looking at the weighting assigned to the port queue, which is used internally by the switch to determine how much traffic will be transmitted before moving onto the next queue. A weighting value of between 1 and 255 can be assigned to each of the port queue.

```
Cat6500(config-if)# wrr-queue bandwidth 1 3
```

The above represents a three to one ratio between the two queues. You will notice that the Cisco IOS version of this command applies to a **specific** interface only.

7.12 Egress ACL Support for Remarked DSCP

When a frame transits the switch, classification performed by the PFC can change the ToS priority setting (IP Precedence or DSCP) in the packet. When the packet arrives at the egress port however, the default

action will result in the original ToS priority (not the PFC remarked ToS priority) that was present on the arriving packet to be used for classification purposes.

Egress ACL support for remarked DSCP is supported by all members of the PFC3 family (including the PFC3a, PFC3B and PFC3BXL). Any egress layer-3 port (either a routed port, or a layer-3 VLAN interface) can have this feature applied to it. Enabling this feature on a port is achieved using the following command:

```
Cat6500(config-if)# platform ip features sequential
```

7.13 Egress DSCP Mutation Mapping

The switch will derive an internal DSCP value from the incoming packet based on the trust setting of the port. Assuming ToS Byte preservation is not configured, this internal DSCP value is used to write the ToS value on the egress packet. Egress DSCP Mutation is a feature that allows this internal DSCP value to be changed, and the changed value be used to as the priority setting for the egress packet.

All versions of the PFC3 now support egress DSCP mutation maps that allow a set of DSCP values to be mutated (or changed) to a set value. Up to 15 maps can be defined and within each map, up to 8 DSCP values can be associated with a mutated value. An egress DSCP mutation map is configured as follows:

```
Cat6500(config)# mls qos dscp-mutation mapname 30 31 32 to 16
```

In the example above, the mutation map called “mapname” is applied to a given interface, and any packet with an internal DSCP setting of 30, 31 or 32 will have the DSCP value of 16 written into the packet prior to being transmitted.

7.14 DSCP to CoS Mapping

When the frame has been placed into the egress port, the port asic will use the assigned CoS to perform congestion avoidance (i.e. WRED) and also use the CoS to determine the scheduling of the frame (i.e. transmitting the frame). At this point the switch will use a default map to take the assigned DSCP and map that back to a CoS value. This default map is displayed in Table 3.

Alternatively, the administrator can create a map that will be used by the switch to take the assigned internal DSCP value and create a new CoS value for the frame. Examples of how you would use CatOS and Cisco IOS to achieve this are shown below.

```
Cat6500(config)# mls qos map dscp-cos 20 30 40 50 52 10 1 to 3
```

This sets DSCP values of 20, 30, 40, 50, 52, 10 and 1 to a CoS value of 3.

7.15 Adjusting the Transmit Queue Size Ratio

Each port has a set amount of buffer space allocated to it to store transient data. The amount of buffer space allocated between transmit queues on that port can be adjusted to meet set requirements of data residing in a given queue. This capability uses a percentage as a way for the administrator to define the allocated buffer space to a given transmit queue.

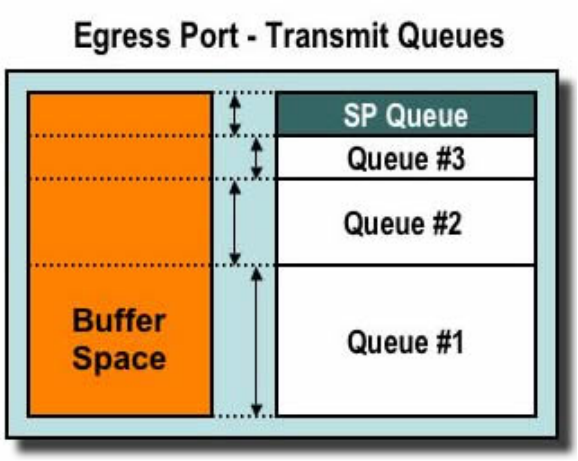

Figure 31 - Transmit Queue Size Ratio

The command to change the ratio of buffer space on the transmit queues is defined as follows:

```
Cat6500(config)# wrr-queue queue-limit 30 30 20
```

This command example above sets the transmit queue size ratio on a 1P3Q8T port setting queue #1 and queue #2 to having 30% each of the allocated buffer space. Queue #3 is set to using 20% of the allocated buffer space. The most obvious question that springs to mind is why the percentages in this example don't add up to 100%. The reason the allocated amount not adding up to 100% is due to the presence of the strict priority queue. This command does not allow the direct allocation of buffer space to the strict priority queue. Rather, the allocated amount given to the high priority queue is assigned to the strict priority queue. In this example, the high priority queue has been allocated 20% of the buffer space, so the strict priority queue will also be allocated the same amount. When this is factored into the equation, the sum of the weights now adds up to 100% (20+20+30+30).

8. Configuring QoS on the Policy Feature Card

The following section provides a guide to the configuration of QoS features on the PFC. The QoS elements that can be configured include Policing (also referred to as Rate Limiting) and Classification using Access Control Lists.

8.1 Classification and Policing with the PFC

The PFC supports the classification and the policing of frames. Classification can use an ACL to assign (mark) an incoming frame with a priority (DSCP). Policing allows a stream of traffic to be limited to a certain amount of bandwidth.

The following sections will describe these capabilities on the PFC from the perspective of both the CatOS and the Cisco IOS OS platforms.

The QoS process differs slightly between each generation of PFC. The QoS processes applied by the PFC1 and PFC2 are shown in the following diagram.

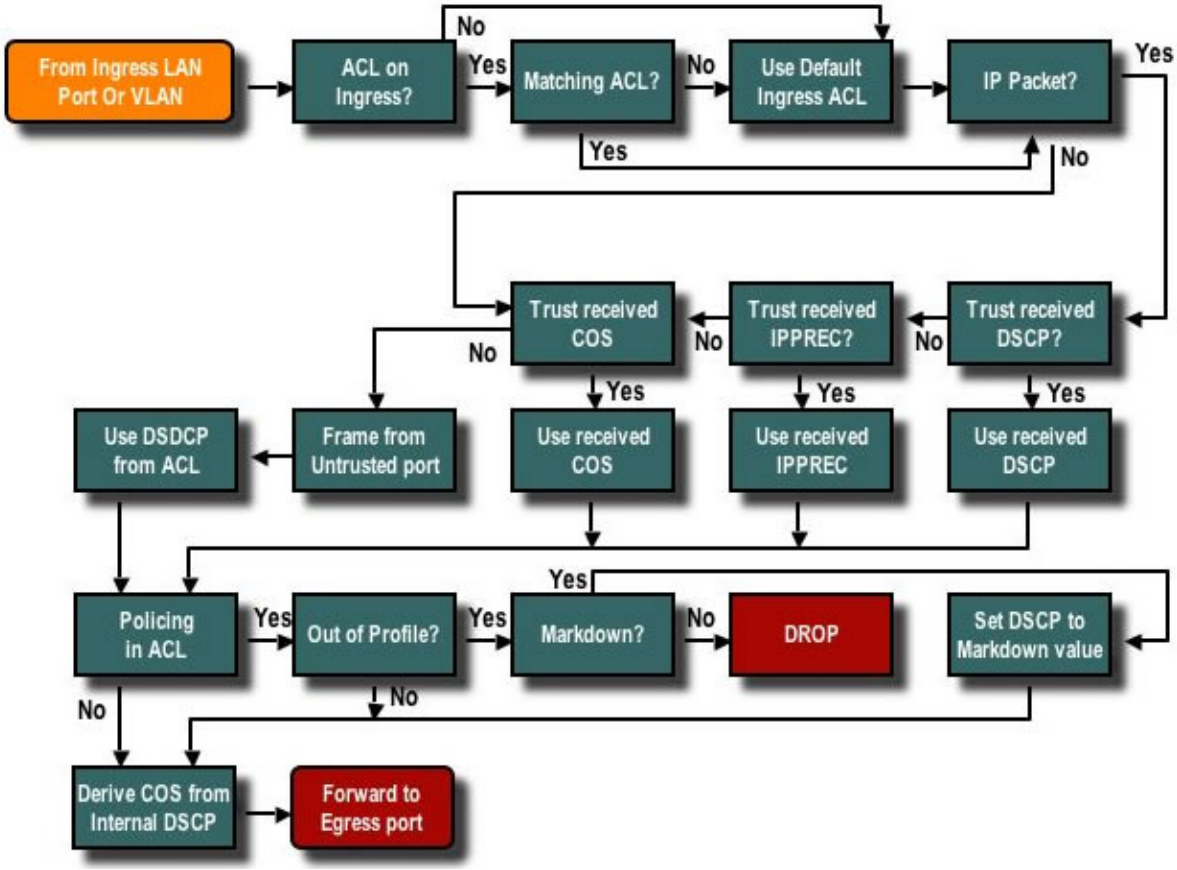

Figure 32 - QoS processing by the PFC1 and PFC2

The PFC3, however, adds egress policing into the equation, so the path taken by a packet through a PFC3 involves more steps. The processing path through the PFC3 is shown in the following diagram.

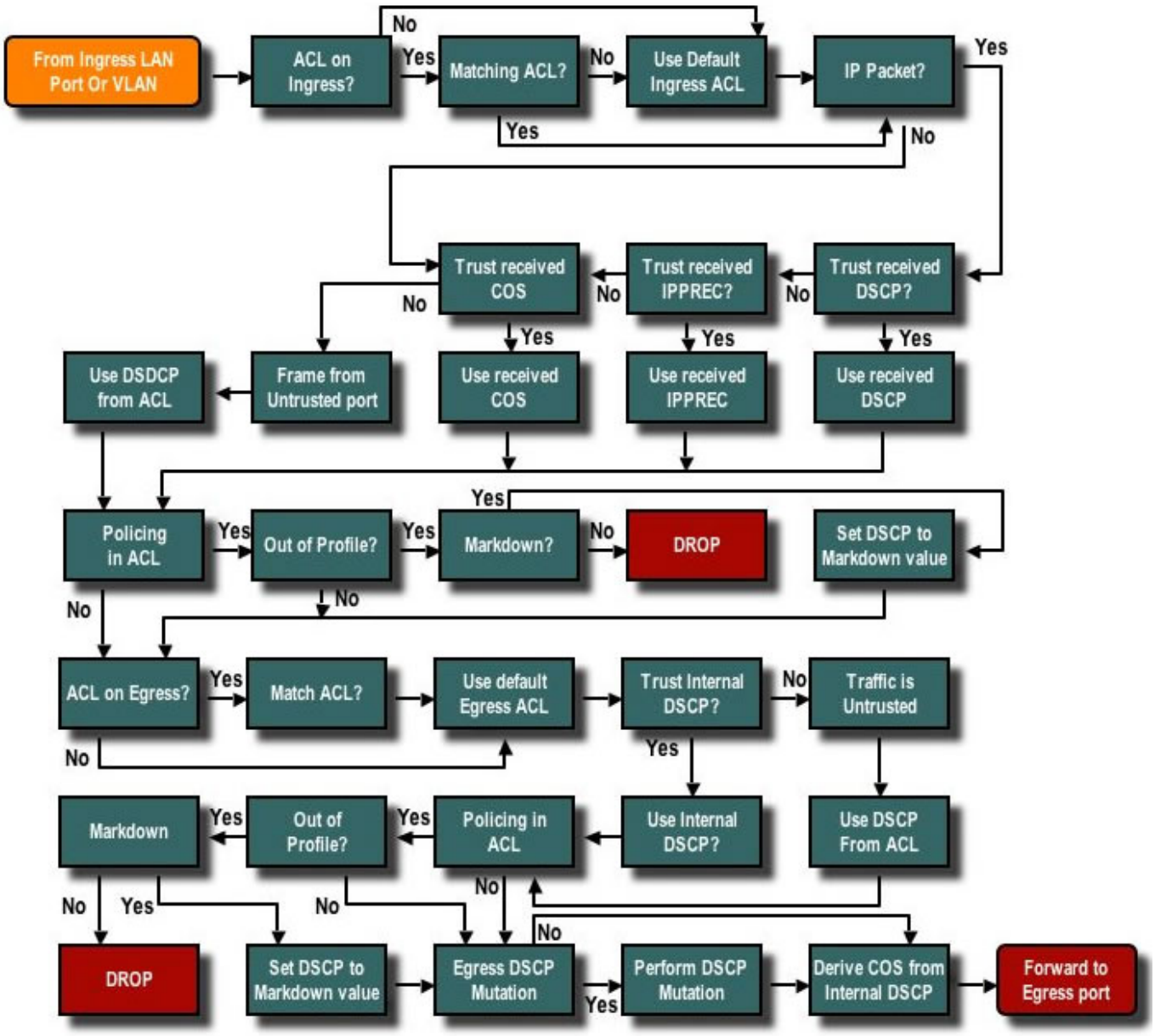

Figure 33 - QoS Processing for the PFC3

The differences between the policing capabilities of the different Supervisors (Policy Feature Card's) are summarised in the following table.

	Sup1A	Sup2	Sup32	Sup720	Sup720-3B	Sup720-3BXL
Default PFC	PFC1	PFC2	PFC3B	PFC3A	PFC3B	PFC3BXL
Ingress Policing	Yes	Yes	Yes	Yes	Yes	Yes
Egress Policing	No	No	Yes	Yes	Yes	Yes
Aggregate Policing	Yes	Yes	Yes	Yes	Yes	Yes
Microflow Policing	Yes	Yes	Yes	Yes	Yes	Yes

User Based Rate Limiting	No	No	Yes	Yes	Yes	Yes
What is counted by the Policer	Length of Layer 3 packet	Length of Layer 3 packet	Length of Layer 2 packet	Length of Layer 2 packet	Length of Layer 2 packet	Length of Layer 2 packet
Leaky Bucket Implementation	Single	Dual	Dual	Dual	Dual	Dual

Table 19 - Policing capabilities of different Supervisor Engines

There were a few key policing enhancements that were introduced with the PFC3x family (PFC3a, PFC3B and PFC3BXL). The first (and most probably major) enhancement the PFC3x provided was support for Egress Policing which is a policing policy that is applied to outbound traffic leaving the switch. While Ingress Switching can be applied to a physical Layer 2 or Layer switch-port, a Routed port or a Switched Virtual Interface (i.e. VLAN interface), Egress policing can apply a policy to all of these interfaces except a physical Layer 2 switch-port. An Egress Policer cannot be applied to a Layer 2 switch-port as it can with Ingress Policing. When the Policy Feature Card performs both Ingress and Egress policing, it will process ingress policer before egress policer. It is also worth noting that an Ingress and Egress policer can exist on a physical interface (or VLAN interface) at the same time. Egress Policing will be discussed in more detail later in this paper.

Another key difference was the way in which the PFC3x and its policing algorithm counted data. Both the PFC1 and PFC2 only counted the data portion of the packet towards the stated limit. So for instance, if a full sized Ethernet packet were sent (1518 byte packet = 1500 bytes of data + the 18 Byte Ethernet Header), only 1500 bytes would be counted towards the stated rate. The PFC3x, however, counts both the Header and Data portion of the packet. Using the above example, a full size packet would be counted by the policer as 1518 bytes. In this way, the PFC3x will arrive at its policed rate more quickly than a PFC1 or PFC2. This needs to be factored into the policing calculation by any customer who has existing policing polices defined should they migrate to a PFC3x based Supervisor.

User Based Rate Limiting (UBRL) is a third key difference from earlier PFC's. User Based Rate Limiting is a form of Microflow policing and provides the benefit in that it allows two flow masks to be used in the system at the same time. Both UBRL and Microflow policing will be explored in more detail later in this paper.

8.2 Configure Policing on the Catalyst 6500

Policing is supported with Cisco IOS, however, the configuration and implementation of the policing function is achieved using **policy maps**. Each policy map can use multiple **class maps** to make up a policy map and these policy classes can be defined for different types of traffic flows. Policing is configured on the Catalyst 6500 using the Modular QoS CLI (MQC). Up to 255 class maps can be defined per policy map with a total of 1024 class maps per Catalyst 6500 chassis. MQC is provided in IOS to allow the separation of class maps from policy maps and from the application of these on an interface. It provides the mechanisms to configure all Policing and Classification features available in the switch.

It is important to note that on the Catalyst 6500, QoS parameters available in Router IOS are not necessarily available. Even with the presence of some of these commands are there in the CLI, does not necessarily mean they are supported on all interfaces. Some MQC QOS commands are applicable only to

routed ports on a FlexWAN or SIP linecard (for instance) yet are not supported on standard Ethernet linecards. Do note that any queuing, congestion management or scheduling features are configured on a port basis and not within the policy map.

Policy map classes use classification criteria as the means to identify which packets this policy will be applied to. The classification aspect of this process use IOS based access control lists and class **match** statements to identify traffic to be policed. Once the traffic has been identified, the policy classes can use aggregate and Microflow policers to apply the policing policies to that matched traffic.

8.3 Rate and Burst

Two key parameters used implicitly in the configuration of policing are the Rate and Burst. The rate (also referred to as the Committed Information Rate – or CIR) is defined as the maximum amount of data that can be forwarded in a given interval (normally referred to in Kbps or Mbps). The burst can be thought of as the total amount of data that can be received in a given interval. A simple example of how these parameters interact could be a policy that uses a Rate of 10Mbps and a Burst of 15Mbps. This policy defines that a maximum of 15Mbps can be received in a given interval from which 10Mbps (the Rate) of data can be sent. Relating this back to the bucket concept, the Burst can be thought of as the depth of the bucket (how much data can be received), while the Rate can be thought of as a hole in the bucket defining how much data can be leaked out of the bucket (or forwarded).

An important point to stress is that the Burst should "NEVER" be less than the stated Rate. If, for argument sake, the Burst were set at 8Mbps, then a Rate of 10Mbps would be impossible to achieve. If the bucket can only ever hold 8Mb, the maximum Rate could only ever be 8Mbps as well.

8.4 PIR and Max Burst

PIR (Peak Information Rate) and the Max Burst are the next set of parameters that must be understood. While the Rate and Burst are associated with the first bucket, the PIR and Max Burst are associated with the second bucket. The Max Burst defines the depth of the second bucket, while the PIR is the amount of data that can be forwarded from second bucket. The use of PIR and Max Burst are parameters associated with policing in a Native IOS implementation. CatOS uses slightly different terminology, namely ERate and EBurst. ERate is the equivalent of PIR and EBurst is the equivalent of Max Burst.

8.5 Hardware Interval

The use of the term "given interval" above relates to a specific hardware defined interval built into the Catalyst 6500 Policy Feature Card hardware. A fixed interval of $1/4000^{th}$ of a second, or 0.25 milliseconds is used in the policing calculation. The hardware interval bears relevance to the replenishment rate of tokens into the token bucket. This is discussed later in this paper. Another key difference is that while Rate is specified in bits, the burst is specified in bytes (**NOT BITS)**.

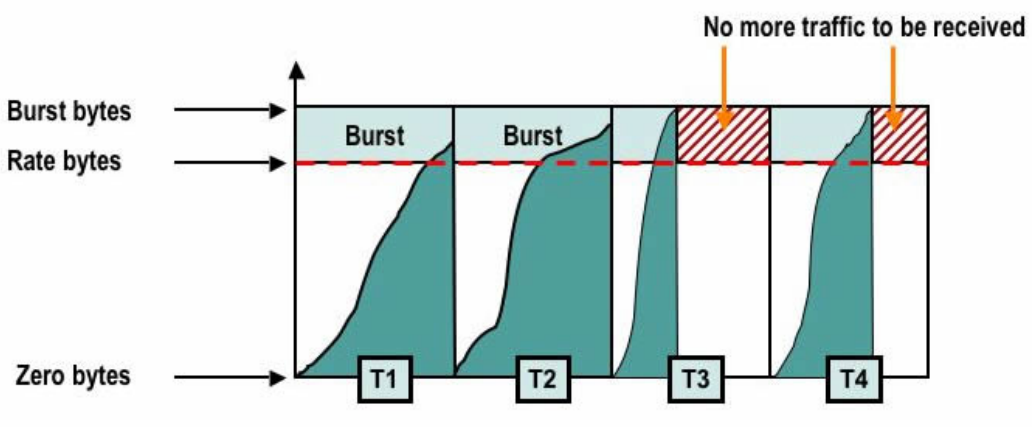

Figure 34 - Policing Per Interval

As can be seen in the above diagram, data begins to arrive at the switch in a given interval and the data will be forwarded as long as the total count of data is within the defined Rate (for that interval). Once the data count for a given interval exceeds the Rate limit, the data will not be forwarded until the next hardware interval starts.

8.6 Aggregates and Microflow's

Aggregates and Microflow's are terms used to define the scope of policing that the PFC performs. A Microflow defines the policing of a single flow. The flow mask installed in the system defines how the Microflow Policer views a flow. Typically the default flow mask for Microflow policing is defined as a session with a unique SA/DA MAC address, SA/DA IP address and TCP/UDP port numbers or as in the case of a PFC3 it can be a source IP address. For each new flow that is initiated through a port or a VLAN, the Microflow can be used to limit the amount of data sent or received for that flow by the switch. In the Microflow definition, packets that exceed the prescribed rate limit can be either dropped or have their DSCP value marked down. Microflow's are applied using the **police flow** command that forms part of a policy map class. Another important aspect of a Microflow policer is that it can only be applied to ingress traffic. It cannot be applied to egress traffic.

To enable Microflow policing in Cisco IOS, it must be enabled globally on the switch. The following command is applicable only for the PFC1. This can be achieved using the following command:

```
Cat6500(config)# mls qos flow-policing
```

Microflow policing can also be applied to bridged traffic that is traffic that is not Layer 3 switched. To enable the switch to support Microflow policing on bridged traffic, this too must be enabled globally on the switch using the following command: As noted, this is not needed for PFC2 or PFC3.

```
Cat6500(config)# mls qos bridged
```

This command also enables Microflow policing for Multicast traffic. If Multicast traffic needs to have a microflow policer applied to it, then this command (mls qos bridged) must be enabled.

Microflow policing can also be used with the PFC3x to rate limit multiple microflows to the same limit based on source or destination address with a single ACL (this is described further in the User Based Rate Limiting section).

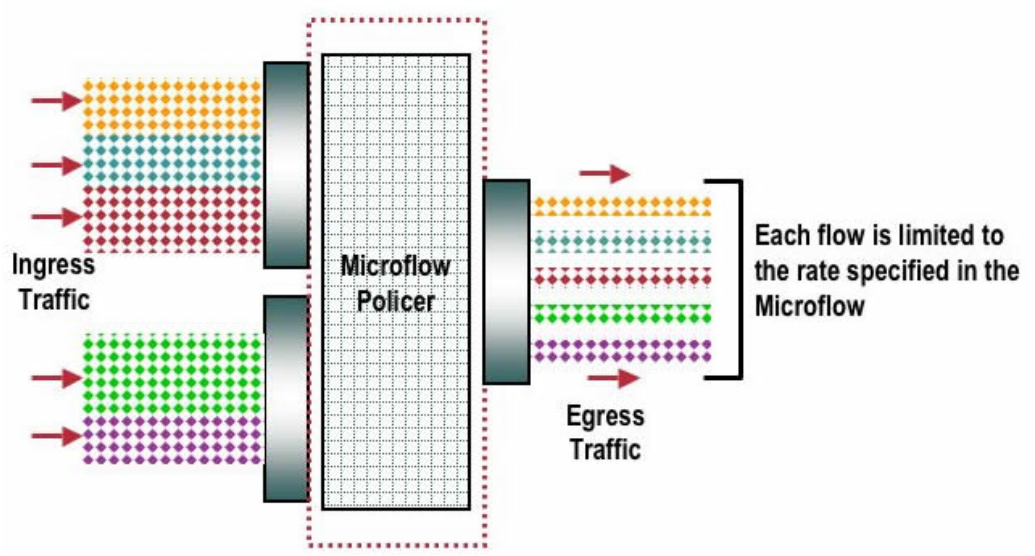

Figure 35 - Microflow Policer

Like a Microflow, an Aggregate can be used to rate limit traffic; however, the Aggregate rate applies to ***all*** traffic inbound on a port or VLAN that matches a specified QoS ACL. The Aggregate Policer can be applied to either a physical interface or a VLAN. If an Aggregate policer is applied to a single interface, then the Aggregate Policer will count all matching traffic (that matches the classifying ACL) coming into the interface towards the policer. If the Aggregate Policer is applied to a VLAN, then all of the matching traffic coming in any of the ports in that VLAN is counted towards the stated Rate. An example of an aggregate could be a 20Mbps Aggregate policer applied to VLAN 10. In VLAN 10 there are five ports (shown on the left hand side of the diagram below). This aggregate would limit the total amount of traffic for those five ports to 10Mbps.

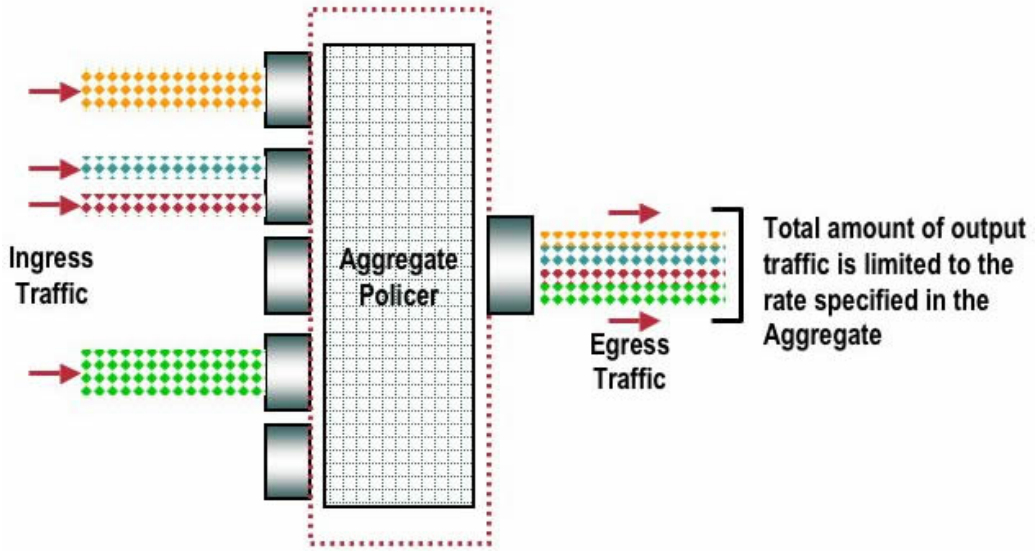

Figure 36 - Aggregate Policer

There are two forms of aggregates that can be defined in Cisco IOS, and those are

1. Per interface aggregate policers, and

2. Named aggregate policers or shared aggregate

Per interface aggregates are applied to an individual interface using the **police** command within a policy map class. These map classes can be applied to multiple interfaces, but the policer polices each interface separately. Named Aggregates are applied to a group of ports and police traffic across all interfaces cumulatively. Named aggregates are applied using the **mls qos aggregate policer** command.

There are system limitations set in place that one needs to be mindful of when implementing policing. When defining Micro-flows, the number of flows you can police is limited by the number of flows supporting in the Netflow Table on the PFC. For the PFC1, PFC2, PFC3 and PFC3B you can rate limit up to 128K+ unique flows using up to 63 different rate + traffic class combinations. The PFC3BXL allows up to 256K flows to be policed. For Aggregate policers, there is a system limit of 1023 policers and this applies across all PFC versions.

8.7 The Rules of the Token Bucket

The Token Bucket is used to determine if a packet can be sent (i.e. data is in profile) or dropped (i.e. data is out of profile). The Token Bucket is a virtual bucket with a set of tokens. A token bucket exists for each aggregate policer you create. There are 1023 aggregate policers that can be defined on a Catalyst 6500. When a Microflow policer is created, a token bucket is created for each flow that is monitored by the Microflow policer. The Catalyst 6500 can support 63 Microflow policers at the same time. This does not mean 63 flows, but 63 different Microflow policing rates. For each Microflow policer, a separate token bucket is kept for each flow that is matched by the policer. If a user were to start up an email session, web session and an FTP session and those sessions match the classification criteria set by the Microflow policer, then a separate token bucket would exist for each of the flows created from those sessions.

The PFC2 introduced the concept of dual rate token bucket policing. This is also supported in all versions of the PFC3. A dual token bucket approach allows for a standard rate and burst to be defined along with an extended rate and burst. An example of a dual rate policer is given later in this paper. It is important to note, however, that dual rate policing is only supported for aggregate policers. Microflow policers operate with a single token bucket.

In order for data to be forwarded by the policer, a set of tokens must exist in the token bucket. Each token amounts to the policer being able to send one bit of data. If a 512-byte packet arrives within the hardware interval of 0.25ms, then 4096 tokens will be required to be in the bucket to send the packet. The requirement of 4096 tokens is based on the number of bits in the packet (i.e. 512 bytes x 8 bits = 4096). If only 4095 tokens are in the bucket then the packet cannot be sent (it is dropped) and the tokens remain in the bucket. When a packet is sent, the tokens are removed from the bucket.

The depth of the bucket is another important factor to consider. The Burst should be set so that it can hold the largest sized packet in the data flow being policed. In IOS, the minimum defined burst size is 1000. The number 1000 equates to a burst size of 1 kilobyte or the buckets ability to hold 8000 tokens (i.e. 1 kilobyte = 8000 bits). If this value were used, and the data flow consisted of maximum size Ethernet packets (1518 byte), then no packet could ever be sent as there would never be enough tokens in the bucket to forward a packet in each interval. The Burst should be set to a value that ensures the switch can at least sustain the configured Rate. If a burst value is defined in the policer, the minimum set will be

equal to the rate in a 4-millisecond interval. For example, if a 10Mbps rate were specified, then the minimum burst would be 5000 which is calculated as follows:

10,000,000 bits x 0.004 seconds / 8 bits = 5000

This is confirmed showing the following CLI where a lower burst of 4000 is configured:

```
c6500(config-pmap-c)# police 10000000 4000 conform transmit exceed drop
Info: Illegal normal burst size, increased to 5000
```

If no burst is specified, then the default burst will be calculated at 250ms. In the above example, this would set the burst to 312500 which is calculated as follows:

10,000,000 x 0.25 seconds / 8 bits = 312500

This can be confirmed as shown by the following CLI

```
c6500(config-pmap-c)# police 10000000 conform transmit exceed drop
c6500(config-pmap-c)#^Z
c6500# show policy
  Policy Map limit_to_10Mb
    Class marketing
      police 10000000 312500 312500 conform-action transmit exceed-action drop
```

Note the resulting burst value in the show output. As a fallback, always remember that the CLI will reset your configured Burst size if it is too small for the configured Rate.

8.8 A Walk Through how the Token Bucket is used

To better explain how the token bucket works, a simple example will be explored below.

Step 1 At the beginning of the time interval (call it T0), the token bucket is primed with a full complement of tokens. This is always the case whenever a policer is activated.

Step 2 Data will start to arrive at the switch. The policer will only process data that fits within the classification criteria and arrives in the given 0.25ms interval. Any data that has not fully clocked in within that interval will be processed in the next interval.

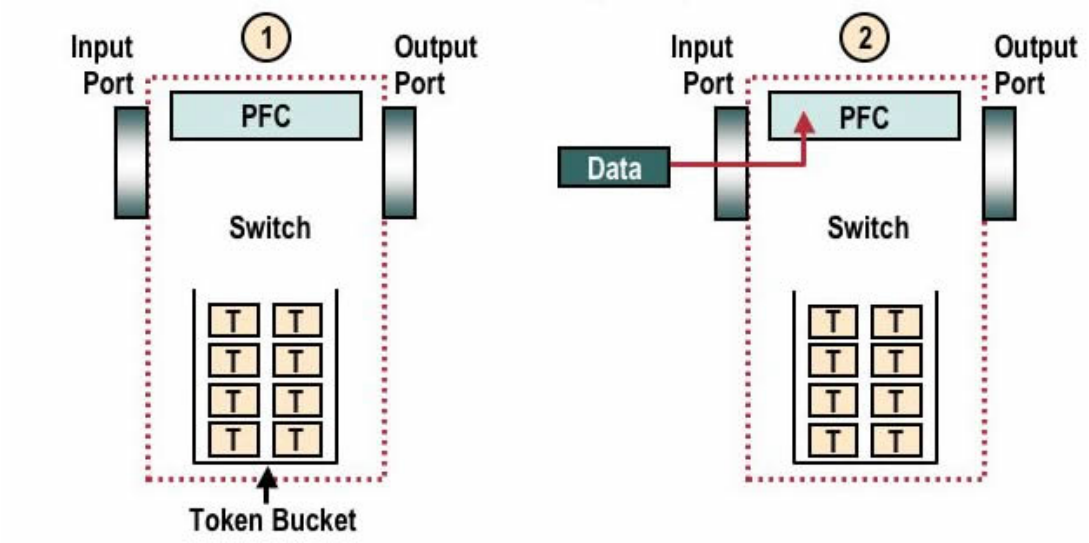

Figure 37 - Step 1 and 2 of the Token Bucket Process

Step 3 and 4 The PFC will inspect the number of tokens that are available in the token bucket. If the number of tokens is greater than or equal to the number of bits in the current packet being inspected, then the packet can be forwarded. If there are not enough tokens, then the packet will either be marked down (and forwarded) or dropped, depending on the policer's configuration.

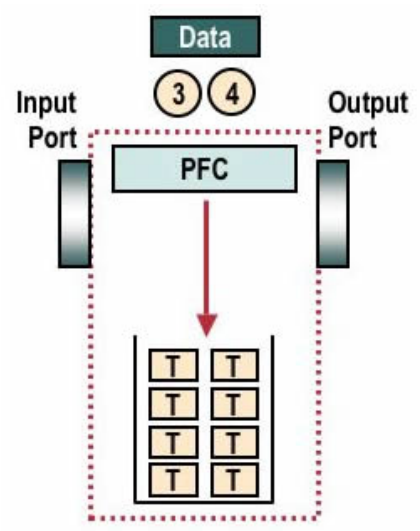

Figure 38 - Steps 3 and 4 of the Token Bucket Process

Step 5 Assuming that there are enough tokens in the bucket, a number of tokens (corresponding to the number of bits in the packet) will be removed from the token bucket.

Step 6 The packet will then have the "conform-action" as configured in the policing statement applied to it. Normally the "conform" action is simply to forward the packet, which is performed by the Policy Feature Card. Other packets that had arrived in that time interval will have the same action applied to them. If enough tokens exist in the bucket for those packets, the packets will be sent.

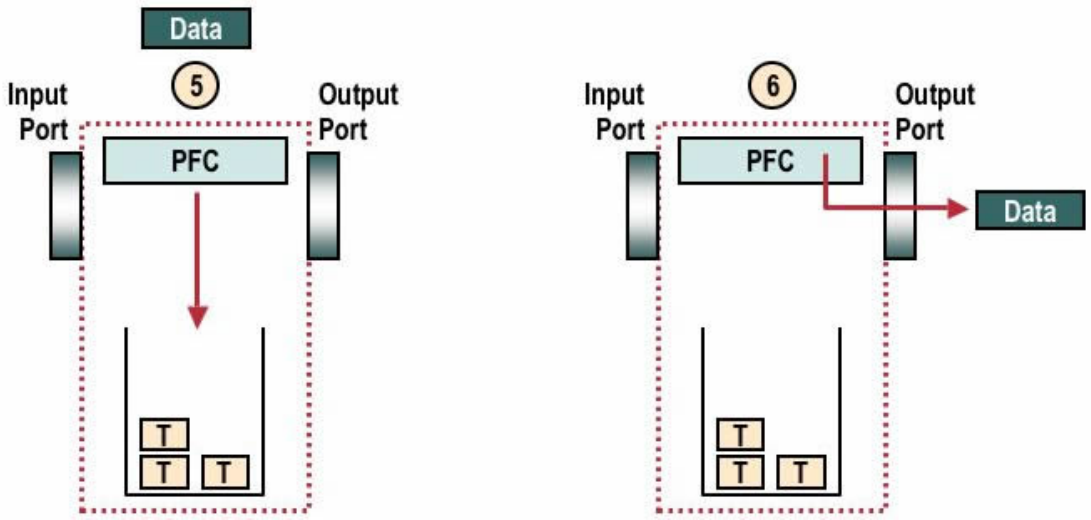

Figure 39 - Steps 5 and 6 of the Token Bucket Process

Step 7 At the end of the time interval, the token bucket is primed with a new complement of tokens. The number of tokens that are added to the bucket is calculated as the Rate divided by 4000.

Step 8 So begins the next time interval and the same process continues. For each packet that arrives in that given interval, if enough tokens exist, then the packet will be sent and the corresponding amount of tokens removed from the bucket. If there are not enough tokens, then the packet will be marked down (and forwarded) or dropped.

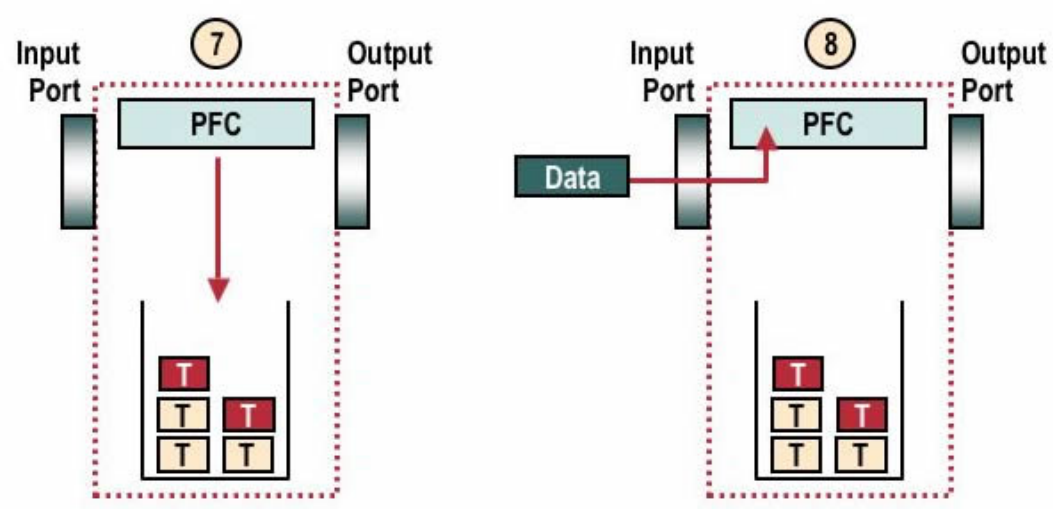

Figure 40 - Step 7 and 8 of the Token Bucket Process

More details on this process will be provided in the next section.

8.9 A Simple Policing Example

Lets start with a simple example and assume a policing policy has been made. For a given Fast Ethernet interface (100Mb) the traffic is to be limited to 10Mbps. The definition of this policy implies the use of an Aggregate Policer. The following example steps through the process of building a policing policy on the Catalyst 6500 running Native IOS.

The first step is to define a class map. A class-map is used to build a set of classification criteria that will determine what data will have this policer applied to it. The class-map uses a match statement that points to a pre defined access list (in our case access list 101). This access list defines an Access Control Entry (ACE) to match any traffic heading to host 10.1.1.1. This class map is shown below.

```
6500(config)# access-list 101 permit ip any host 10.1.1.1
6500(config)# class-map identify-server-traffic
6500(config-cmap)# match access-group 101
```

Once the class-map is defined, the policy map can then be built as follows.

```
6500(config)# policy-map police-to-10
6500(config-pmap)# class identify-server-traffic
6500(config-pmap-c)# police 10000000 5000 confirm-action transmit exceed-action drop
```

This configuration set defines a policy map named “police-to-10”. Within this policy map is a class map (defined in the previous step), which is used to provide the classification criteria. Up to 255 class maps can exist within a single policy map. Within the class map is the policer statement. This policer configures a normal rate of 10,000,000 bits per second (this equates to 10Mbps). The Burst defines a token bucket with a depth of 5,000 bytes (5000 bytes x 8 bits = 40,000 tokens). If a burst value is set that is lower than what can be sustained via the rate, an error message will be generated, and the system will reset the burst to the minimum allowed for that rate. An example of this is shown below.

```
6500(config-pmap-c)# police 10000000 1000 confirm-action transmit exceed-action drop
Info: Illegal normal burst size, increased to 5000
```

The “conform-action” defines the action to take for traffic that is within the normal rate (10Mbps). In this case, the action is configured to forward the traffic. The “exceed-action” defines what happens to data that is in excess of the normal rate. Data that exceeds the 10Mbps rate will be dropped.

After the policer has been created within the policy map, it needs to be applied to an interface. This is achieved using the service policy command as shown in the following example.

```
6500(config)# interface F3/1
6500(config-if)# service-policy input police-to-10
```

The keyword “input” defines the direction that the policer will be applied. In the example above, an input policer has been defined. If an egress policer were to be configured, then the “output” keyword would be used in place of “input”.

To see how this policer works, a table will be used to better explain what happens when the policer starts. The Policer assumes the following environment.

- Single Ingress Aggregate Policer applied to a Fast Ethernet Interface.
- Traffic arrives at a rate of 100Mb/sec (fully utilizing the FE interface).
- 100Mb arrival rate equates to 25,000 bits per interval (10,000,000 / 4,000)
- Policer is set to rate limit traffic to 10Mb/sec

- A constant stream of 64 byte packets arrives at the interface.
- PFC3B is used, so L2 Header plus Data is counted.
- Burst set to 5,000.
- Conforming traffic will be transmitted as per configuration statement.
- Traffic exceeding the rate will be dropped as per configuration statement.
- Token Rate Replenishment for each interval is calculated as 10Mb / 4000 = 10,000,000 / 4000 = 2500

The following table provides an insight into how this policer works.

Time Interval	Bits clocked in interval	Tokens at start of interval	How many packets can be sent	Number of bits that are forwarded	Number of bits not forwarded	Tokens left over at end of interval
T0	25000	40,000	48	24,576	424	15,424
T1	25000	17,924	35	17,920	7080	4
T2	25000	2504	4	2048	22952	456
T3	25000	2946	5	2560	22440	386
T4	25000	2836	5	2560	22440	276
T5	25000	2776	5	2560	22440	216
T6	25000	2716	5	2560	22440	156
T7	25000	2656	5	2560	22440	96
T8	25000	2596	5	2560	22440	36
T9	25000	2536	4	2048	22952	488
T10	25000	2988	5	2560	22440	428

Table 20 - Policer in Action

As the table shows, at time interval T0, the bucket has a full complement of tokens. The initial stream of packets arrives and 48 packets can be sent. The 48 packets equates to 24,576 bits (48 packets x 512 bits per packet), so 24,576 tokens are removed from the bucket leaving 15,424 from the original 40,000 tokens left for the next interval. At time interval T1, another 2500 tokens are added to the bucket. The number of packets forwarded this time is reduced from the previous interval, as there are fewer tokens in the bucket. One interesting point is in time interval T9 where the forwarded packet count drops to 4 packets, however, this recovers to 5 packets in the next interval. A pattern would then become established with roughly 5 packets being sent every 7 or 8 intervals and 4 packets being sent every 8^{th} or 9^{th} interval.

Statistically over a period of time, the number of packets sent should closely equate to the defined Rate. If it is assumed that for every 9 intervals we send 8 lots of 5 packets and one lot of 4 packets (using a rough average from the table above), this would equate to a final Rate of just over 10Mb every second ([4000 / 9] x [[8 x 5 x 512] + [1 x 4 x 512]] = 10,002,242 bits =~ 10Mbps).

8.9.1. DSCP Markdown Maps

DSCP Markdown maps are used when the policer is defined to markdown out of profile traffic instead of dropping it. Out of profile traffic is defined as that traffic that exceeds the defined burst setting.

A default DSCP markdown map is set up when QoS is enabled. This default markdown map is listed in Table 3 earlier in the document. The CLI allows an administrator to modify the default markdown map using the **set qos policed-dscp-map** command. An example of this is shown below

```
Cat6500(config)# mls qos map policed-dscp normal-burst 32 to 16
```

This example defines a modification to the default policed DSCP with a Rate of 1Mbps is applied to a switchport. A user on this port starts three different applications (for example, web, email and a telnet session) and each of these applications created a single flow. The result of the Microflow policer would be that only 3Mb of traffic would be allowed through this port. If the user then started another Web window that started four flows, then the total traffic allowed would be 7Mbps (i.e. 1Mbps x 7 flows). With a Microflow, it is not the total amount of traffic that is limited, rather the amount of traffic that **each flow** is allowed to use.

8.10 User Based Rate Limiting (UBRL)

User Based Rate Limiting is a form of Microflow Policing. As mentioned above, Microflow policing supports the policing of individual flows. The switch uses a flow mask to determine what constitutes a flow. There are a number of different flow masks available in the PFC1 and PFC2, and they include:

- Destination Only IP (this is the default)
- Source and Destination IP
- Full Flow (Source and Destination IP, Protocol and Source and Destination Port)

When Microflow policing is enabled, full flow masks are used. This means that a Microflow policer will apply to each flow with a unique source/destination IP address and a unique source/destination port number. To explain this further, lets assume a Microflow policer with a Rate of 1Mbps is applied to a switchport. A user on this port starts three different applications (for example, web, email and a telnet session) and each of these applications created a single flow. The result of the Microflow policer would be that only 3Mb of traffic would be allowed through this port. If the user then started another Web window that started four flows, then the total traffic allowed would be 7Mbps (i.e. 1Mbps x 7 flows). With a Microflow, it is not the total amount of traffic that is limited, rather the amount of traffic that **each flow** is allowed to use.

The PFC3x introduced support for multiple flow masks to be used at the same time. It also added support for a range of new flows masks including the following:

- **Destination:** Destination IP Address
- **Source:** Source IP Address
- **Source & Destination:** Source and Destination IP Address
- **Interface, Source & Destination:** Input interface, source and destination IP address
- **Full:** Source, Destination IP address, IP protocol, TCP/UDP source and destination ports if present
- **Interface, Full:** Input interface, Source, Destination IP address, IP protocol, TCP/UDP source and destination ports if present

Of these flow masks mentioned above, it is with the combination of the Destination Only and Source Only IP flow masks where User Based Rate Limiting comes into play. When these flow masks are used,

it changes the way in which the system views a flow. In our example above where a single user initiates a number of applications that created a number flows, UBRL would now view all those flows as a single flow. Why is this so? It is because the flow mask in use (source-only) will only be interested in the source IP address as the flow identifier. As all of the applications are sourced from the same IP address in our example, the Microflow would view “ALL” traffic from that user as a single flow. This means that in our example of the 1Mb Microflow being applied to that switchport, the user would rate limited to 1Mb for all traffic originating from that IP address. More importantly, in the situation where the “full flow mask” was being used, there would have been seven flow records created in the Netflow table. With the “source-only” flow mask being used, only a single flow record now exists in the Netflow table. UBRL thus allows us to scale the Netflow table to support more flow records.

It also added support for a Source Only IP flow mask to the flow masks already supported. This is where User Based Rate Limiting comes into play. When a user configures User Based Rate Limiting, they will be using the source-only and destination-only flow masks. When these flow masks are used, it changes the way in which the system views a flow. In our example above where a single user initiates a number of applications that created seven flows, UBRL would now view those 7 flows as a single flow. Why is this so? It is because the flow mask in use (source-only) will only be interested in the source IP address as the flow identifier. As all of the applications are sourced from the same IP address in our example, the Microflow would view “ALL” traffic from that user as a single flow. This means that in our example of the 1Mbps Microflow being applied to that switchport, the user would rate limited to 1Mbps for all traffic originating from that IP address. More importantly, in the situation where the “full flow mask” was being used, there would have been seven flow records created in the Netflow table. With the “source-only” flow mask being used, only a single flow record now exists in the Netflow table. UBRL thus allows us to scale the Netflow table to support more flow records.

When a UBRL policy is defined, the flow mask that is used with this policy is defined along with the stated rate. The flexibility UBRL yields is in allowing one policer to be applied for traffic outbound and another for return traffic as shown in the following example.

First, create the ACL’s to permit the traffic sourced from the subnets to any destination address and for return traffic.

```
Cat6500(config)# access-list 101 permit ip 10.0.1.0 0.0.0.255 any
Cat6500(config)# access-list 102 permit ip any 10.0.1.0 0.0.0.255
```

Then, you need to add this ACL to the class-map, and then match the class-map to the appropriate access-group:

```
Cat6500(config)# class-map outward_traffic
Cat6500(config-cmap)# match access-group 101
Cat6500(config)# class-map return_traffic
Cat6500(config-cmap)# match access-group 102
```

Once that’s complete, the policy-map must be created to set the rate limits for the users in the access list, and to configure what to do with the traffic if the policy is not adhered to:

```
Cat6500(config)# policy-map UBRL
Cat6500(config-pmap)# class outward_traffic
```

```
    Cat6500(config-pmap-c)# police flow mask src-only 20000000 13000 conform-action
     transmit exceed-action drop
  Cat6500(config-pmap)# class return_traffic
    Cat6500(config-pmap-c)# police flow mask dest-only 30000000 13000 conform-action
     transmit exceed-action drop
```

This statement created a rate limits for outbound traffic of 20Mbps with a burst of 52Mbps (13000*4000 = 52Mb), and return traffic for 30Mbps with a burst of 52Mbps. If traffic matches this profile, and be within the rate limit, the action to *transmit* the traffic is set with the **confirm-action** statement. Should traffic exceed the rates of 20 and 30Mbps, the action to *drop* additional traffic is set with the **exceed-action** parameter.

8.10.1. Egress Policing

Egress policing is only supported by the PFC3. The PFC3 supports egress policing of traffic using both IP and MAC based ACL's. The egress policing of IPX traffic is supported by the PFC3 but with MAC ACL's. Egress policing can only be applied to a routed (layer 3) interface or a VLAN (SVI) interface and is not permitted on a layer 2 switchport.

Configuration of egress policing only differs from the policing configuration examples above in the application of the policy to the interface. The policer is created in the same manner, building the class map and policy map as in the above examples. Using the "police-to-10" policy created earlier, this same policy can be turned into an egress policer.

```
Cat6500(config)# interface fastethernet 5/2
Cat6500(config-if)# service-policy output police-to-10
```

The use of the service-policy command is used to apply a policy to an interface. The keyword to note in the example above is the use of the "output" parameter. This tells the PFC3 to apply this policy for outbound (egress) traffic on this interface.

8.11 Configuring Classification

The following section describes the QoS configuration components used to support classification on the PFC using Cisco IOS

8.11.1. CoS to DSCP Mapping (Cisco IOS)

On ingress to the switch, a frame will have a DSCP value set by the switch. If the port is in a trusted state, and the administrator has used the **mls qos trust-cos** keyword (on GE and 10GE ports or 10/100 ports on the WS-X6548 and WS-X6148 line cards), then the CoS value set in the frame will be used to determine the DSCP value set for the frame. As mentioned before, the switch can assign levels of service to the frame as it transits the switch based on the internal DSCP value.

When QoS is enabled, the switch creates a default map. Please refer to table 3 for default settings. This map is used to identify the DSCP value that will be set based on the CoS value. Alternatively, the administrator can set up a unique map. An example of this is shown below:

```
Cat6500(config)# mls qos map cos-dscp 20 30 1 43 63 12 13 8
```

The above command sets the following map:

CoS	0	1	2	3	4	5	6	7
DSCP	20	30	1	43	63	12	13	8

While it is very unlikely that the above map would be used in a real life network, it serves to give an idea of what can be achieved using this command.

8.11.2. IP Precedence to DSCP Mapping

Like the CoS to DSCP map, a frame can have a DSCP value determined from the incoming packets IP Precedence setting. This still only occurs if the port is set to trusted by the administrator, and they have used the **mls qos trust-ipprec** keyword. This keyword is only supported on GE ports and 10/100 ports on the WS-X6548 line cards. For 10/100 ports o the WS-X6348 6148, and WS-X6248 line cards, ACL's should be used to assign ip precedence trust to incoming data.

When QoS is enabled, the switch creates a default map. Please refer to Table 3 for default settings. This map is used to identify the DSCP value that will be set based on the IP Precedence value. Alternatively, the administrator can set up a unique map. An example of this is shown below:

```
Cat6500(config)# mls qos map ip-prec-dscp 20 30 1 43 63 12 13 8
```

The above command sets the following map:

IP Precedence	0	1	2	3	4	5	6	7
DSCP	20	30	1	43	63	12	13	8

While it is very unlikely that the above map would be used in a real life network, it serves to give an idea of what can be achieved using this command.

8.11.3. PFC Classification

When a frame is passed to the PFC, the process of classification can be performed to assign a new priority to an incoming frame. The caveat here is that this can only be done when the frame is from an untrusted port, or the frame has been classified as being untrusted.

A policy map class action can be used to

1. TRUST COS
2. TRUST IP-PRECEDENCE
3. TRUST DSCP
4. NO TRUST

The TRUST DSCP keyword assumes that the frame arriving into the PFC already has a DSCP value set prior to it entering the switch. The switch will maintain this DSCP value.

With TRUST IP-PRECEDENCE, the PFC will derive a DSCP value from the existing IP Precedence value resident in the ToS field. The PFC will use the IP Precedence to DSCP map to assign the correct

DSCP. A default map is created when QoS is enabled on the switch. Alternatively a map created by the administrator can be used to derive the DSCP value.

Like TRUST IP-PRECEDENCE, the TRUS COS keyword instructs the PFC to derive a DSCP value from the CoS in the frame header. There will also be a CoS to DSCP map (either a default one of an administrator assigned one) to assist the PFC in deriving the DSCP.

The **NO TRUST** form of the keyword is used when a frame arrives from an un-trusted port. This allows the frame to have a DSCP value assigned during the process of policing.

Lets look at an example of how a new priority (DSCP) can be assigned to different flows coming into the PFC using the following policy definition.

```
Cat6500(config-pmap)# class test access-group 102
Cat6500(config-pmap-c)# no trust
Cat6500(config-pmap-c)# set ip dscp 24
Cat6500(config-pmap-c)# exit
Cat6500(config-pmap)# exit
```

The above example shows the following:

1. An access list being created to identify http flows coming into the port
2. A policy map called *new-dscp-for-flow*
3. A class map (names *test*) that uses access list 102 to identify the traffic that this class-map will perform its action for.
4. The class map test will set the trust state for the incoming frame to un-trusted and assign a DSCP of 24 to that flow
5. This class map will also limit the aggregate of all http flows to a maximum of 1Mb.

8.12 MAC ACL Filtering

On the earlier versions of the PFC (PFC1 and PFC2), MAC ACL's could only be applied to inspect non-IP traffic such as IPX, DecNET, Appletalk, etc as well as layer 2 Ethernet traffic. With the PFC3B and PFC3BXL and a minimum of 12.2(18)SXD software, MAC ACL's can now be applied on specific interfaces to inspect all ingress traffic types, including IPv4, IPv6, MPLS and other MAC layer traffic. This feature is referred to as protocol independent MAC filtering and can be applied to any of the following interfaces:

1. VLAN interfaces with no IP address assigned
2. (Physical) Switch ports configured to support Ethernet over MPLS (EoMPLS)
3. (Logical) Sub-Interfaces on switch ports configured for EoMPLS.

This feature is enabled on a per interface basis and can be applied as follows:

```
Cat6500(config-if)# mac packet-classify
```

9. Appendix One – Buffers and Queues Summary

Module	Port Type	RX Queue	TX Queue	Total Buffer	RX Buffer	TX Buffer
CEF720 Modules						
WS-X6704-10GE	4 x 10GE	1Q8T	1P7Q8T	16MB	2MB	14MB
WS-X6704-10GE	4 x 10GE + DFC	8Q8T	1P7Q8T	16MB	2MB	14MB
WS-X6724-SFP	24 x GE SFP	1Q8T	1P3Q8T	1.3Mb	166KB	1.2MB
WS-X6724-SFP	24 x GE SFP + DFC	2Q8T	1P3Q8T	1.3Mb	166KB	1.2MB
WS-X6748-GE-TX	48 x GE TX	1Q8T	1P3Q8T	1.3Mb	166KB	1.2MB
WS-X6748-GE-TX	48 x GE TX + DFC	2Q8T	1P3Q8T	1.3Mb	166KB	1.2MB
WS-X6748-SFP	48 x GE SFP	1Q8T	1P3Q8T	1.3Mb	166KB	1.2MB
WS-X6748-SFP	48 x GE SFP + DFC	2Q8T	1P3Q8T	1.3Mb	166KB	1.2MB
CEF256 Modules						
WS-X6501-10GEX4	1 x 10GE	1P1Q8T	1P2Q1T	64.2MB	256KB	64MB
WS-X6502-10GE	1 x 10GE	1P1Q8T	1P2Q1T	64.2MB	256KB	64MB
WS-X6516-GBIC	16 x GE GBIC	1P1Q4T	1P2Q2T	512KB	73KB	439KB
WS-X6516A-GBIC	16 x GE GBIC	1P1Q4T	1P2Q2T	1MB	135KB	946KB
WS-X6516-GE-TX	16 x 10/100/1000	1P1Q4T	1P2Q2T	512KB	73KB	439KB
WS-X6524-100FX-MM	24 x 100FX Multimode	1P1Q0T	1P3Q1T	1.1MB	28KB	1088KB
WS-X6548-RJ-21	48 x 10/100	1P1Q0T	1P3Q1T	1.1MB	28KB	1088KB
WS-X6548-RJ-45	48 x 10/100	1P1Q0T	1P3Q1T	1.1MB	28KB	1088KB
WS-X6548-GE-TX	48 x 10/100/1000	1Q2T shared between 8 ports	1P2Q2T shared between 8 ports	1.4MB shared between 8 ports	185KB shared between 8 ports	1.2MB shared between 8 ports
WS-X6548V-GE-TX	48 x 10/100/1000 with Cisco inline power	1Q2T shared between 8 ports	1P2Q2T shared between 8 ports	1.4MB shared between 8 ports	185KB shared between 8 ports	1.2MB shared between 8 ports
WS-X6548-GE-45AF	48 x 10/100/1000 with 802.3af inline power	1Q2T shared between 8 ports	1P2Q2T shared between 8 ports	1.4MB shared between 8 ports	185KB shared between 8 ports	1.2MB shared between 8 ports
dCEF256 Modules						
WS-X6816-GBIC	16 x GE GBIC	1P1Q4T	1P2Q2T	512KB	73KB	439KB
Classic Modules						
WS-X6024-10FL-MT	24 x 10Base FL	1Q4T	2Q2T	64KB	8KB	56KB
WS-X6148-RJ-21	48 x 10/100	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6148-RJ-21V	48 x 10/100 with Cisco inline power	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6148-21AF	48 x 10/100 with 802.3AF inline power	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6148-RJ-45	48 x 10/100	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6148-RJ-45V	48 x 10/100 with Cisco inline power	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6148-45AF	48 x 10/100 with 802.3AF inline power	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6148-GE-TX	48 x 10/100/1000	1Q2T shared between 8 ports	1P2Q2T shared between 8 ports	1.4MB shared between 8 ports	185KB shared between 8 ports	1.2MB shared between 8 ports

WS-X6148V-GE-TX	48 x 10/100/1000 with Cisco inline power	1Q2T shared between 8 ports	1P2Q2T shared between 8 ports	1.4MB shared between 8 ports	185KB shared between 8 ports	1.2MB shared between 8 ports
WS-X6148-GE-45AF	48 x 10/100/1000 with 802.3AF inline power	1Q2T shared between 8 ports	1P2Q2T shared between 8 ports	1.4MB shared between 8 ports	185KB shared between 8 ports	1.2MB shared between 8 ports
WS-X6224-100FX-MT	24 x 100FX	1Q4T	2Q2T	64KB	8KB	56KB
WS-X6248-RJ-45	48 x 10/100	1Q4T	2Q2T	64KB	8KB	56KB
WS-X6248-TEL	48 x 10/100 (RJ21)	1Q4T	2Q2T	64KB	8KB	56KB
WS-X6248A-TEL	48 x 10/100 (Telco)	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6316-GE-TX	16 x 10/100/1000	1P1Q4T	1P2Q2T	512KB	73KB	439KB
WS-X6324-100FX-MM	24 x 100FX	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6324-100FX-SM	24 x 100FX	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6348-RJ-45	48 x 10/100	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6348-RJ-45V	48 x 10/100 with Cisco inline power	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6348-RJ-21V	48 x 10/100 with Cisco inline power	1Q4T	2Q2T	128KB	16KB	112KB
WS-X6408-GBIC	8 a GE GBIC	1P1Q4T	1P2Q2T	512KB	80KB	432KB
WS-X6408A-GBIC	8 x GE GBIC	1Q4T	2Q2T	512KB	73KB	439KB
WS-X6416-GE-MT	16 x 10/100/1000	1P1Q4T	1P2Q2T	512KB	73KB	439KB
WS-X6416-GBIC	16 x GE GBIC	1P1Q4T	1P2Q2T	512KB	73KB	439KB

10. Appendix Two – QoS Summary Feature List

The following list provides a summary of the QoS features found in the Catalyst 6500, where the feature is processed and if the feature is applied on ingress or egress.

QoS Feature	Processed in..?	Applied on Ingress or Egress	Supported PFC
Port Trust	Linecard Port ASIC	Ingress	N/A
Default CoS Assignment	Linecard Port ASIC	Ingress	N/A
CoS to DSCP Map	Linecard Port ASIC	Ingress	N/A
IP Precedence to DSCP Map	Linecard Port ASIC	Ingress	N/A
CoS Mutation	Linecard Port ASIC	Ingress	N/A
Ingress Classification	Policy Feature Card	Ingress	PFC1, 2, 3A, 3B, 3BXL
Ingress Aggregate Policing	Policy Feature Card	Ingress	PFC1, 2, 3A, 3B, 3BXL
Egress Classification	Policy Feature Card	Egress	PFC3A, 3B, 3BXL
Egress Aggregate Policing	Policy Feature Card	Egress	PFC3A, 3B, 3BXL
Microflow Policing	Policy Feature Card	Ingress	PFC1, 2, 3A, 3B, 3BXL
User Based Rate Limiting	Policy Feature Card	Ingress	PFC3A, 3B, 3BXL
DSCP to CoS Map	Linecard Port ASIC	Egress	N/A
Egress DSCP Mutation	Policy Feature Card	Egress	PFC3A, 3B, 3BXL
Weighted Round Robin	Linecard Port ASIC	Egress	N/A
Deficit Weighted Round Robin	Linecard Port ASIC	Egress	N/A
Shaped Round Robin	Linecard Port ASIC	Egress	N/A
Strict Priority Queuing	Linecard Port ASIC	Egress	N/A
Tail Drop	Linecard Port ASIC	Egress	N/A
Weighted Random Early Discard	Linecard Port ASIC	Egress	N/A

11. Appendix Three – Comparison of QoS Features between PFC's

Over the years a number of different PFC versions have been released. The table below provides a high level overview of the major differences between the QoS capabilities of each PFC.

Feature	PFC1	PFC2	PFC3A	PFC3B	PFC3BXL
ACL Classification	√	√	√	√	√
Microflow Policing	√	√	√	√	√
Ingress Aggregate Policing	√	√	√	√	√
Egress Aggregate Policing			√	√	√
User Based Rate Limiting			√	√	√
Egress DSCP Mutation			√	√	√
Ingress CoS Mutation			√	√	√
Number of QoS ACL's	16K	32K	32K	32K	32K
Number of QoS Masks	2K	4K	4K	4K	4K
Number of QoS Labels	512	512	512	4096	4096

12. Appendix Four - Default QoS Settings for Selected Catalyst 6500 Linecards

The following section will display the default QoS settings for some of the major linecards. A show command will be used to display those settings on one of the linecards ports.

12.1.1. WS-X6148-RJ-45

```
Cat6500#show queueing interface f4/1
Interface FastEthernet4/1 queueing strategy:  Weighted Round-Robin
  Port QoS is enabled
  Port is untrusted
  Extend trust state: not trusted [COS = 0]
  Default COS is 0
  Transmit queues [type = 2q2t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR low             2
       2         WRR high            2

    WRR bandwidth ratios:  100[queue 1] 255[queue 2]
    queue-limit ratios:     70[queue 1]  30[queue 2]

    queue tail-drop-thresholds
    --------------------------
    1     80[1] 100[2]
    2     80[1] 100[2]

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    2     1      4 5
    2     2      6 7

  Receive queues [type = 1q4t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         Standard            4


    queue tail-drop-thresholds
    --------------------------
    1     100[1] 100[2] 100[3] 100[4]

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    1     3      4 5
    1     4      6 7


  Packets dropped on Transmit:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
```

```
    1     1               0  [0 1 ]
    1     2               0  [2 3 ]
    2     1               0  [4 5 ]
    2     2               0  [6 7 ]

  Packets dropped on Receive:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
    1     1               0  [0 1 ]
    1     2               0  [2 3 ]
    1     3               0  [4 5 ]
    1     4               0  [6 7 ]
```

12.1.2. WS-X6516-GE-TX/WS-X6516-GBIC

```
Cat6500#show queueing interface g9/9
Interface GigabitEthernet9/9 queueing strategy:  Weighted Round-Robin
  Port QoS is enabled
  Port is untrusted
  Extend trust state: not trusted [COS = 0]
  Default COS is 0
  Transmit queues [type = 1p2q2t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR low             2
       2         WRR high            2
       3         Priority            1

    WRR bandwidth ratios:  100[queue 1] 255[queue 2]
    queue-limit ratios:     70[queue 1]  15[queue 2]

    queue random-detect-min-thresholds
    ----------------------------------
      1    40[1] 70[2]
      2    40[1] 70[2]

    queue random-detect-max-thresholds
    ----------------------------------
      1    70[1] 100[2]
      2    70[1] 100[2]

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    2     1      4 6
    2     2      7
    3     1      5

  Receive queues [type = 1p1q4t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         Standard            4
       2         Priority            1
```

```
    queue tail-drop-thresholds
    --------------------------
    1     100[1] 100[2] 100[3] 100[4]

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    1     3      4 6
    1     4      7
    2     1      5


  Packets dropped on Transmit:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
    1     1               0  [0 1 ]
    1     2               0  [2 3 ]
    2     1               0  [4 6 ]
    2     2               0* [7 ]
    3     1               0* [5 ]
                                * - shared transmit counter

  Packets dropped on Receive:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
    1     1               0  [0 1 ]
    1     2               0  [2 3 ]
    1     3               0  [4 6 ]
    1     4               0* [7 ]
    2     1               0* [5 ]
                                * - shared receive counter
```

12.1.3. WS-X6516a-GBIC

```
Cat6500#show queueing interface g1/1
Interface GigabitEthernet1/1 queueing strategy:  Weighted Round-Robin

  QoS is disabled globally
  Trust state: trust DSCP
  Extend trust state: not trusted [COS = 0]
  Default COS is 0
    Queueing Mode In Tx direction: mode-cos
    Transmit queues [type = 1p2q2t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR low             2
       2         WRR high            2
       3         Priority            1

    WRR bandwidth ratios:  255[queue 1]   1[queue 2]
    queue-limit ratios:    100[queue 1]   0[queue 2]
```

```
  queue random-detect-min-thresholds
  ----------------------------------
    1    100[1] 100[2]
    2    100[1] 100[2]

  queue random-detect-max-thresholds
  ----------------------------------
    1    100[1] 100[2]
    2    100[1] 100[2]

  queue thresh cos-map
  ---------------------------------------
  1     1      0 1 2 3 4 5 6 7
  1     2
  2     1
  2     2
  3     1

  Queueing Mode In Rx direction: mode-cos
  Receive queues [type = 1p1q4t]:
  Queue Id    Scheduling  Num of thresholds
  -----------------------------------------
     1         Standard           4
     2         Priority           1


  queue tail-drop-thresholds
  --------------------------
  1     100[1] 100[2] 100[3] 100[4]

  queue thresh cos-map
  ---------------------------------------
  1     1      0 1 2 3 4 5 6 7
  1     2
  1     3
  1     4
  2     1


Packets dropped on Transmit:
  BPDU packets:  0

  queue thresh    dropped  [cos-map]
  ---------------------------------------------------
  1     1              0  [0 1 2 3 4 5 6 7 ]
  1     2              0  []
  2     1              0  []
  2     2              0* []
  3     1              0* []
                              * - shared transmit counter

Packets dropped on Receive:
  BPDU packets:  0

  queue thresh    dropped  [cos-map]
  ---------------------------------------------------
```

```
1     1               0  [0 1 2 3 4 5 6 7 ]
1     2               0  []
1     3               0  []
1     4               0* []
2     1               0* []
                              * - shared receive counter
```

12.1.4. WS-X6548-RJ-45

```
Cat6500#show queueing interface f3/1
Interface FastEthernet3/1 queueing strategy:  Weighted Round-Robin
  Port QoS is enabled
  Port is untrusted
  Extend trust state: not trusted [COS = 0]
  Default COS is 0
  Transmit queues [type = 1p3q1t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR                 1
       2         WRR                 1
       3         WRR                 1
       4         Priority            1

    WRR bandwidth ratios:  100[queue 1] 150[queue 2] 200[queue 3]

    queue random-detect-min-thresholds
    ----------------------------------
      1    70[1]
      2    70[1]
      3    70[1]

    queue random-detect-max-thresholds
    ----------------------------------
      1    100[1]
      2    100[1]
      3    100[1]

    WRED disabled queues:

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    2     1      2 3 4
    3     1      6 7
    4     1      5

  Receive queues [type = 1p1q0t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         Standard            0
       2         Priority            1

    queue-limit ratios:     80[queue 1]  20[queue 2]

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1 2 3 4 6 7
```

```
    2     1      5


  Packets dropped on Transmit:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
    1     1               0  [0 1 ]
    2     1               0  [2 3 4 ]
    3     1               0  [6 7 ]
    4     1               0  [5 ]

  Packets dropped on Receive:
    BPDU packets:  0

    queue thresh    dropped  [cos-map]
    ---------------------------------------------------
    1     1               0  [0 1 2 3 4 6 7 ]
    2     1               0  [5 ]
```

12.1.5. WS-X6704-10GE

```
Cat6500#show queueing interface teng13/1
Interface TenGigabitEthernet13/1 queueing strategy:  Weighted Round-Robin

  QoS is disabled globally
  Trust state: trust DSCP
  Extend trust state: not trusted [COS = 0]
  Default COS is 0
    Queueing Mode In Tx direction: mode-cos
    Transmit queues [type = 1p7q8t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       01         WRR                 08
       02         WRR                 08
       03         WRR                 08
       04         WRR                 08
       05         WRR                 08
       06         WRR                 08
       07         WRR                 08
       08         Priority            01

    WRR bandwidth ratios:  100[queue 1]   0[queue 2]   0[queue 3]   0[queue 4]
0[queue 5]   0[queue 6]   0[queue 7]
    queue-limit ratios:    100[queue 1]   0[queue 2]   0[queue 3]   0[queue 4]
0[queue 5]   0[queue 6]   0[queue 7]

    queue tail-drop-thresholds
    --------------------------
    1     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
    2     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
    3     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
    4     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
    5     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
    6     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
```

```
7     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]

queue random-detect-min-thresholds
----------------------------------
  1    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  2    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  3    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  4    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  5    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  6    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  7    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]

queue random-detect-max-thresholds
----------------------------------
  1    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  2    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  3    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  4    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  5    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  6    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
  7    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]

WRED disabled queues:      1  2  3  4  5  6  7

queue thresh cos-map
---------------------------------------
1     1      0 1 2 3 4 5 6 7
1     2
1     3
1     4
1     5
1     6
1     7
1     8
2     1
2     2
2     3
2     4
2     5
2     6
2     7
2     8
3     1
3     2
3     3
3     4
3     5
3     6
3     7
3     8
4     1
4     2
4     3
4     4
4     5
4     6
4     7
```

```
  4     8
  5     1
  5     2
  5     3
  5     4
  5     5
  5     6
  5     7
  5     8
  6     1
  6     2
  6     3
  6     4
  6     5
  6     6
  6     7
  6     8
  7     1
  7     2
  7     3
  7     4
  7     5
  7     6
  7     7
  7     8
  8     1

  Queueing Mode In Rx direction: mode-cos
  Receive queues [type = 1q8t]:
  Queue Id    Scheduling  Num of thresholds
  ----------------------------------------
     01         WRR                 08

  WRR bandwidth ratios:  100[queue 1]
  queue-limit ratios:    100[queue 1]

  queue tail-drop-thresholds
  --------------------------
  1     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]

  queue thresh cos-map
  ---------------------------------------
  1     1      0 1 2 3 4 5 6 7
  1     2
  1     3
  1     4
  1     5
  1     6
  1     7
  1     8


Packets dropped on Transmit:

  queue     dropped  [cos-map]
  ---------------------------------------------
  1               0  [0 1 2 3 4 5 6 7 ]
```

```
    2                0  []
    3                0  []
    4                0  []
    5                0  []
    6                0  []
    7                0  []
    8                0  []

  Packets dropped on Receive:

    queue     dropped  [cos-map]
    ---------------------------------------------
    1                0  [0 1 2 3 4 5 6 7 ]
```

12.1.6. WS-X6748-GE-TX/WS-X6748-SFP

```
Cat6500#show queueing interface g1/4
Interface GigabitEthernet1/4 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 = 1p3q8t]:
    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]

    queue tail-drop-thresholds
    --------------------------
    1     70[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
    2     70[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
    3     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]

    queue random-detect-min-thresholds
    ----------------------------------
      1    40[1] 70[2] 70[3] 70[4] 70[5] 70[6] 70[7] 70[8]
      2    40[1] 70[2] 70[3] 70[4] 70[5] 70[6] 70[7] 70[8]
      3    70[1] 70[2] 70[3] 70[4] 70[5] 70[6] 70[7] 70[8]

    queue random-detect-max-thresholds
    ----------------------------------
      1    70[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
      2    70[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
      3    100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]

    WRED disabled queues:

    queue thresh cos-map
    ---------------------------------------
    1     1      0
```

```
1     2      1
1     3
1     4
1     5
1     6
1     7
1     8
2     1      2
2     2      3 4
2     3
2     4
2     5
2     6
2     7
2     8
3     1      6 7
3     2
3     3
3     4
3     5
3     6
3     7
3     8
4     1      5

Queueing Mode In Rx direction: mode-cos
Receive queues [type = 2q8t]:
Queue Id    Scheduling  Num of thresholds
-----------------------------------------
   01         WRR                 08
   02         WRR                 08

WRR bandwidth ratios:  100[queue 1]   0[queue 2]
queue-limit ratios:    100[queue 1]   0[queue 2]

queue tail-drop-thresholds
--------------------------
1     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]
2     100[1] 100[2] 100[3] 100[4] 100[5] 100[6] 100[7] 100[8]

queue thresh cos-map
---------------------------------------
1     1      0 1 2 3 4 5 6 7
1     2
1     3
1     4
1     5
1     6
1     7
1     8
2     1
2     2
2     3
2     4
2     5
2     6
2     7
```

```
2     8
```

12.1.7. WS-X6816-GBIC

```
Cat6500#show queueing interface g3/1
Interface GigabitEthernet3/1 queueing strategy:  Weighted Round-Robin

  QoS is disabled globally
  Trust state: trust DSCP
  Extend trust state: not trusted [COS = 0]
  Default COS is 0
    Queueing Mode In Tx direction: mode-cos
    Transmit queues [type = 1p2q2t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR low             2
       2         WRR high            2
       3         Priority            1

    WRR bandwidth ratios:  255[queue 1]   1[queue 2]
    queue-limit ratios:    100[queue 1]   0[queue 2]

    queue random-detect-min-thresholds
    ----------------------------------
      1    100[1] 100[2]
      2    100[1] 100[2]

    queue random-detect-max-thresholds
    ----------------------------------
      1    100[1] 100[2]
      2    100[1] 100[2]

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1 2 3 4 5 6 7
    1     2
    2     1
    2     2
    3     1

    Queueing Mode In Rx direction: mode-cos
    Receive queues [type = 1p1q4t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         Standard            4
       2         Priority            1


    queue tail-drop-thresholds
    --------------------------
    1     100[1] 100[2] 100[3] 100[4]

    queue thresh cos-map
    ---------------------------------------
    1     1      0 1 2 3 4 5 6 7
    1     2
```

```
  1     3
  1     4
  2     1


Packets dropped on Transmit:
  BPDU packets:  0

  queue thresh    dropped  [cos-map]
  ---------------------------------------------------
  1     1               0  [0 1 2 3 4 5 6 7 ]
  1     2               0  []
  2     1               0  []
  2     2               0* []
  3     1               0* []
                               * - shared transmit counter

Packets dropped on Receive:
  BPDU packets:  0

  queue thresh    dropped  [cos-map]
  ---------------------------------------------------
  1     1               0  [0 1 2 3 4 5 6 7 ]
  1     2               0  []
  1     3               0  []
  1     4               0* []
  2     1               0* []
                               * - shared receive counter
```