# *NYC Utility*

# *QoS Strategy and White Paper*

## *April 2005*

***Applied Methodologies, Inc.***

This paper is a published work containing proprietary information. It is not to be disclosed in whole or in part without the express written authorization of Applied Methodologies, Inc. Copyright 2005 Applied Methodologies, Inc.

***Applied Methodologies would like to thank the following NYC Utility professionals for their contribution to this paper.***

# Table of Contents

# *1.0 Introduction*

The NYC Utility enterprise has a myriad of applications and business system services that traverse its data network. For economical, flexibility, and operating contingency benefits NYC Utility is also considering turning their traditional “data” network into a “converged” network. What this means is that the traditional data network will support services that were not on the data network before such as Voice/Telephony and Video as well as other “content” that could be integrated into the network. Currently there are limited non-traditional data services traversing the network such as VoIP deployments and Video Conferencing. By migrating Voice services to the data network NYC Utility can position itself to reduce the cost for telephony and leverage the capacity of its existing data network. The reduced cost of telephony can be achieved by migrating from the traditional PBX based services. Instead of maintaining two different networks for two different services with their respective operating costs, one network to handle both services can be utilized. NYC Utility can leverage its current data network to provide levels of service to its current traditional data applications and future ones as well as provide for a basic form of security against Denial of Service and Worm attacks. Internet Protocol(IP) is becoming, if not already, the predominate transport protocol for just about any type of data or communication service today.

According to Alex Hadden-Boyd, director of marketing for IP communications in the Product and Technology Marketing Organization at Cisco, "If you think of IP as a universal translator," says Hadden-Boyd, "the various devices and applications on the network are starting to merge. PCs, PDAs, pagers, wireless phones, desk phones, and video endpoints are coming together. Users want to integrate not just the devices themselves, but also the desktop applications that run on them. Audio conferencing, videoconferencing, video telephony, Web conferencing--they can all be tied together through IP."

As the demands to tie everything together with IP increases so does the need to ensure that the demands for different applications are met for correct operation. If the future of corporate enterprises entails an all or majority IP based environment for all services then some technology needs to ensure that one less critical IP application(Web browsing) does not step over another critical IP application(such as Voice calls).

How can this be done? With Quality of Service(QoS) technologies, tools and deployment methodologies.

This document will provide the necessary information pertaining to the application of QoS onto NYC Utility's enterprise network to strategically position NYC Utility's network for future Voice, Video and other Data based applications. This document provides the following information:

- **General introduction to QoS concepts and internal NYC Utility QoS reference**
- **Strategic recommendations regarding QoS planning and implementation in NYC Utility**
- **Various device, application and protocol matrixes to facilitate planning**
- **Audit of NYC Utility's current network components to determine their QoS capabilities**
- **Design considerations and principals**
- **Identification of NYC Utility's candidate QoS applications**
- **Outline of NYC Utility's QoS model**
- **Outline of NYC Utility's QoS solution based on the models**
- **Introduction to NYC Utility's custom QoS tool set**
- **Recommended deployment approaches**
- **Outline of QoS management tools**
- **Outline of troubleshooting tools and methods**
- **Initial lab result findings for pre-deployment planning**
- **Next steps**
- **Appendixes providing additional information.**

The information in this paper is compiled from various sources such as Cisco Press texts, Cisco CCO web site pages, Internet QoS related Request For Comments (RFC), industry related web sites and periodicals, lab testing and from Applied Methodologies experience with other client implementations.

A specific recommendation is identified using ***bold*** with the words **"It is recommended"** throughout this document. It was easier to identify and state the recommendation in the context of its current section. Adding a "Summary of Recommendations" section at the end of each section would have been redundant and enlarged this document for no additional benefit.

## *1.1 General understanding of QoS and its uses*

QoS in a nutshell is the process of marking traffic or flows of packets and classifying them for a class(level) of service. The level of services is based on a priority hierarchy, from low priority to high priority for example. High priority marked packets will get "preferential" treatment as they traverse across a network while lower priority marked packets may be allowed to get the same treatment as the higher priority packets, but in the presence of higher marked packets, they don't. When the mixture of higher and lower priority marked packets are traversing a level of congestion may occur at any point. QoS helps to prevent or mitigate congestion by employing different queuing and scheduling features across network components.

The higher marked packets receive a class(level) of service to ensure that those marked packets never get dropped while the lower priority marked packets are candidates to be dropped, this ensures that there is enough buffer and queue space on the intermediate node's interfaces(routers and switches) to handle the higher class marked packets for their class of service. For example, packets for general web browsing will always get a lower priority traversing the network as opposed to a voice packet or video application that will always get the highest priority and guarantees of successful delivery across the network.

Voice over IP packets get the highest priority so call quality and handling is exceptional to the user's experience while less time sensitive traffic will fall into other priority levels indicative of their application's performance requirements.

The ultimate goal is to provide a managed unfairness approach applied to all applications regardless of their classification. A method and technology is needed to mark the data packets for "preferential" treatment throughout the network, plus to mark the packets to "differentiate" them so they can be handled differently throughout the network.

In theory such a concept and approach sounds easy to accomplish. Such a concept can be achieved easily, for example, for a small business with a small set of applications and users in a homogenous equipment and platform environment. The levels of priority and marking Voice traffic over web traffic; is not too difficult in that type of environment. It is not easy for an enterprise environment such as NYC Utility with a myriad of applications, hardware and software platforms, and many different service requirements. From a business and technological perspective to achieve such a concept is a challenging task that requires a careful amount of research, planning and testing in regards to the technology, solutions required, and the implementation approach.

More than a working knowledge of QoS tools and syntax is needed to deploy end-to-end QoS in a holistic manner. Implementing QoS improperly or incompletely could have negative ramifications to one or more application systems. So a cohesive plan must be drafted and followed prior to any implementation of QoS.

First, it is vital to understand the performance, bandwidth and behavioral requirements of various applications that require preferential or differential treatment within NYC Utility's network. Next, it is critical to understand the different technical options available with respect to QoS to achieve the required service levels.

## *1.2 QoS Models*

There are various models, architectural designs and technologies that create and manage a QoS solution to provide a level of managed unfairness across applications traversing an enterprise network.

There are two primary types of Internet standards based QoS architectures that use various technologies/protocols in the industry today.

**Integrated Services(IntServ)**

**Differentiated Services(DiffServ)**

It is beyond the scope of this paper to provide an in-depth explanation of both architectures, and it is expected of the reader to be familiar with them. A brief conceptual summary of the two methods from excerpts of various Cisco White papers and network architectural texts is outlined below and provides a basic idea of what is available to NYC Utility in terms of an architectural foundation to utilize QoS in its enterprise. Also a link to a white paper is also included to assist the reader.

Cisco IOS Software supports two fundamental QoS architectures: Differentiated Services (DiffServ) and Integrated Services (IntServ). It is important to note that these approaches are complimentary in nature. DiffServ enables better QoS scalability, while IntServ provides a guaranteed traffic delivery. Together, they can form a robust QoS deployment.

### *1.2.1 IntServ*

An IntServ approach utilizes Resource Reservation Protocol (RSVP) to signal a specific service. This makes it ideal for end-to-end delay and jitter-sensitive applications, such as Voice and Video. The IntServ approach of using Signaled QoS is one of the most misunderstood areas in the industry. This paper will attempt to clarify some common myths.

Signaled QoS is most appropriate for "intolerant" traffic, which does not like delay, jitter or packet loss (i.e.: Voice and Video). For former cases, it may be applicable for tolerant traffic - like data. For example, the pre-emption capabilities of RSVP could handle an extremely high-priority data burst (i.e.: executing a Stock trade during a bear run).

The framework of IntServ preserves the end-to-end semantics of QoS for IP. Key endpoints are the sender and receiver applications that request desired service from the

network for a set of flows, as defined by the source address, destination address, transport protocol, source port and destination port.

What is advantageous of a "signaling" mechanism is that it provides near ATM like VP/VC circuit setup for a voice call. Traditional telcos. will build(reserve) a circuit for a call. This can be accomplished in software "virtually" across the enterprise.

A simple example is as follows: A user needs to make a call, picks up his phone, his voice packets are marked, the marking process signals the access edge device to send a special path request to the receiver. The receiver responds with a path response indicating that there is reserved, and guaranteed, bandwidth available for the call. The returning path messages "signals" all the routers in between the two end-to-end callers to reserve and guarantee bandwidth for this marked flow of packets until another packet to tear down the reservation is seen. The end devices in a simplex vector signal, reserve, confirm and tear down the "reserved" bandwidth and delay for the call. This process will happen twice for full duplex operation. Simply put IntServ is basically an "on demand" process. It provides a preemptive on demand way of handling a class of traffic.

**Advantages of IntServ:**

1. **Conceptual simplicity -** facilitating the integration with network policy administration

2. **Discrete per flow QoS** - making it architecturally suitable to voice calls.

3. **Strict bandwidth guarantee and controlled load for Voice and Video traffic -** Includes end-to-end maintenance, so a loss of bandwidth in the core is handled appropriately.

4. **Admission Control -** Admitting calls based on available resources must be accomplished on an end-to-end basis. A local mechanism will do the job for a short time period, but it fails when congestion occurs at any point along the call path. So rather than having 12 calls with bad voice quality, ensuring that at least 10 calls go through, Call Admission Control(CAC) which can indicate to endpoints weather the desired bandwidth is available.

5. **Failure notification -** needs to be communicated to the endpoints, so a call will proceed without ensuring adequate reservation. While this can function, voice quality is severely impacted when congestion occurs but the call still works.

6. **Emergency situations -** Every voice or video implementation needs to have some mechanism to handle emergency situations. During such a period, there is a high likelihood of congestion due to the "panic factor". Only a Signaled mechanism can provide an adequate preemptive scheme that prevents a major disruption.

**Disadvantages of IntServ are:**

1. **Stateful -** All network elements must maintain state and exchange signaling message on a per flow basis which might require a significant amount of bandwidth on large networks.
2. **Periodic refresh messages are used -** might require protection from packet loss to keep the session(s) intact.
3. All intermediate nodes must implement RSPV **– platform support**

**Many can perceive the aforementioned fundamental value-adds of Signaled QoS; however, several myths do still exist with regards to RSVP deployment. These include:**

1. **Scalability -**RSVP is per-flow, but it cannot scale in the core
   If RSVP were perceived as an IntServ architecture as such, this would be a true statement. IntServ utilizes RSVP on a per-flow basis, which may cause scalability concerns; however, these fears are grossly overstated. Cisco routers have demonstrated the ability to handle more than 10,000 RSVP flows with minimal CPU and Memory impact. RSVP deployment does not pose a scalability problem for smaller networks.

2. **Admission Control** - RSVP is not available to perform admission control on Voice Gateways. Contrary to the aforementioned myth, RSVP is an ideal choice to handle admission control. It is the only known and industry-standardized mechanism to provide end-to-end admission control. RSVP has been an integral part of Cisco IOS Software since its invention in the mid '90s.

3. **Call Setup Delay -** Time taken to establish end-to-end reservations is extensive. Since Cisco IOS Software Release 12.2T, RSVP control messages can be marked with DiffServ Code Point (DSCP) marking, ensuring that RSVP control messages are treated at a priority. This minimizes potential delays, which are caused by the scheduling treatment accorded to the RSVP Control messages. (Please note that RSVP Control Messages are transmitted via User Datagram Protocol (UDP).)

### *1.2.2 DiffServ*

The concept of DiffServ offers different network service levels to a set of packets and packet by packet thereby "enabling" scalable service discrimination in the internet or enterprise network without the need for per-flow based state and signaling at every hop. Packets of a particular service(application) belong to a particular "class" and treatment of each "class" is described by PHB or (Per Hop Behaviors) with which the node must comply. A very rich, detailed, granular and scalable amount of services can be constructed by a combination of using the following technique:

- **Setting bits in a field in the IP header upon network entry or at a network boundary using legacy IP precedence or the more scalable DSCPs**
- **Using these field bit markings the nodes inside the network forward the packets based on the markings.**
- **Conditioning the marked packets at the network boundaries in accordance with the requirements of rules of each "class" or service(application)**

The major working attribute of DiffServ is the definition of different PHBs that an application can receive from the network. The three DiffServ PHBs are outlined below:

**Expedited Forwarding(EF)** Provides a strict priority service for packets marked in this manner
**Assured Forwarding(AF)** Provides a qualified delivery guarantee and makes the provision of oversubscription to this service.

**Class Selectors(CS)** Provides code points in the IP header field is basically a way of using/identifying the IP Precedence bits in the presence of DSCP bits set.

DiffServ constructs services from PHBs by classifying packets into "classes" at the edge, boundary, middle(re-classify) of the network and marking the packets accordingly. Optionally, the packets can be metered with policing or shaping technique. The core of the network implements the PHBs and uses the **"individual"** packet markings to make the necessary queuing and dropping decisions. Remember the example from the beginning of this section where the packets are marked and then their markings are "reviewed" as they cross the network? Individual packets are marked for a certain class(level) of service and are handled consistently at any point in the network to ensure a level of managed unfairness just like the example at the beginning of this section outlined. The http packets will be marked for a certain PHB versus the voice packet which will have a "higher" marking PHB and will be handled at a higher level of service at the actual HOP. As apposed to IntServ Flows, DiffServ is granular at the packet level and classifications can be coarse to dense as needed. With DiffServ the bits called Differentiated Services Code Points or (DSCP) can be scaled to 64 different markings. This means you can mark a packet 64 different ways.

**Advantages of DiffServ:**

1. **Scalability –** No state or flow information is required to be maintained(though this was never really an issue with IntServ and the combination could provide exceptional service levels)

2. **Performance –** the packet contents need to be inspected only once for classification purposes. At that time, the packet is marked and all subsequent QoS decisions are made on the value of a fixed field in the packet header, reducing processing requirements. This process could be conducted in hardware ASICs

3. **Interoperability –** All vendors run IP and IP is the most prevalent and flexible protocol in the enterprise. Also provides a class migration Ipv6.

4. **Flexibility –** The DiffServ model does not prescribe any particular feature(such as a queuing technique) to be implemented by a network node. The node can use whatever features optimize its hardware and architecture, as long as it is consistent with the behavior expectations defined in the PHBs.

**Disadvantages of DiffServ:**

1. **No end-to-end bandwidth reservations are present –** so, guarantees of services can be impaired by network nodes that do not implement the PHBs appropriately over congested links or by nodes that are not engineered correctly for the expected traffic volume of a specific class.

2. **The lack of a per-flow/per-session CAC –** makes it possible for applications to congest each other. High priority traffic hurts other high priority traffic. For example if only enough bandwidth for 10 voice calls exists and an $11^{th}$ call is permitted, all 11 calls suffer call-quality deterioration..

IntServ provides for a rich end-to-end QoS solution, by way of end-to-end signaling, state-maintenance (for each RSVP-flow and reservation), and admission control at each network element. DiffServ, on the other hand, addresses the clear need for relatively simple and coarse methods of categorizing traffic into different classes, also called class of service (CoS), and applying QoS parameters to those classes. To accomplish this, packets are first divided into classes by marking the type of service (ToS) byte in the IP header. A 6-bit bit-pattern (called the Differentiated Services Code Point **(DSCP)** in the IPv4 ToS Octet or the IPv6 Traffic Class Octet is used to this end.

As the IntServ and DiffServ models have evolved, the general popularity of one method versus the other has swung back and forth and their coexistence has become an ever-increasing struggle, with committed advocates on both sides. Today the debates over the advantages of each continue without a clear, industry-wide agreed-upon resolution. The realization has begun that neither method offers a complete solution and, that elements of both should be combined to provide the most general method applicable to the widest range of traffic and applications types.

DiffServ mechanisms provide bandwidth guarantees at different levels, but none of them provides bandwidth reservations. Reservation on the other hand implies that a flow of packets can be recognized and that a certain amount of bandwidth has been agreed to be set aside for that flow.

With the advent of faster platform processors, IOS enhancements and faster links(Gigabit Ethernet) using IntServ RSVP is again a consideration. Since NYC Utility has moved to towards Gigabit Optical Backbones consisting of high speed and fabric capable switches with the capacity to enable 100mbs to the desktop. Scaling for multiple on demand call flows could be possible in the future.

Because most of the implementations discussed and recommended by Cisco refer to DiffServ the most scalable and flexible to use to date, most of the options in Cisco's IOS and its QoS Baseline Model utilize the DiffServ approach. The strategies and configurations outlined in this paper will focus on the DiffServ method. However, it should be noted that advances with IntServ might lend to its use in a separate or in a combined manner with DiffServ within NYC Utility in the future. A possible scenario is that all voice traffic is on demand RSVP IntServ based and all other applications are DiffServ based.

## *2.0 How does QoS pertain to NYC Utility*

NYC Utility's Information department is receiving more requests for the support of voice and video based applications. Today these applications, deployed in production or in a limited test pilot are running side by side with the general network traffic. There is enough bandwidth today to accomplish this on a limited basis but as the need for more voice and video increases scaling the bandwidth is not feasible. NYC Utility needs to implement QoS to ensure that there is enough bandwidth and throughput for its critical applications and utilize its recent investment in network bandwidth upgrades to its fullest potential. So, NYC Utility can utilize the granularity of the DiffServ model to mark packets for various levels of QoS to provide the needed levels of service while taking full advantage of its bandwidth investment.

The RFC standards(see ***Appendix B.)*** based DiffServ DSCP table is illustrated below to show the number of possible markings available for packet classification and how granular/scalable this model is:

| **DROP Precedence** | **Class #1** | **Class #2** | **Class #3** | **Class #4** |
|---|---|---|---|---|
| **Low Drop Prec** | (AF11) 001010 | (AF21) 010010 | (AF31) 011010 | (AF41) 100010 |
| **Medium Drop Prec** | (AF12) 001100 | (AF22) 010100 | (AF32) 011100 | (AF42) 100100 |
| **High Drop Prec** | (AF13) 001110 | (AF23) 010110 | (AF33) 011110 | (AF43) 100110 |

These markings are a standards based approach to a set of default markings that a vendor can apply to its products. These default markings are also referred as PHBs or Per Hop Behaviors( mentioned in the previous DiffServ section). What this means simply is that a router or switch that is DiffServ aware receives a packet with a DiffServ marking will apply a certain forwarding priority behavior at that router/switch hop. As DiffServ marked packets traverse across routers their markings invoke a certain PHB at each hop. The Cisco QoS Baseline Model utilizes these markings and these are the markings that NYC Utility will utilize as well in its implementation.

There is another basic set of markings defined for backward compatibility with the legacy IP precedence bits. These are referenced as Class Selectors - **CS1, CS2 through CS7** is the same as using the IP Precedence bits.

| | |
|---|---|
| **CS1—Class Selector 1** | **(precedence 1)** |
| **CS2—Class Selector 2** | **(precedence 2)** |
| **CS3—Class Selector 3** | **(precedence 3)** |
| **CS4—Class Selector 4** | **(precedence 4)** |
| **CS5—Class Selector 5** | **(precedence 5)** |
| **CS5—Class Selector 6** | **(precedence 6)** |
| **CS7—Class Selector 7** | **(precedence 7)** |

***Note: When discussing the general application definitions these markings DSCPxx, AFxx or CSx will be referenced.***

**TIP:** DSCP value name to decimal conversion formula

To convert from the AF name to the decimal equivalent , you can use a simple formula: Think of the AF values as AF**xy**, the formula is

**8x + 2y** = decimal value.

For example, AF41 gives you a formula of **(8 * 4) + (2 * 1) = 34.**

Keep this in mind when using a protocol analyzer to troubleshoot a QoS or application issue. Some analyzers will show in the IP header the AF name, others will just show the decimal value and still others may show both. Using the NYC Utility QoS Model matrixes with the DSCP value tables and this formula will help in identifying any values not on the matrix or set incorrectly.

## *2.1 Sample DSCP packet*

Below is what a typical unmarked packet would look like:

```
------------ IP Header ------------
Version = 4
Header length = 20
Differentiated Services (DS) Field = 0x00
   0000 00.. DS Codepoint = Default PHB (0)
   .... ..00 Unused
Packet length = 200
Id = af60
Fragmentation Info = 0x0000
   .0.. .... .... .... Don't Fragment Bit = FALSE
   ..0. .... .... .... More Fragments Bit = FALSE
   ...0 0000 0000 0000 Fragment offset = 0
Time to live = 128
Protocol = UDP (17)
Header checksum = CCAD (Verified CCAD)
Source address = 222.1.1.8
Destination address = 222.1.1.12
```

**Below is what a typical DSCP marked VoIP packet would look like**

```
------------ IP Header ------------
Version = 4
Header length = 20
Differentiated Services (DS) Field = 0xB8
   1011 10.. DS Codepoint = Expedited Forwarding (46)
   .... ..00 Unused
Packet length = 200
Id = ed36
Fragmentation Info = 0x0000
   .0.. .... .... .... Don't Fragment Bit = FALSE
   ..0. .... .... .... More Fragments Bit = FALSE
   ...0 0000 0000 0000 Fragment offset = 0
Time to live = 128
Protocol = UDP (17)
Header checksum = 118C (Verified 118C)
Source address = 10.1.1.2
Destination address = 10.1.1.3
```

A complete tutorial of DiffServ and its operation is beyond the scope of this document. **It is recommended** that the audience for this document familiarize themselves with DiffServ concepts by reviewing the following link:

**DiffServ—The Scalable End-to-End QoS Model**

**http://www.cisco.com/warp/public/cc/pd/iosw/ioft/iofwft/prodlit/difse_wp.htm**

**Note:** *QoS related RFC document number listings are outlined in Appendix B.*

## *2.2 Quality of Service Application categories*

Most applications fall into a QoS model's hierarchy neatly and some don't. This section outlines the major and generic applications that require QoS and provides some background and recommendations pertaining to each type of application class. This section is also a **"reference"** section that could be used to refer to when studying the remaining sections of this paper. From this section NYC Utility can understand the Cisco defined model of applications that fall into its ***Baseline QoS Mode***l. NYC Utility can use these definitions and compare them to its own application suite and classify which applications fall into which category.

The application definitions listed here form the basis of the application categories from **Cisco's QoS Baseline QoS Model.**

Cisco provides a QoS Baseline Model that customers can follow directly or model their own from the Baseline to easily classify, quantify and assign applications to the appropriate class of service based on the applications unique characteristics. The application types defined(from highest priority/importance to lowest) by the Cisco QoS Baseline are as follows:

| |
|---|
| **Voice** |
| **Interactive-Video** |
| **Streaming-Video** |
| **Call Signaling** |
| **IP Routing/Network Control** |
| **Network Management** |
| **Mission-Critical Data** |
| **Transactional Data** |
| **Bulk Data** |
| **Best Effort** |
| **Scavenger** |

Below are the details of each application type in the model and their relation to NYC Utility's use. There are recommendations and best practices in this section as to what DiffServ markings are applied to each application class.

### *2.2.1 Voice bearer and Voice signaling traffic*

Voice traffic is probably one if not the most important type of traffic on the network. It is this type of traffic that always gets the highest priority in a QoS model. Since Voice is so important to an enterprise's way of conducting business and is usually the impetus for QoS. Because of this extra attention is given here in this section to voice this section can serve NYC Utility as a general reference for the remaining sections of this paper.

VoIP deployments require the provisioning of explicit priority servicing for VoIP bearing streaming traffic(UDP based RTP packets) and a guaranteed bandwidth service for Call Signaling traffic(SIP, Skinny H225/245 packets).

**Voice bearer traffic**
The list below outlines the recommended Cisco Baseline QoS requirements for handling Voice bearer traffic:

**It is recommended** that Voice traffic should be marked to **DSCP EF** per the Cisco QoS baseline and RFC 3246. This gives voice bearer traffic the highest level of priority and ensures that the PHBs will always forward this traffic first.

- Loss should be no more than 1%
- One direction latency from, mouth to ear, should be no more than 150ms.
  However, there is room for 200ms by most specs, but this should only be done if absolutely required and 150ms cannot be met.
- Average one-way jitter should be targeted at less than 30ms.
  This entails working with and tuning the jitter buffer if applicable on various end devices. Further discussion and recommendations on jitter issues is listed in this section.
- Guaranteed priority bandwidth within the range of 21 to 320 kbs **per** call depending of the codec, sampling rate and layer 2 overhead.

Voice Quality is directly affected by all three QoS quality Factors: **Loss**, **Latency** and **Jitter** so these issues will be discussed a little further here with relationship to NYC Utility's needs.

**Loss**
Loss causes voice clipping and skips. Packet Loss Concealment (PLC) is a technique used to mask the effects of lost or discarded VoIP packets. The method of PLC used depends upon the type of codec that NYC Utility uses. A simple method used by waveform codecs such as G7.11(PLC for G7.11) is to replay the last received sample with increasing attenuations at each repeat; the waveform does not change much from one sample to the next. This technique can be effective at concealing the loss of up to 20ms of samples.

The Packetization Interval determines the size of samples contained within a single packet. Assuming a 20ms default Packetization Interval, the loss of two or more consecutive packets results in a noticeable duration of voice quality. So, assuming a random distribution of drops within a single voice flow a drop rate of 1 percent in a voice stream would result in a loss that could be be concealed every 3 minutes on average. A 0.25 percent drop rate would result in a loss that could not be concealed once every 53 minutes.

The larger the Packetization Interval, for the given probability of packet loss could result in worse perceived call quality. **It is recommended** that NYC Utility, in its design and testing keep the Packetization Interval around 20ms.

For low bit rate, frame based codecs, such as G.729 and G.723 use more sophisticated PLC techniques that can conceal up to 30 to 40 ms of loss with tolerable quality when available history is used for the interpolation is still relevant.

With frame based codecs the Packetization Interval determines the number of frames carried in a single packet. As with waveform based codecs, if the Packetization Interval is greater than the loss that the PLC algorithm can interpolate for, PLC cannot effectively conceal the loss of a single packet.

VoIP networks are typically designed for very close to 0 percent VoIP packet loss, with the only possible losses resulting from layer 2 bit errors and network outages.

***Note:*** *NYC Utility should review and fully understand what type of Codec it is using, its bandwidth requirements and its relevant packet interval and PLC options used in its IP phones and Call Manager system.*

**Latency**

Latency can cause voice quality degradation if it is excessive. A design consideration that NYC Utility should and must meet is the latency target set by the ITU of G.114. This states that 150ms of one-way, end-to-end (mouth to ear) delay ensures users satisfaction for telephony applications. NYC Utility's QoS Design and policy standards should adopt this budget to the various components of network delay. There are many different delay factors in NYC Utility's environment and they are listed here for reference:

- **Processing delay(processing the packet in the router and switch) = x milliseconds**
- **Queuing/Scheduling delay(moving the packet into the right queue and its turn waiting in the queue) = x milliseconds**
- **Serialization delay(putting the packet actually onto the wire or air) x = milliseconds**
- **Propagation delay(the time it takes for the packet to move across the network) x = milliseconds**
- **Service delay(VoIP gateway codec and de-jitter buffer) x = milliseconds.**

If the end-to-end voice delay becomes too long the conversation begins to sound like echoing. Cisco recommends designing to the ITU standard of 150ms. Again, if constraints do exists and this target cannot be met the delay boundary can be increased to 200ms without significant impact on the quality of the call and mean opinion scores(MOS).

**Jitter**

Jitter buffers, also know as play out buffers, are used to change asynchronous packet arrivals into a synchronous stream by turning variable network delays into constant delays at the destination end systems. The role of the jitter buffer is to trade off between delay and the probability of interrupted play out because of late packets. Later or out-of-order packets are discarded.

If the jitter buffer is set either arbitrarily large or small, it imposes unnecessary constraints on the characteristics of the network. A jitter buffer set too large adds to end-to-end delay, meaning that less delay budget is available for the network, hence the network needs to support a tighter delay target than practically necessary. If a jitter buffer is so small to accommodate the network jitter, buffer underflows or overflows can occur.

Adaptive jitter buffers aim to overcome these issues by dynamically tuning the jitter buffer size to the lowest acceptable value. NYC Utility should review its VoIP product's jitter buffer characteristics to understand the buffer sizes utilized between IP phones. This information can assist NYC Utility in tuning and troubleshooting VoIP.

VoIP Bearer traffic, because of its strict service-level requirements is best suited to use the expedited forwarding (EF) PHB.

VoIP bandwidth streams consumption in bps, is calculated by adding the VoIP sample payload, in bytes, to the 40 byte IP, UDP and RTP(considering that cRTP or any other L3 compression is not used)then multiplying this value by 8 to convert it into bits and then multiplying that product again by the Packetization rate(50pps is usually the default).

The service parameters menu in Cisco Call Manager Administration can be used to adjust the packet rate. NYC Utility should keep this in mind for tuning and will be listed as an option in Section 11.0 of this paper.

A careful amount of consideration should be applied to defining and outlining what NYC Utility's specific bandwidth requirement for each VoIP stream is before even considering any QoS implementation.

These calculations can rear simple and almost exact requirements with just L3 overhead. However, the results can be much more exact and realistic to the bandwidth requirements including L2 packet overhead numbers, preambles and MAC headers.

**Voice bandwidth requirement table without L2 overhead**

| Bandwidth Consumption | Packetization Interval | Voice payload in bytes | Packets Per Second | Bandwidth per conversation |
|---|---|---|---|---|
| G.711 | 20ms | 160 | 50 | 80kbs |
| G729a | 20ms | 20 | 50 | 24kbs |

**Voice bandwidth requirement table with L2 overhead**

| Bandwidth Consumption | Packets Per Second | 802.1Q Ethernet bandwidth per conservation | PPP bandwidth per conversation |
|---|---|---|---|
| G.711 | 50 | 93kbs | 84kbs |
| G729a | 50 | 37kbs | 28kbs |

The following tables must be completed so NYC Utility can provision and understand the proper amount of bandwidth per stream, which can be applied to a QoS class.

Having these "bandwidth per stream" types outlined and defined helps in the creation of the overall end-to-end QoS design and capacity planning for the expected number of voice calls that the enterprise can handle. Also, keeping a table of the codec and bandwidth requirements available is useful for troubleshooting and adding to when changes are made. Fore example a newer codec could be used in the future and its flow requirements should be added to the table.

If VAD(Voice activity Detection) is to be employed remember that VAD can be used to reduce the payload by transmitting 2 bytes of payload during silent instead of the full payload size(vendor implementation dependant). However, for bandwidth engineering, VAD should be taken into account.

Cisco has a great tool for calculating the amount of bandwidth required per codec for the number of calls provisioned.

**http://tools.cisco.com/Support/VBC/do/CodecCalc1.do**

**It is recommended** that NYC Utility consider using a bit rate codec over the waveform codec in the future for the following reasons:

- **Lower amount of packets required for call thus providing more bandwidth for additional calls across the network, especially for the calls going over slower T-1 links.**
- **Provides just as high of quality as a waveform codec such as G.711**
- **Less overhead on the per hop device's queues from reduced number of packets for the call and its overall length.**
- **Can conceal(PCL) more loss in the 30-40ms ranges**

There is no immediate need to change over from G.711 to G.729a. As the quality of the G729 series improves NYC Utility could convert and realize the bandwidth saving and scalability gains from having a switch port provisioned for one 64kbs call at G.711. For example, eight calls on the same port at 8kbs using G.729a without changing any QoS configurations can now be made on the same port as opposed to one call on the port currently at 64kbs using G.711.

### *2.2.2 Session Initiation Protocol(SIP)*

**It is recommended** that NYC Utility consider testing SIP as its future signaling protocol over H323 for Voice for the following reasons:

The Session Initiation Protocol (SIP) is an application level signaling protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network. SIP can support any type of single-media or multimedia session, including teleconferencing.

SIP is just one component in the set of protocols and services needed to support multimedia exchanges over the Internet. SIP is the signaling protocol that enables one party to place a call to another party and to negotiate the parameters of a multimedia session. The actual audio, video, or other multimedia content is exchanged between session participants using an appropriate transport protocol. In many cases, the transport protocol to use is the Real-Time Transport Protocol (RTP). Directory access and lookup protocols are also needed.

The key driving force behind SIP is to enable Internet telephony, also referred to as Voice over IP (VoIP). There is wide industry acceptance that SIP will be the standard IP signaling mechanism for voice and multimedia calling services. Further, as older Private Branch Exchanges (PBXs) and network switches are phased out, industry is moving toward a voice networking model that is SIP signaled, IP based, and packet switched, not only in the wide area but also on the customer premises.

SIP supports five facets of establishing and terminating multimedia communications:

1. **User location:** Users can move to other locations and access their telephony or other application features from remote locations.
2. **User availability:** This step involves determination of the willingness of the called party to engage in communications.
3. **User capabilities**: In this step, the media and media parameters to be used are determined.
4. **Session setup**: Point-to-point and multiparty calls are set up, with agreed session parameters.
5. **Session management:** This step includes transfer and termination of sessions, modifying session parameters, and invoking services.

- SIP employs design elements developed for earlier protocols. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a client request that invokes a particular method, or function, on the server and at least one response. SIP uses most of the header fields, encoding rules, and status codes of HTTP. This provides a readable text-based format for displaying information. SIP incorporates the use of a Session Description Protocol (SDP), which defines session content using a set of types similar to those used in Multipurpose Internet Mail Extensions (MIME).

- Light and quick - SIP provides the same functionality as H323 for basic call setup, conferencing etc. but not all the extra overhead of the H323 protocol suite's(H245/H225 and Q931) terminal capabilities fields which most are not needed to achieve similar functionality of a complete corporate VoIP system.

- SIP often runs on top of the User Datagram Protocol (UDP) for performance reasons, and provides its own reliability mechanisms, but may also use TCP. If a secure, encrypted transport mechanism is desired, SIP messages may alternatively be carried over the Transport Layer Security (TLS) protocol.

- SIP is text based. Can be integrated into Web based telephony applications

- SIP is simple and does not require multiple protocols or a "suite" of protocols to operate.

- Simplifies design and crossing of stateful inspection devices such as firewalls and gateways. SIP Tunneling of UDP under NAT(STUN)

- SIP based calls can tolerate up to 64 seconds before timing out.

- Cisco Systems says the company is getting ready to add support for the Session Initiation Protocol (SIP) in the 5.0 release of Cisco Call Manager (CCM), the company's telephony server.

- At the date of this writing, Cisco is the only major networking vendor not to support SIP phones with its corporate telephony server. Both Avaya and Nortel Networks already deliver SIP support, although through proprietary extensions. Cisco phones must use Cisco's Skinny Client Control Protocol (SCCP), an H.323 derived protocol lacking presence, IM, and many of the convergence capabilities readily available in the SIP/SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) environments.

- Release 5.0 is due in the fall of next year. Cisco wouldn't confirm the report, saying that though Cisco has pledged SIP support, no public statements have been made regarding when that specific release of CCM will be

- There are improvements in H323 to SIP gateways being developed continuously.

- The National Emergency Number Association's(NENA) new Public Safety Answering Point (PSAP) for E911 services utilizes SIP.

**It is recommended** that NYC Utility should check with its VoIP vendor(Cisco) to see where they are headed with SIP.

However, since NYC Utility's current VoIP deployment is setup for H323 for Voice and Video NYC Utility can continue to use H323 for the following reasons

- **Configuration currently supported**
- **Knowledge of protocol**
- **PBX and gateway integration**
- **Interactive voice/video will still require H323**
- **Advances in H323 offer additional capabilities(Fast Start) and Video integration**

NYC Utility can find out more details about H323 and SIP protocols at:

**http://www.sipcenter.com/sip.nsf/index**

### *2.2.3 Video*

Two main types of video traffic "exist" Interactive-Video(videoconferencing) and Streaming Video. The following guidelines are recommended to be used for handling both types:

**Interactive-Video**

When planning for Interactive-video(video conferencing) traffic, the following guidelines are recommended:

Interactive-Video traffic should be marked **DSCP AF41**: Excess videoconferencing traffic can be marked down by a policer to **AF42 or AF43.**

- **Loss should be no more than 1 percent(like in the voice section earlier)**
- **One-way latency should be no more than 150ms(like voice)**
- **Jitter should be no more than 30ms**
- **Assign interactive-Video to either a preferential queue or a second priority queue(when supported). When using Cisco IOS LLQ, over provision the minimum-priority bandwidth guarantee to the seize of the videoconferencing session plus 20 percent. For example a 384kbs videoconferencing session requires 460kbs of guaranteed priority bandwidth.**

Because IP videoconferencing (IP/VC) includes a G.711 audio codec for voice, it has the same loss delay and delay-variation requirements as voice-but the traffic patterns of videoconferencing are very different form those of voice.

Be aware that videoconferencing uses varying packet sizes in its traffic flow and is extremely variable in terms of packet rates. Because IP/VC packet sizes and rates vary, the header overhead percentages also varies, so an absolute value of overhead cannot be calculated accurately for a streams. A protocol trace of NYC Utility's typical or planned IP/VC should be conducted to determine the general protocol and traffic behavior of the stream to help plan for the proper QoS bandwidth provisioning tool to use to ensure its quality. A general conservative rule of thumb for IP/VC bandwidth provisioning is to assign an LLQ band equivalent to the IP/VC rate plus 20%.

**Streaming-Video**

The following Cisco QoS Baseline guidelines are recommended when working with Streaming video:

- **Streaming Video (whether unicast or multicast) should be marked to DSCP CS4 as designated by Cisco's QoS Baseline standard.**
- **Loss should be no more than 5%**
- **Latency should be no more than 4 to 5 seconds (dependant on the video application's budder capabilities).**
- **There are no significant jitter requirements.**
- **Guaranteed bandwidth(CBWFQ) requirements depend on the encoding format and rate of the video stream.**
- **Streaming-Video is typically unidirectional; therefore, remote Loop sites might not require provisioning for Streaming-Video traffic on their WAN serial links or VPN links.**
- **Non compliant organizational Streaming video applications (either unicast or multicast) such as entertain video content, may be marked as scavenger DSCP CS1, provisioned in the Scavenger traffic class and assigned a minimal bandwidth (CBWFQ) percentage.**

Streaming Video applications have more lenient QoS requirements because they are not delay sensitive(the video can take several seconds to cue/buffer up)and are largely not jitter sensitive(because of the application buffering). However, Streaming-Video might contain valuable content, such as e-learning applications or multicast company meetings, which requires service guarantees.

It is under these guidelines and looser restrictions that, in NYC Utility's model, Streaming and Interactive Video can be put into one classification and that classification will be referred to just as VIDEO. This classification will have the same marking as the Cisco defined classification of **Interactive Video AF41**.

***Note:*** *For the remainder of this paper the classification of Video entails Interactive and Streaming unless noted otherwise.*

## *2.2.4 Control Plane QoS requirements*

It is critical to provision QoS for control-plane traffic such as IP routing protocol and network management protocols.

**IP routing protocols**

The following Cisco QoS baseline guidelines are recommended:

- **IP routing traffic should be marked to DSCP CS6. This is the default behavior on Cisco IOS platforms.**
- **IGP protocols such as EIGRP are protected an additional internal priority just inside the router via the PAK_Priority mechanism.**
- **EGP protocols such as BGP are recommended to have an explicit class for IP routing with a minimal bandwidth guarantee. Remember that BGP is TCP based so it has its own flow control and recovery capabilities.**

**DLSW+/RSRB Considerations**

Some companies use DLSW+ or RSRB to support legacy IBM equipment. NYC Utility does have some instances of RSRB in its enterprise. DLSW+ traffic for example, by default, is marked to IP Precedence 5(DSCP CS5). This default marking could interfere with VOIP provisioned traffic. RSRB does not mark TCP or FST based traffic for any IP precedence value.

Since RSRB traffic is not classified at the switch level **it is recommended** to classify and mark such traffic on the routers that support it. **The recommended marking** for RSRB will be **AF31** to match that of all other NYC Utility applications as it is in use today. However, if there is a need to provide a higher class of service to the RSRB peers than a different marking reflecting a higher class of service can be applied.

**Protocol Independent Multicasting (PIM)** packets are already marked to **DSCP 48** and processed accordingly at each router.

**Spanning Tree considerations**

NYC Utility follows a no loop architecture discipline which, by design, limits the use of spanning tree when possible. There will still be spanning tree functionality where the local users reside on their respective switch or switch stack plus any remaining trunk links between switches. Where QoS is of concern there is no option for the BPDU since these are layer 2 packets. However, **it is recommended that NYC Utility** consider reviewing moving their access layer switches or any switch still employing Spanning Tree to 802.1w, Rapid Spanning Tree(RST). RST will converge from a failed link within one second. Cisco also introduced PVRST(Per VLAN Rapid Spanning Tree) which runs RSTP without the more complex Multiple Spanning Tree(MSTP)802.1s.

RST is fast enough to preserve VoIP calls within the local switch, especially if the call is local between two users on the same switch and switch stack.

**It is recommended** that NYC Utility investigate with its vendor, Cisco, on advances in L2 recovery for its switch line. Extreme for example now has a SONET like automatic protection switching feature for its switches. Extreme calls it Ethernet Automatic Protection Switching(EAPS). EAPS provides sub 50ms failover, which is beneficial to voice calls. NYC Utility should check with Cisco to see if Cisco has something similar in the works.

### *2.2.5 Network Management*

The following Cisco QoS Baseline for Network management traffic is as follows:

- **Network-Management traffic should be marked to DSCP CS2**
- **Network-Management applications should be protected explicitly with a minimal bandwidth guarantee.**

Network management traffic is important in performing trend and capacity analyses and troubleshooting. Therefore, a separate minimal bandwidth queue can be provisioned for Network Management traffic, which could include SNMP, NTP, SYSLog, Concord, CiscoWorks and other management traffic.

### *2.2.6 QoS requirements for DATA*

NYC Utility utilizes many homegrown and commercial applications over its network. Data traffic characteristics vary from one application to another. It becomes more of a planning issue in regards to applications if the same application's traffic patterns and requirements can vary within different functions and from versions. A cited example of this is a case where a company had QoS deployed for VoIP and its mission critical application SAP. The SAP application was recently upgraded and the same function that required x amount of traffic/bandwidth in the old version somehow in the new release bloated up to 4 times its original amount. This was the normal behavior and traffic levels for this new release of the application. However, with the QoS model in place the SAP application users were complaining and it appeared that **"QoS was broken"**. It turned out that is was the new required upgraded application. The new SAP GUI was much larger than the older version. This example shows how when dealing with planning for QoS you must take into account your critical applications as well as voice. NYC Utility should be aware of such situations and the QoS model it uses must be flexible enough to make changes when applications change.

The Cisco QoS baseline model has five main classes for dealing with data traffic:

- **Locally Defined(company defined)**
- **Mission Critical Data**
- **Transactional Data/Interactive Data**
- **Bulk Data**
- **Best Effort and Scavenger**

These are reviewed in the following sections

### *2.2.7 Locally Defined Mission Critical Data*

The locally defined or (company defined) Mission critical class is probably the most misunderstood class specified in the Cisco QoS Baseline. This class is similar to that of the Transactional data class for applications. For example in the Transactional data class an SAP application may fall under it but in the Locally defined mission critical and more important revenue generating Oracle based application will be listed in this class. Because the admission criteria for this class is non technical(meaning it is determined by the business relevance and organizational objectives) applications may not always fit neatly into this class. Applications assigned to this "special" class can become an organizational and politically charged issue. **It is recommended** to assign as few applications to this class(from the Transactional class) as possible. **It is also recommended** that management or executive endorsement for application assignments to the Locally-Defined Mission Critical class be obtained. This paper outlines a QoS Application Inclusion/Removal policy in Appendix A. to prevent such events from happening.

The recommended Cisco QoS baseline for Locally Defined Mission-Critical Data is as follows:

- Locally-Defined Mission–Critical Data traffic should be marked to **DSCP AF31**

- Excessive Locally-Defined Mission Critical traffic can be marked down by a policer to **AF32** or **AF33**. However, Cisco IPT equipment is currently using DSCP **AF31** for Call signaling traffic. Until Cisco IPT equipment mark Call signaling to DSCP **CS3**, a temporary placeholder code point of DSCP **25** can be used to identify Locally Defined traffic. See the flexibility of the DiffServ model.

- Locally defined Mission-Critical Data traffic should have an adequate bandwidth guaranteed for the interactive, foreground operations that is supports

### *2.2.8 Transactional Data/Interactive Data*

The Transactional Data/Interactive Data class is a combination of two similar types of applications – Transactional Data classic Client/Server applications and interactive messaging applications. The response time requirements of Transactional based Data client server systems and generic client server systems are different and should be considered. The applications that fall into this category are SAP, Oracle, PeopleSoft, BEA et. al. Also, remember that there are two and three tier client server systems so NYC Utility must keep these types of applications architectures in mind. A question that arises is how QoS be handled in a 3 tier client/server system like SAP or the use of applications servers in front of the databases servers. This issue will be listed as a design consideration.

The recommended Cisco QoS Baseline for Transactional Data is as follows:

- Transactional Data traffic should be marked to **DSCP AF21**
- Excess Transactional Data traffic can be marked down by a policer to **AF22** or **AF23.**
- Transactional Data traffic should have an adequate bandwidth guarantee for the interactive, foreground operations that it supports.

Some defined Interactive type applications: Telnet, Citrix, Oracle Thin-clients, Instant Messenger, Netmeeting Whiteboard

Some defined Transactional type applications: SAP, People Soft, Oracle Financials, B2B, Supply Chain Management, Application Servers/proxies, Oracle Databases, Broadvision, IBM BUS, Microsoft SQL server. BEA Systems, DLSW+

### *2.2.9 Bulk Data*

The Bulk Data class is intended for applications that are relatively non-interactive and not drop sensitive, and that typically span their operations over a long period of time as background occurrences. Such applications included FTP, E-mail, backup operations, database synchronization/replication, video content distribution and any other application in which users typically cannot proceed because the are waiting for the completion of the operation.

The recommended Cisco QoS Baseline guidelines for dealing with Bulk data applications are:

- Bulk Data traffic should be marked to **DSCP AF11**.
- Excess Bulk Data traffic could be marked down to **AF12** or **AF13**
- Bulk Data traffic should have a moderate bandwidth guarantee but should be constrained from dominating a link.

Some defined Bulk Data application types include: Database syncs, network based backups, Lotus Notes, Microsoft Outlook, SMPT, POP3, IMAP, Exchange, Video content distribution and large FTP transfers

### *2.2.10 Best Effort data*

When addressing the QoS needs of Best Effort traffic the following Cisco QoS Baseline guidelines are recommended:

- Best Effort traffic should be marked to **DSCP 0**
- Enough bandwidth should be assigned to Best-Effort class as a whole because the majority of applications default to this class. **It is recommended** to reserve at least 25% of the bandwidth for Best Effort.

Some defined Best Effort application types are: All non-critical traffic, HTTP web browsing, other miscellaneous traffic.

### *2.2.11 Scavenger Class*

The Scavenger class is intended to provide deferential services or less than best effort services to certain applications. Applications assigned to this class have little or no contribution to the enterprise business objectives and are typically entertainment oriented in nature. These include peer-to-peer media applications like KaZaa, Napster and gamming applications like Doom etc.

The recommended Cisco QoS Baseline guidelines for Scavenger class traffic is as follows:

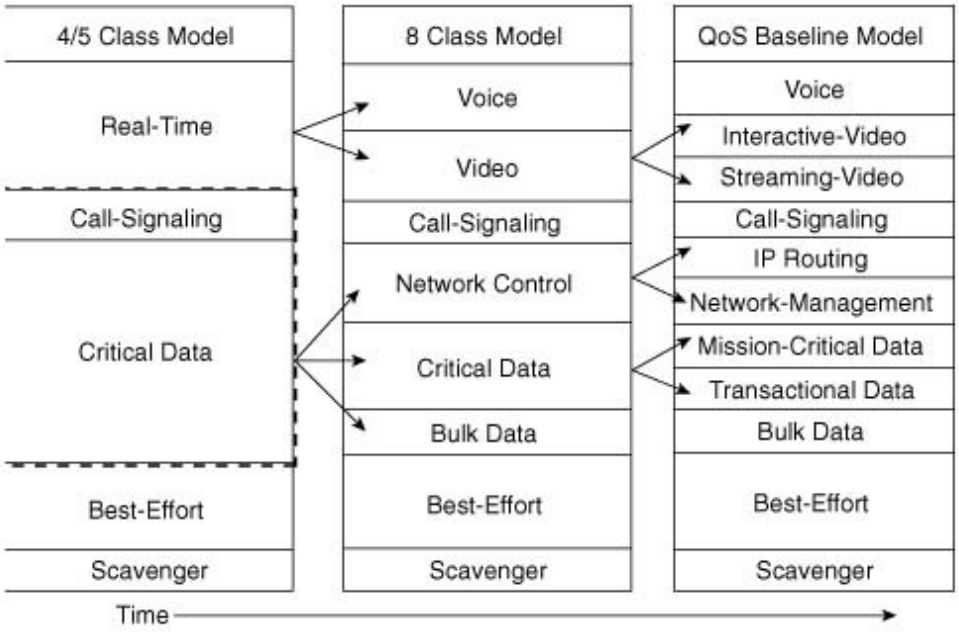
- Scavenger traffic should be marked to **DSCP CS1**

- Scavenger traffic should be assigned the lowest configurable queuing service. Assigning CBWFQ of 1 percent to Scavenger for example.

Assigning Scavenger traffic to minimal bandwidth queue forces it to be squelched to virtually nothing during periods of congestion, but it allows it to be available if bandwidth is not being used for business purposes during off peak hours. The Scavenger class is also used for the DoS and worm mitigation strategy component.

## *2.3 Cisco QoS Baseline Model*

**A diagram of Cisco's QoS Baseline Model in comparison with other generic models is below.**




The significance of the above diagram is that it shows that there is a progressive approach to obtaining a very fine and efficient QoS model, which is the Cisco QoS Baseline. NYC Utility could, for example start out with the QoS model on the left to keep things simple for an initial foray into QoS and then migrate to the Cisco QoS baseline on the right in the future. This paper again is to serve as a roadmap with approaches and configuration examples to help NYC Utility implement the first two models that reflect NYC Utility's goals and provide a foundation for the third "Cisco Baseline" model if they wish to move towards that model in the future. A summary of the Applications and their preferential treatment using the Cisco QoS baseline model is below. As you can see the marking and prioritizing of packets can become very granular.

| | |
|---|---|
| **Voice** | **EF** |
| **Interactive-Video** | **AF41** |
| **Streaming-Video** | **CS4** |
| **Call Signaling** | **DSCP 25** |
| **IP Routing/Network Control** | **CS6** |
| **Network Management** | **CS2** |
| **Mission-Critical Data** | **AF31** |
| **Transactional Data** | **AF21** |
| **Bulk Data** | **AF11** |
| **Best Effort** | **DSCP 0** |
| **Scavenger** | **CS1** |

# *3.0 NYC Utility QoS based Applications*

Now that the concept of marking packets for different levels of service and the QoS model to place those marked packets into has been discussed in section 2.0 we can now move into the section of identifying NYC Utility's applications requiring QoS, their profiles, behaviors and defining their class of service and where they fit into a QoS model.

## *3.1 Candidate Applications for QoS*

As of this writing NYC Utility has identified the following applications for inclusion into its initial QoS Model. The NYC Utility specific QoS model shall be discussed in more detail in the next section.

**Voice over IP(VoIP)**

- Cisco IP phones
- Softphones IP Communicator

**Voice Call signaling and CAC protocols from applications such as**

- Call Manger(publisher/subscriber)
- Cisco Voice Gateways software

**Interactive Video/Streaming video or just Video**

- Polycomm solutions in place
- Cisco's Meeting Place
- Microsoft's Netmeeting

**Control Plane/Routing protocols**

- EIGRP
- RIP(wherever left running)
- HSRP
- RSRB
- PIM
- Spanning Tree(if applicable)

**Network Management**

- SNMP Cisco based Ping and Tracerout traffic
- HP Openview
- CiscoWorks
- Insight Manager
- Magic trouble ticket system
- General DataComm racks
- Concord
- Cisco based Ping and Tracerout traffic
- SNMP Trap traffic

**Bulk Ddata**

- NYC Utility back office applications
- Custom built(home grown) applications
- FTP and FTP type transactions

**Best Effort**

- Web traffic and anything else

**Scavenger**

Traffic that is less than best effort but is marked so it can be classified and dropped when it is present in the company of the other applications listed above.

**It is recommended** that NYC Utility use and fill out the following application and traffic matrixes to outline the specifics of each applications traffic patterns and protocol behavior. This is required for the following reasons:

- **Outlines idiosyncrasies of the application/protocol that might have been missed**
- **Provides the needed information for the design remaining sections of this document and in the future when designing QoS solutions or implementing them**
- **Better facilitate the provisioning of the application/protocol into the proper service class**
- **Provides a single location of the current and future "provisioned" applications/protocols for inclusion into a specific class level.**
- **Provides NYC Utility a repository to add and remove provisioned applications/protocols and up to date documentation on what are NYC Utility's QoS applications**

## *3.2 NYC Utility QoS Toolset Matrixes*

The following matrixes outlined below are provided to assist NYC Utility in ongoing definition, classifying and identifying of certain application and platform QoS variables. Some of these matrixes are filled out from QoS testing and should be used in the future when applications and platform changes are made. The purpose of such matrixes is to provide NYC Utility a single repository of QoS variable information so NYC Utility can manage its current and future QoS solutions in an easier manner. **It is recommended** that NYC Utility update such matrixes and keep them in a public folder for the staff to reference.

### *3.2.1 Application/Traffic Profile Matrix.*

This matrix outlines the application's name layer 4 protocol used, what type of application(voice or data) and bandwidth requirements. **It is recommended** that to verify any application's profile before inclusion into the matrix a sniffer trace of a typical transaction of the application in use should be conducted and reviewed. This activity is required to ensure that the protocols, port numbers and packet flows are correct before committing to access-lists in the various class map configurations. The information in this and the following matrixes are critical to properly placing the right application or protocol into the proper QoS class level. If the application information is incorrect in this matrix it could be used incorrectly when configuring QoS on NYC Utility's network components, possibly causing application performance issues.

Application Inventory Matrix link

http://www.amilabs.com/NYutility/appinvmatrix.htm

### *3.2.2 Voice Codec Matrix*

Even though the information listed here is duplicated in the application matrix, this matrix is specific for just Voice Coded bandwidth requirements and is intended to provide a simple outline of Codec requirements when provisioning.

**NYC Utility's Codec requirements.**

| Bandwidth Consumption | Packetization Interval | Voice payload in bytes | Packets Per Second | Bandwidth per conversation |
|---|---|---|---|---|
| G.711 | 20ms | 160 | 50 | 80kbs |
| G729a | 20ms | 20 | 50 | 24kbs |

### *3.2.3 Application to DiffServe DSCP Matrix*

A quick outline of the DSCP markings for a specific application in use in the NYC Utility QoS model. Application DSCPs can be added, removed or changed in this matrix.

| Application Name | Protocol | DSCP Value | NYC Utility QoS Model Class |
|---|---|---|---|
| VoIP | RTP | EF | Real-Time |
| IP Communicator SP | RTP | EF | Real-Time |
| Call Manager | | | Call Signaling |
| Polycom | | | Real-Time |
| Meeting Place | | | Real-Time |
| Netmeeting | RTP | EF | Real-Time |
| Concord | SNMP | CS2 | Critical Data/Network Management |
| HP OpenView | SNMP | CS2 | Critical Data/Network Management |
| CiscoWorks | SNMP | CS2 | Critical Data/Network Management |
| GDC T-1 Racks | SNMP | CS2 | Critical Data/Network Management |
| Magic Trouble Ticket | Email based | AF11 | Critical Data/Bulk class |
| FTP | TCP | AF11 | Critical Data/Bulk class |
| Netmotion | ??? | AF31 | Critical Data/General applications |
| General Back office apps. | IP/TCP/UDP | AF31 | Critical Data/General applications |
| | | | |

### *3.2.4 Network component interface Queue matrix*

**Note:** *it is assumed that the reader is already familiar with the Catalyst series of Queuing nomenclature. This matrix is helpful for the configuration of different platform queuing allotments.*

| Router/Switch Platform Interface | Minimum Default Egress Queue Setup | Scheduler Algorithim |
|---|---|---|
| **Fast Ethernet on 3750** | **1P3Q3T** | **SRR/WTD** |
| **Gigabit Ethernet on 3750** | **1P3Q3T** | **SRR/WTD** |
| **Fast Ethernet on 3550** | **1P3Q1T** | **WRED/TD** |
| **Gigabit Ethernet on 3550** | **1P3Q2T** | **WRED/TD** |
| **Fast Ethernet on 4500** | **1P3Q1T** | **DBL/RR** |
| **Gigabit Ethernet on 4500** | **1P3Q1T** | **DBL/RR** |
| **Fast Ethernet#1 on 6500** | **See CAT6500 Matrix** | **WRED/TD** |
| **Fast Ethernet#2 on 6500** | **See CAT6500 Matrix** | **WRED/TD** |
| **Gigabit Ethernet #1 on 6500** | **See CAT6500 Matrix** | **WRED/TD** |
| **Gigabit Ethernet #2 on 6500** | **See CAT6500 Matrix** | **WRED/TD** |
| **Trunk Links on 6500 to 6500** | **See CAT6500 Matrix** | **WRED/TD** |
| **Trunk Links on 6500 to 5500** | **See CAT6500 Matrix** | **WRED/TD** |
| **Gigabit Ethernet on 10702** | **LLQ** | **PXF-FIFO/CBWF** |
| **SRP on 10702** | **LLQ** | **PXF-FIFO/CBWF A and B side** |
| **Fast Ethernet on 7500** | **LLQ** | **CBWF/LLQ** |
| **Gigabit Ethernet on 7500** | **LLQ** | **CBWF/LLQ** |
| **Serial on 7500** | **LLQ** | **CBWF/LLQ//SHAPING** |
| **Serial on 7300** | **LLQ** | **CBWF/LLQ//SHAPING** |
| **Fast Ethernet on 7300** | **LLQ** | **CBWF/LLQ** |
| **Serial on 2600** | **LLQ** | **CBWF/LLQ//SHAPING** |
| **Fast Ethernet on 2600** | **LLQ** | **CBWF/LLQ** |
| **Serial on 3800** | **LLQ** | **CBWF/LLQ//SHAPING** |
| **Fast Ethernet on 3800** | **LLQ** | **CBWF/LLQ** |
| **Gigabit Ethernet on 3800** | **LLQ** | **CBWF/LLQ** |
| **Serial on 2800** | **LLQ** | **CBWF/LLQ//SHAPING** |
| | | |
| | | |

### *3.2.5 Catalyst 6500 Platform Line Card Queue matrix*

***Appendix F.*** provides the following matrix that outlines the Catalyst 6500 based production switches in NYC Utility and their Line Card Queue allotments.

### *3.2.6 Application QoS Inclusion/Removal Policies*

A procedure is required to prevent the eventuality of all or a large majority of the applications to become handled at the same QoS level thus turning the network into a non-QoS environment even when QoS is deployed.

NYC Utility needs to implement an inclusion policy for new or promote applications from one class of service to another. This inclusion or promotion policy should outline a process for a request to be made from the application owners to NY Utility network support staff and meet a certain criteria to be included or promoted to a particular class or be promoted from one class to another. Subsequently the network support staff should have a notification policy for any application that is to be demoted in class based on a similar criterion.

**A recommended** draft policy that NYC Utility should consider and tailor to its own needs is outlined in ***Appendix A.***

## *4.0 QoS Hardware and Software IOS platforms*

This section outlines the hardware and software IOS platforms that NYC Utility currently has today to implement QoS and what is required of any platform to be QoS capable. This section has recommendations pertaining to each platform as well as a summary of recommendations at the end of this section.

### *4.1 What NYC Utility has today to achieve QoS*

NYC Utility currently has a variety of Cisco based hardware and software platforms in place throughout its enterprise. These different platforms pose a challenge in accomplishing NYC Utility's ultimate goal of providing an End-to-End QoS solution. The reason why it is a challenge is that different Cisco hardware and software platforms perform the same QoS functions differently. For example Queuing structures and scheduling is handled differently from the 3750 platform, 6500 platform and router platforms. Queuing is handled in ASIC hardware on some platforms and software on others. Classifying and marking packets for a QoS class level is different between the platforms is another example. Another example is that different Catalyst PFC and MSFC set L2 and L3 packet sizes inconsistently. The Catalyst 6500 MSFC or Sup720 supports NBAR while the 3750 does not. There is also the issue of same platform but different software versions affecting the way QoS is handled policing and queuing methods, options and advancements differ from feature to feature. For example, LLQ policing is handled consistently across platforms starting with IOS 12.2.

A rich set of QoS features are available on Cisco Routers in IOS. These features are flexible to use since they are software based. However, software based features could entail a CPU tax when they are implemented. With line rates of the media to which the policies are applied, implementing such policies in software at Fast, Gigabit and 10Gigabit Ethernet speeds would impair any Cisco hardware platform. Therefore Cisco has ported QoS logic from IOS to hardware ASICs to provision QoS policies at line rates within campus environments. The benefits of porting some of the IOS QoS capabilities to ASICs is as follows:

- **Classification and marking can be offloaded from routers and moved as close to the source host as possible.**
- **Trust boundaries can be defined and enforced**
- **Policing can be performed right at the source for marking down traffic or dropping unwanted traffic.**
- **Increased policing support such as per port/per Vlan and micro flow policing.**
- **Real-time applications such as voice can be guaranteed within campus environments because of preferential/priority hardware queuing**
- **Congestion can be avoided within the campus using WRED or similar tools**

Also, the main caveat that arises form the porting of software(IOS) QoS to hardware is that QoS features become hardware specific – that is they can vary from platform to platform and even from line card to line card. This increases the complexity of configuring and managing campus QoS because many idiosyncrasies need to be kept in mind for each platform or line card. This caveat is further exacerbated because some platforms require CatOS, others require IOS and still others have both. This is the case within NYC Utility. To address this issue common syntaxes such as the MQC from Cisco IOS and common macros like Auto QoS are continuing to be developed to make QoS consistent and easier to deploy across Catalyst platforms in the future.

The Cisco Quality of Service (QoS) Behavioral Model is the conceptual framework underlying the **MQC -- the Modular Quality of Service (QoS) Command-Line Interface (CLI)** -- which is the configuration language used to implement traffic classification and QoS policy actions on Cisco routers and switches. The model -- and the MQC -- provides a way to implement QoS consistently on Cisco routers and switches, irrespective of the implementation details of those platforms.

What this means is that NYC Utility must carefully select its network component's platform QoS tools that meet its needs and **ENSURE** that the software and hardware platforms used are consistent, otherwise unpredictable issues can occur or intermittent issues relating to an application may be masked by something as minor as the amount of queue space on an interface is different between hardware platforms and software versions.

## *4.2 NYC Utility's Platform inventory*

### *4.2.1 Access-layer Catalyst 3750*

IOS based platform that supports the MQC but in limited capability compared to a router's IOS or a MSFC/Sup720. Most of the QoS capabilities are in the ASICs giving the platform the performance it needs handling the overhead of QoS at the port level but do not provide features like NBAR or Percentage keyword based policing available thus making the configurations for classifying, marking and policing a little more difficult to create and maintain.

The software version on the 3750s in production are currently 12.1(19) EMI.
Tests conducted in the lab with this hardware and software platform indicate that an IOS upgrade is necessary for this switch. When removing a policy map statement the switched crashed. **NYC Utility uses 24 and 48 port 3750 as its standard access platform**

### *4.2.2 Access-layer Catalyst 3550*

IOS based platform that supports the MQC but in limited capabilities compared to a router's IOS or a MSFC/Sup720. Most of the QoS capabilities are in the ASICs giving the platform the performance it needs handling the overhead of QoS at the port level but do not provide features like NBAR or Percentage keyword based policing are not available thus making the configurations for classifying, marking and policing more difficult to create and maintain. Some features are the same as the 3570 like the Class and Policy map statements but others are different like where the priority queue resides(See the NYC Utility Access Layer Model matrixes to see the queuing differences). NYC Utility uses 24 and 48 port 3550 as its standard access platform.

### *4.2.3 Access-layer Catalyst 2900 Series 2924*

These switches do not support QoS just beyond basic interface queues and do not have MQC. These switches are deemed legacy and are not considered part of the NYC Utility End-to-End QoS solution. **It is recommended** that these switches be replaced or the classification and marking boundary moved to the next level up from these switches if these switches are connected to 3550s or 3750s.

### *4.2.4 Access/Distribution Layer Catalyst 4500 Series*

These switches run IOS on the Supervisor IV and not CatOS which means that these switches have an almost full set of IOS based QoS options, including MQC. However, NBAR is not available on the version in production today 12.1(20) and (19). These switches support a consistent MQC and QoS feature set similar to those from the 3550, 3750 and 6500. There is no MLS command prefix for the QoS commands but the same commands on the 3550 and 3750 platforms do apply. The catalyst 4500 series handles queuing and congestion avoidance differently than others in the Catalyst family. The 4500 uses dynamic buffer limiting (DBL). Also the priority queue is set at 3.(Refer to the NYC Utility Access Layer Model matrixes). These switches may also be deployed in a Core capacity like the switch located in DC1 and DC2 and thus will have its QoS configuration similar to that of a 6500 switch.

### *4.2.5 Core-layer Catalyst 5500*

These switches do support QoS but in a very limited manner and do not support MQC. The QoS capabilities come in the form of various interface priority queues and MLS capabilities in the Supervisor 3 and NFFC. Most of the QoS capabilities are in the ASICs giving the platform the performance it needs handling the overhead of QoS at the port level. However, even with the latest software(which is CatOS based), this platform just does not scale well and its different QoS attributes offer no advantage to utilizing them. These switches are considered legacy.

### *4.2.6 Core-Layer Catalyst 6500*

The catalyst 6500 series of switches support a full array of QoS capabilities. They support the MQC and have many hardware and software options to enhance the capabilities of QoS. However, as stated at the beginning of this section such capabilities vary from software platform and version. For example the CatOS version of the software supports QoS capabilities in the ASICs giving the platform the performance it needs handling the overhead of QoS at the port level. ASIC based queuing and scheduling is very fast, but not very flexible in terms of configuration, maintenance and troubleshooting. When configured as an access-layer switch, the preferred software in the supervisor is CatOS; when configured as a distribution or core layer switch, the preferred software is Cisco IOS. **It is recommended** that all Core Layer 6500s and any 6500s acting in a distribution or special function such as in the Control Centers or Server Farms be upgraded to Native Cisco IOS on the switches to ensure consistency of services and provide QoS capabilities to these areas when needed in the future..

The use of MSFCs and Sup 720s in combination are adequate for QoS today. There is no immediate need to upgrade all Catalysts 6500s MSFCs to Supervisor 720s for QoS needs

today. As long as both MSFCs and Supervisor 720 are running in Native mode the QoS capabilities for these platforms will be available and consistent.

The Catalyst 6500 switches in production today are still in the 6.4 line of code. Catalyst Hybrid or CatOS is currently up to 8.4. By having the switches remain in Hybrid mode NYC Utility must then manage two different sets of QoS configurations. The consistency in terms of configuration commands, management and troubleshooting across platforms become separated. It is for this main reason that **it is recommended** the switches be upgraded to full MSFC Native IOS or Sup720 Native IOS mode for consistency among the NYC Utility QoS Tool Set commands.

Also, Native mode IOS provides Catalyst 6500 based switches with Generic Online Diagnostics(GOLD), Nonstop Forwarding(NSF) and Stateful Switchover(SSO) if redundant supervisors are utilized. Finally, the Native mode Catalyst 6500 running IOS on a supervisor 720 module offers the richest set of QoS features equaling that of a router. This includes the MQC and NBAR for classification. The CatOS version running is 6.5 while the IOS versions on the MSFC and 720 supervisors differ. The PFCs are used for classifying, marking mapping and policing packets and the individual line cards handle the queuing and dropping of packets. Not all line cards support DSCP or IPP trusting. Plus different line cards have different queue setups.

The NYC Utility Catalyst 6500 based line card port queuing matrix is outlined in ***Appendix F.***

**It is recommended** that NYC Utility investigate the use of GOLD, NSF and SSO on the switches and routers that support these features for added failover capabilities.

***Note:*** *as of this writing NBAR is currently supported on the MSFC for Vlan and Routed interfaces. NBAR can be loaded on a Supervisor 720 and activated on an interface but there will be no statistics gathered. Cisco is aware of this and will provide details on NBAR support for the Supervisor 720 in future releases.*

### *4.2.7 Core-Layer Cisco 7500 Wan Routers*

These routers support a full array of QoS features including advanced features such as MQC, NBAR and RSVP. For the backbone links there is a large amount of configuration options available. For the serial links there is a rich set of QoS policing and shaping tools as well as RSVP for CAC and call provisioning. The features available on this platform are similar if not the exact of those from the MSFC or Superisor720 IOS platforms. The only requirement of these routers to be considered is the RSP level, memory and IOS version, to run the NYC Utility QoS model's configuration commands. This router is **NOT** Cisco Survivable Remote Site Telephony (SRST) capable.

### *4.2.8 Core-Layer 7200 Series Router*

The 7200 series can support a full array of QoS features including MQC and NBAR. These routers will have the same IOS capabilities in general and pertaining to QoS as that of the 7500. In this context the features are about the same in software but the platforms processing and interfaces just differ. The only requirement here is the 7200s require to run QoS need a memory upgrade. This router is Cisco Survivable Remote Site Telephony (SRST) capable.

The 7200VXR are not candidates for QoS since they support Token-Ring and ESCONN connections and will be replaced. There is no need to increase the complexity of QoS by supporting another unique configuration and IOS version for this platform.

### *4.2.9 Core-Layer Cisco 10700 DPT Router*

The Cisco 10720 Internet Router provides IP services to users at optical speeds at the edge or core of their networks. The Cisco 10720 Internet Router provides network access using Ethernet and Dynamic Packet Transport (DPT), Packet over SONET (POS), or IEEE 802.17 RPR technology for optical connectivity. This router runs IOS and MQC but at the present time does not provide advanced QoS features such as NBAR and CoS to DSCP mapping functions. NYC Utility currently has 20 of these routers deployed(according to CiscoWorks). **It is recommended** that QoS options defined for these routers are tested off line on lab versions or at Cisco local office lab.

**Note:** Completed QoS testing notes for this router are included in Appendix C.

**A special note about the SRP.** The Core layer utilizes SRP which is a MAC layer protocol for destination stripped based rings. The reason SRP is mentioned here is that the protocol utilizes a priority mechanism and automatically reads the bits from an IP headers precedence bits to prioritize packets. The following information is from a SRP white paper explaining SRP Packet Priority functions.

SRP provides support for packet prioritization and expedited priority packet handling for the transmit queue and transit buffer. The motivation for this is to provide support for real time applications (such as voice and video over IP), mission critical applications and control traffic which have strict delay bounds, jitter constraints, and therefore require expedited handling. The priority field in the SRP MAC header is set by the node sourcing the packet onto the ring. The node utilizes a mapping between the value of the precedence bits in the Type of Service (ToS) field of the IP header into the priority field of the SRP MAC header.

There are eight levels (3 bits) of priority in IP and only two priority queues (high and low) in SRP. To handle this, the node utilizes a configurable priority threshold value to determine if the packet should be placed in the high- or low-priority transmit or transit queues. For packets transiting a node, the priority field in the SRP MAC header is inspected and the packet is then placed in either the high- or low-priority transit buffer based on the value of the configured priority threshold. Output scheduling is then determined by the transmit-side packet processing algorithms discussed below.

In order to choose the next packet to transmit, the scheduler must choose between high- and low-priority transit packets and high- and low-priority transmit packets according to the following principles:

- Respect packet priority by scheduling high-priority packets before low-priority packets.

- Enforce ring packet conservation by avoiding discarding packets which are already in circulation around the ring.

The rules are enforced via the following packet-handling hierarchy:

**1. High-priority transit packets**
**2. High-priority transmit packets from host**
**3. Low-priority transmit packet from host**
**4. Low-priority transit packets**

However, the packet priority hierarchy is modified by placing thresholds on the low-priority transit queue depth to ensure that:

1. The transit buffer does not overflow while serving locally sourced traffic.

2. The low-priority transit traffic does not wait too long behind locally sourced low-priority traffic.

High-priority transit packets are always sent first. High-priority transmit packets are sent as long as the low-priority transit buffer depth is less than high threshold. Low-priority transmit packets are sent as long as the low-priority transit buffer depth is less than low threshold and MY_USAGE is less than ALLOW_USAGE. If nothing else can be sent, the low-priority transit packets are sent.

Packets with an IPP already marked such as EIGRP will work fine with this automatic IP to MAC transmit priority on the SRP rings. However, packets marked with DSCP markings will need to be mapped somehow so end-to-end high priority packets traversing the DPT ring don't end up with a lower priority while on the ring or lose their marking when exiting the ring. Section 6.2.1 provides a table of DSCP range values to IPP to maintain such a linkage between the two.

NYC Utility will have to test on a pair of DPT routers whether DSCP markings can be re-marked on egress of the DPT ring to ensure true end-to-end consistent packet markings so QoS policies will operate as well in a consistent manner. NYC Utility needs to ensure that the DPT ring's SRP does not break QoS in any way. The SRP-fa only applies to low-priority packets. High-priority packets do not follow the SRP-fa rules and can be transmitted at any time as long as there is sufficient transit buffer space. High-priority packets can be rate limited, with features like committed access rate (CAR) before they are sourced onto the ring. However, it must be noted that in many Cisco texts CAR is a legacy QoS tool. **It is recommended** that NYC Utility test any Policers or Shaper(if required for use in the future) on a test SRP interface before applying onto a production router.

The default behavior of the 10720 regarding QoS is to set the SRP COS value to 0 for all traffic. This subsequently means that the 10720 will only forward low-priority traffic in default mode. The exception to this behavior is routing protocols (pak priority) and L2 protocols (IPS and topology packets). Please note that this default behavior is independent of layer 3 QoS values and in default mode the layer 3 QoS values are not touched. The 10720 offers via MQC options to explicitly assign a distinct SRP COS value to a given output queue via the MQC bandwidth statement. On 10720 there is no **automatic** mapping from IP precedence to srp-priority. If the configured COS value is above the SRP discriminator (default = 5) the traffic goes out a high otherwise it is low. If traffic is assigned via MQC to a "priority" queue the COS value will be 7 and will go out as high priority. You can adjust how the low and high queues are assigned by using the **"set srp-priority"** command which will be outlined in ***Appendix D.*** A WRAP situation will not change how QOS is handled either.

**It is recommended** that NYC Utility test the behavior of the IPP to SRP priority to verify if packets marked with a DSCP value fall into a DSCP to IPP range and SRP handles these packets accordingly. Or, test configurations to map DSCP values to IPP on egress SRP interfaces and verify that SRP prioritize the packets accordingly and that upon exit of the DPT ring the IP packet retains its DSCP value.

**It is recommended** that NYC Utility test its QoS and VoIP deployments with a wrapped ring condition to ensure that end-to-end delay(propagation and jitter) budgets can survive a core in a wrapped state. This is to ensure that the priority packets still get forwarded accordingly and are not subject to the fairness algorithm when in wrapped mode. This can be tested off hours manually by issuing a Forced Switched(FS) or Manual Switched(MS) CLI command..

### *4.2.10 Cisco 3800 Router*

The Cisco 3800 router is a newer class of distribution and access layer router that supports the latest in IOS innovations and hardware innovations such as embedded cipher modules. These routers run the full gamut of QoS options and provide years of future service. NYC Utility will be deploying these routers in the field to replace older 3600 and 2600 series. This router is Cisco Survivable Remote Site Telephony (SRST) capable.

### *4.2.11 Cisco 2800 Router*

The Cisco 2800 router is a newer class of distribution and access layer router that supports the latest in IOS innovations and hardware innovations such as embedded cipher modules. These routers can run the full gamut of QoS options easily and provide years of future service. NYC Utility will be deploying these routers in the field to replace older 2600 and possibly any 2500 series. This router is Cisco Survivable Remote Site Telephony (SRST) capable.

Also, for sites that may employ the use of a 2800/3800 router with a built in switch module so a 3750 does not have to be deployed to the site, the built in switch module cannot perform the same QoS functions and have a similar configuration applied to it as if there were a 3750 switch in place. A policy map was applied to the interface(just like an access-layer switch) to mark packets form the built in switch port. This function had no affect and packets were not marked from the switch port. However, a service policy applied to the router interface to mark the packet on the way into this interface worked fine. This shows that this type of platform(router with built in switch module) can implement QoS at the router level. **Remember** that the switch module, depending on the hardware update, does not support inline power for phones. Cisco may have an updated module to support inline power. The switch module was tested in the lab with a 3700 router to confirm the behavior outlined above.

***Note:*** *NYC Utility will also upgrade any future VoIP required site to using a 3800 or 2800 router and a 3750 switch.*

### *4.2.12 Cisco 2600 Series Router*

There are still plenty of Cisco 2600 routers deployed in the NYC Utility enterprise. Some are of XM class and others are from the original series. These routers do support CEF and MQC so they can run a full set of QoS options like NBAR. These routers are not as powerful in terms of CPU performance but with enough memory implementing QoS should not pose any issues. **It is recommended** that based on the IOS version requirements for NYC Utility's QoS solution that the 2600 routers have enough memory to facilitate features such as MQC and NBAR. This router is Cisco Survivable Remote Site Telephony (SRST) capable. The minimum platform is 12.2(15)T with 48mb of ram and 16 mb of flash memory. **It is recommended** that if 2600 routers are used in a path between QoS endpoints that the router be of XM class. Non XM class lack the performance and feature options for QoS.

### *4.2.13 Cisco 3600 Series Routers*

These routers are, just like the 2600, capable of running CEF and a full set of IOS QoS options depending on the version. However, this platform has been know to have hardware problems and is end of sale life unless, it is an A model. Even though this platform has the processor capabilities to run QoS **it is recommended** that unless absolutely necessary 3600 routers should be replaced with 3700/3800/2800 routers before QoS is to be deployed to them. Cisco's recommended replacements for the Cisco 3640 are the Cisco 3700 Series Routers. On an interim basis, Cisco will make the 3640A available as an alternative for customers with configurations that are not yet supported on the 3700. The 3600 router is **NOT** Cisco Survivable Remote Site Telephony (SRST) capable. The 3700 **is** SRST capable. The 3700 series routers may contain built in switch modules thus providing access layer switching within the switch itself without the need for an external switch. For 3700 routes that fall within this category the MQC can be applied to the local switch interfaces on the router.

The standard access-layer platform that NYC Utility is moving towards is 3750 switches(EMI or LMI) at the access layer and Catalyst 6500 in the core/backbone in each major business office. For remote sites the use of 3750 switches and either 2800 or 3800 series routers will also be used. Core Wan routers will continue to be 7500 and 7200 series.

The 2600 routers will still continue to be utilized and do provide enough QoS capabilities to support the needs of NYC Utility but are limited in processing abilities compared to the 2800 and 3800 series. **It is recommended** that any site running a 2600 and older switch combination that requires QoS for any of the QoS applications should be reviewed to determine if a forklift upgrade is necessary or if the in place equipment can handle the requirements.

### *4.2.14 2500 Series Routers*

The 2500 series routers are considered legacy and end of life product. These routers, though resilient and have served NYC Utility very well for many years, do not provide the capabilities to support QoS and any advanced features. **It is recommended** to replace any or all 2500 routers in between any QoS based path with an updated platform.

## 4.3 Software Platforms

A majority of the QoS features on the Catalyst switches are performed in hardware ASICs. However, there features based on switch version that are required to be present to help keep the configurations and functionality across the enterprise consistent.

- **MQC - all switches at a minimum must be able to support this functionality. This leaves out the 2900 and the 5500 series(the 5500 may be able to support MQC on the RSM but that just adds to the list of yet another IOS support version and possibly different configuration supported)**

- **MSFCs and Supervisor 720s - Support IOS software and can apply all the features of MQC as well as NBAR for additional QoS classification and security uses.**

- **Using the MSFC and Supervisor 720 as an additional QoS aggregation point for easier classification and bandwidth reservation.**

- **Using the MSFC and Supervisor 720 for DOS worm mitigation by implementing NBAR on them.**

One thing to keep in mind when reading the next paragraph on IOS software for the routers is that the features that you have available on the router IOS may not always be available on MSFC and Supervisor 720. However, having IOS used on both routers and switches does provide a consistent set of tools if like features are available and used.

Unlike Catalyst queuing, which is done in hardware, WAN router QoS is performed in Cisco IOS software. In this text when referring to WAN routers it is implied that the loop routers, core WAN routers and DPT routers.

If the WAN router is serving many remote branch site routers, the collective CPU required to administer complex QoS policies might be more than some older device can provide. This is not the case with the loop routers which only use two connections. Also, the Core WAN routers don't have as many serial links connected to them as in the past as a result of the migration of major business office sites to the DPT ring. The main consideration to keep in mind is that QoS entails a marginal CPU load, WAN links and QoS policies should be designed to limit the average CPU utilization on the wan router.

Cisco's own testing has shown a significant decrease in data application response times when Real-time traffic exceeds one third of a link's bandwidth capacity. Extensive testing and production-network customer deployments have shown that limiting the sum of all LLQs to thirty three percent is a conservative and safe design ratio for merging real-time applications with data applications over the WAN links.
Recent versions of Cisco IOS software automatically size the final interface out buffer **(Tx-ring)** to optimal lengths for Real-Time applications such as Voice or Video.

On some older version of Cisco IOS software, **Tx-rings** might need to be reduced on slow speed links to avoid excessive serialization delay.

It is with this functionality that LLQ policing is handled consistently across platforms starting with a minimal version of IOS 12.2. Also, with release 12.2(2)T introduced the capability to mark voice sourced packets on the voice gateway with DSCPs.

For all WAN routers **it is recommended** that the minimum IOS version be the 12.2(2)T or later. The IOS version selected should support Cisco's Service Assurance Agent(SAA) capabilities for enhanced production level testing and troubleshooting services. 12.2(2)T provides the capability to mark voice-sourced packets on the voice gateway with DSCP, plus the capability to mark signaling packets separate from voice or video packets. Traffic can be marked at the source and this conducive for SRST or CME implementations.

**It is recommended** that all routers, regardless of platform, must at a minimum run **CEF** and be MQC and DiffServ aware/capable

| | IOS T FITS NYC Utility | IOS S NOT Pertinent to NYC Utility | IOS XR NOT Pertinent to NYC Utility |
|---|---|---|---|
| Target Networks | • Access<br>• Enterprise<br>• Managed CPE/WAN edge | • Large enterprise core<br>• Service provider edge<br>• Service provider core (today) | • Service provider core<br>• (near term)<br>• Service provider edge (future) |
| Key Attributes | • Broad platform, support/small footprint<br>• **Integrated security, voice, QoS**<br>• Flexible feature-option packaging | • Enhanced scalability, availability, security<br>• Optimized for critical enterprise core and service provider edge networks<br>• Broad feature set for enabling flexible service-delivery | • Terabit-scale core IP/MPLS routing<br>• Unprecedented scalability and performance<br>• Continuous system operation<br>• Exceptional service flexibility |
| Target Applications | • **Firewall, intrusion** detection<br>• **IP telephony**<br>• **Wireless networking**<br>• **QoS**<br>• IPv4 and IPv6 routing | • High-end platform support<br>• Core and edge IP/MPLS routing<br>• MPLS VPNs<br>• Any Transport over MPLS (ATOM)<br>• Enterprise core infrastructures | • Core IP/MPLS routing<br>• Large-scale peering<br>• POP consolidation<br>• Converged infrastructures<br>• Continuous system operation |

**It is recommended** that NYC Utility consider looking into migrating to the 12.3T releases not only for QoS needs but for these other features that could be coupled with QoS.

- **Secured Cisco Survivable Remote Site Telephony (SRST)**
- **Network Admission Control (NAC) 12.3(8)T**
- **Inline Intrusion Prevention System 12.3(8)T**
- **Easy Secure Device Deployment (EzSDD) 12.3(8)T**
- **Optimized Edge Routing (OER) 12.3(8)T**
- **Transparent Firewall 12.3(7)T**
- **IPv6 Firewall 12.3(7)T**
- **PKI Certificate Authority (CA) 12.3(4)T**
- **Online Certificate Status Protocol (OCSP) 12.3(2)T**
- **Warm Reload 12.3(2)T**
- **PKI-AAA 12.3(1)**
- **Advances in IntServ/DiffServ integration.**
- **Stateful switchover**

### *4.3.1 NBAR*

Cisco Network-Based Application Recognition (NBAR) is an intelligent application classification engine within Cisco IOS Software that uses deep, stateful packet inspection to recognize a wide variety of applications and protocols, including Web-based and other difficult-to-classify protocols that use dynamic TCP/UDP port assignments. NBAR looks into the TCP/UDP payload and classifies packets based on payload characteristics such as transaction identifier, message type, or other similar data. NBAR in its use for QoS classification of traffic for marking eases the administrator's need for using access lists and port numbers to identify and classify application packets.

NBAR can also detect worms in the packet payload. NBAR plays an important role in threat mitigation because it works with QoS features to block or rate limit network resources to undesirable traffic.

Once a match value unique to the attack is identified, deploying NBAR can be an effective, tactical first step to block malicious worms while you are busy patching the network to establish a defenses against the attack. For example, with Code Red, you can use a match on "*.ida" URL in the HTTP GET request. With Blaster, you can look for SQL packets of a specific length. NBAR uses regular expression matching to classify traffic by URL, text, or host fields within a HTTP request. The HTTP subport classification capability in NBAR classifies all Code Red virus packets by locking on HTTP GET requests looking for a file with the "*.ida" extension.

**Starting with Cisco IOS Software Release 12.3T,** Cisco IOS NBAR recognizes nearly 100 different protocols and applications. New application support for NBAR can easily be delivered through a protocol description language module (PDLM). Written by Cisco engineers, PDLMs contain the rules used by NBAR to recognize an application; they can usually be loaded at run time without an IOS upgrade or router reboot.

The NBAR User-Defined Custom Application Classification feature gives you the ability to define customized protocols over a range of TCP and UDP ports and inspect the packet payload for a matching signature pattern at a known offset in a specific traffic flow direction. The custom protocol capability in NBAR can be used to classify the SQL Slammer worm, and an associated QoS drop action ensures that the packet is discarded before reaching the server. ***It is recommended*** that NBAR be deployed on all switch and router platforms that can support it for current and future QoS/Security uses.

**It is also recommended** that NYC Utility utilize IOS versions on routers and switches that support SAA. An excerpt form Cisco's website outlines SAA and its uses.

Service Assurance Agent (SAA) is embedded software within Cisco IOS devices that performs active monitoring. Active monitoring is the generation and analysis of traffic to measure performance between Cisco IOS devices or between Cisco IOS devices and network application servers. Active monitoring provides a unique set of performance measurements: network delay or latency, packet loss, network delay variation (jitter), availability, one-way latency, website download time, as well as other network statistics. SAA can be used to measure network health, verify service level agreements, assist with network troubleshooting, and plan network infrastructure. SAA is supported on almost all Cisco IOS devices. Enterprises and service providers routinely deploy SAA for network performance statistics and within IP networks that utilize quality of service (QoS), Voice over IP, security, Virtual Private Network (VPNs), and Multiprotocol Label Switching (MPLS). SAA provides a scalable and cost effective solution for IP service level monitoring and eliminates the deployment of dedicated active monitoring devices by including the "probe" capabilities within Cisco IOS.

NYC Utility can benefit from this **"built in"** capability today for its QoS and VoIP deployments and in the future for other applications such as Video.

Now that the Application classifications, QoS models and the relevant hardware and software platforms that can run QoS have been identified, the following sections cover the specific details of the recommended QoS strategy that NYC Utility can follow.

### *4.3.2 Recommended Software Platforms for QoS*

The recommended software platforms for all platform devices that will implement QoS is outlined below in a matrix for easy reference.

| Hardware platform | Software class | Version | Comment |
|---|---|---|---|
| | | | |
| Catalyst 3550 | IOS | 12.1.14 SMI | 12.2(25)SEA EMI tested |
| Catalyst 3750 | IOS | 12.1.19 SMI | 12.2(25)SEA tested |
| Catalyst 4500 | IOS | 12.1(20)EW2 | |
| Catalyst 5500 | CAT OS | N/A | Not applicable |
| Catalyst 6500 | IOS MSFT Native | 12.1(26)E | c6Sup2_RP-JS-M |
| Catalyst 6500 | IOS 720 Native | 12.2(18)SXD3 | s72033_rp-JK9S-M |
| Catalyst 6500 | CAT OS Hybrid | N/A | Not applicable |
| 2500 | IOS | N/A | Not applicable |
| 2600 | IOS | 12.2(15)T | 12.3 non T for bandwidth % |
| 2600XM | IOS | 12.3(13) | For bandwidth % option |
| 3640 | IOS | Any 12.2T | Or higher |
| 2800 | IOS | 12.3(11)T | What shipped with router |
| 3700 | IOS | 12.2(15)T10 | What shipped with router |
| 3800 | IOS | 12.3(11)T | What shipped with router |
| 7200 | IOS | Any 12.2T | Or higher |
| 7300 | IOS | Any 12.2T | Or higher |
| 7500 | IOS | Any 12.2T | Or higher |
| 10700 | IOS | 12.0(26)S2 | Latest is 12.0(30) |
| | | | |
| | | | |
| | | | |

**Note:** *be aware of any security type IOS versions that load SDM. QoS can work in conjunction with such IOS and features, but remember that the IOS will load up in a locked down state for certain features. Please verify if any IOS security features are enabled before deploying QoS to the device.*

# *5.0 NYC Utility's QoS Strategy*

To leverage its current data network infrastructure to strategically position NYC Utility's network to support converged applications today and tomorrow. The goal is to provide a familiar set of QoS mechanisms across the enterprise that will operate in a consistent manner to support a converged environment of Voice, Video and traditional data based applications. This is not easily accomplished with the use of many different platforms in between any two end points for the different platforms exhibit inconsistent hardware and software behaviors based on its features. To achieve QoS operating consistency from end-to-end all devices must support a common denominator hardware and software service relating to QoS.

## *5.1 NYC Utility's Current and Future QoS Needs*

NYC Utility currently has no formal QoS solution in place today. All applications, including limited deployments of voice are utilizing best effort class of service today. The network is configured for FIFO(First in and First Out) and WFQ(Weighted Fair Queuing), which is acceptable but with the addition of voice, fax and interactive video traffic these applications may not work properly together in the current network environment.

### *5.1.1 NYC Utility's Current QoS Needs*

NYC Utility needs to provide a basic method to ensure that Voice and Interactive Video traffic will always have the bandwidth needed between any two points in the NYC Utility network.

NYC Utility needs to provide "End-to-End" QoS across its enterprise network. End-to-End QoS is defined as ***"QoS should be available anywhere between two end users within the NYC Utility enterprise".*** This could be between two business offices for VoIP such as between building A and building B or provide voice contingency services for a business office such as building A or a disaster recovery building. Another end-to-end scenario could be to support VoIP from one business office to a substation. The initial need is for VoIP but Video conferencing is also another up and coming requirement for QoS.

### *5.1.2 NYC Utility's Future QoS Needs*

- **Scalable Voice service(additional calls crossing the enterprise without a drop in quality)**
- **Video - future video applications and enhancement to Meeting Place, Rich Media Content(RMC)**
- **Critical applications support for any application that needs it**
- **ERP type applications**
- **Control Center applications**
- **Custom applications**

## *5.2 General QoS Design Principals*

The way to begin a QoS deployment is to not by glossing over the offered QoS toolset and picking a la carte tools to deploy or just throwing Auto QoS out into the environment and "see" what happens. In other words, do not enable QoS features simply because they exist. Instead, start from a high level and clearly define the enterprise objectives first then outline how those objectives can be met with the QoS features available today. The previous sections have outlined the hardware and software platforms and the application class definitions with their respective and recommended markings that QoS features can apply to them. The previous sections also outlined the NYC Utility candidate applications for inclusion into NYC Utility QoS Model.

### *5.2.1 NYC Utility's QoS Design Considerations*

To be able to provide the Quality of Service levels for the applications and application types defined in the previous sections a QoS model must be created and applied against NYC Utility's application and network infrastructure. The model used could be a pre-defined model from a vendor such as Cisco or a custom one from NYC Utility. Regardless of where the model comes from NYC Utility should have design considerations outlined. These design considerations enable NYC Utility to create their solution based on meeting these considerations or select the proper pre-defined QoS model based on the amount of design considerations that the model meets.

## The current NYC Utility QoS Model Design considerations

### Overall Design considerations:

- **Must be able to troubleshoot and manage easily**
- **Expandable/scalable to other QoS technologies(like IntServ)**
- **Support for future applications**
- **Support for future protocol enhancement such as SCTP, SRTP, Q.SIG**
- **Easily migrated to IPv6**
- **Support of future QoS enhancements such as ECN**
- **Easy to implement, deploy and roll back**
- **Configuration consistency across enterprise platforms**
- **Provide basic DoS mitigation capabilities**
- **Must be manageable with existing network management systems**
- **Asymmetric routing paths consideration - End-to-End Quality of Service must work regardless of path taken between end points with in the NYC Utility network**
- **Network failure and wrapping of core – QoS should continue to work under any network outage condition.**
- **Need for backward compatibility with IP Precedence**
- **Standards based and not locked into any proprietary vendor solution**
- **Handling of valid traffic based on port and protocol, but illegal VoIP traffic like FWD and Skype are blocked**
- **No "jerkiness" on video conferencing sessions**
- **18-33% of any link's bandwidth reserved for Voice traffic**
- **25% of any links bandwidth is reserved for Best Effort class**
- **18-33% of WAN links traffic dedicated to voice**
- **15% of Wan links traffic dedicated for Video**
- **Must provide a basic Denial of Service mitigation switch**

### VoIP bearer and signaling specific design considerations:

- **G.711 is used for best quality. Enough bandwidth available for a defined number of calls per user**
- **No garbled or hiccupped voice**
- **Support ITU G.114 end-to-end delay of 150 milliseconds**
- **Support a packetization interval of 20 milliseconds**
- **Loss should not exceed 1%**
- **Jitter should meet a targeted budget of less than 30 milliseconds**
- **Guaranteed priority bandwidth of anywhere between 21 to 320 kbs per call(codec specific)**
- **Do not allows illegal(non Call Manager defined or approved) codec on the network**
- **150 bps, plus layer 2 overhead per phone of guaranteed bandwidth is required voice control traffic; more may be required depending on the call signaling protocol in use**
- **Allow at least 5 percent of traffic bandwidth for call signaling traffic**
- **Support for VoIP over wireless.**

## Cisco Call Manager CAC considerations

- **Must support Call manager 4.0.02a and the upcoming 5.0 release**
- **Must support  MCS subscribers**
- **Must support 1 MCS Publishers**
- **Must be able to scale above 300 users per Subscriber server per location**
- **Must support Cisco's IP communicator Soft-phone**
- **Must support SRST**
- **Must support POTS gateways**

Every department may have two Vlans, one data and one voice. Both Vlans will each have its own subnet and the voice Vlan in each department will have a different subnet. This also leads to having the subnet number documented for voice traffic. It is rumor that Cisco may do away with the Voice Vlan concept, if this is true than the following information regarding Voice Vlans is no longer relevant. If the Voice Vlan architecture is to be used in NYC Utility then see the recommendations below.

***It is recommended*** that NYC Utility outline(if not already done) a voice Vlan subnet plan for the following reasons.

1. **Easily manage and provision IP address for voice Cisco IP Phone users**
2. **Identifies by subnet number managed voice device**
3. **Identifies by subnet number where the call stream is originating or terminating by reviewing sniffer traces or management applications that highlight peer IP traffic conversation pairs.**

The main downside to deploying Voice Vlans is that the number of L3 subnets required for every data Vlan doubles. Since NYC Utility utilizes VLSM addressing and is using the private address space of 10.x.x.x/24 the number of subnets required is not a major issue. With a 24 bit subnet mask NYC Utility has 65536 subnets each accommodating 254 hosts minus the broadcast and local subnet addresses.

As the number of Voice Vlan subnets grows ***it is recommended*** that NYC Utility consider summarizing its EIGRP routing entries so lower end platforms speaking EIGRP are not deluged with subnet information about Voice Vlans. This can be accomplished with EIGRP summary commands and or advanced EIGRP features such as EIGRP STUB.

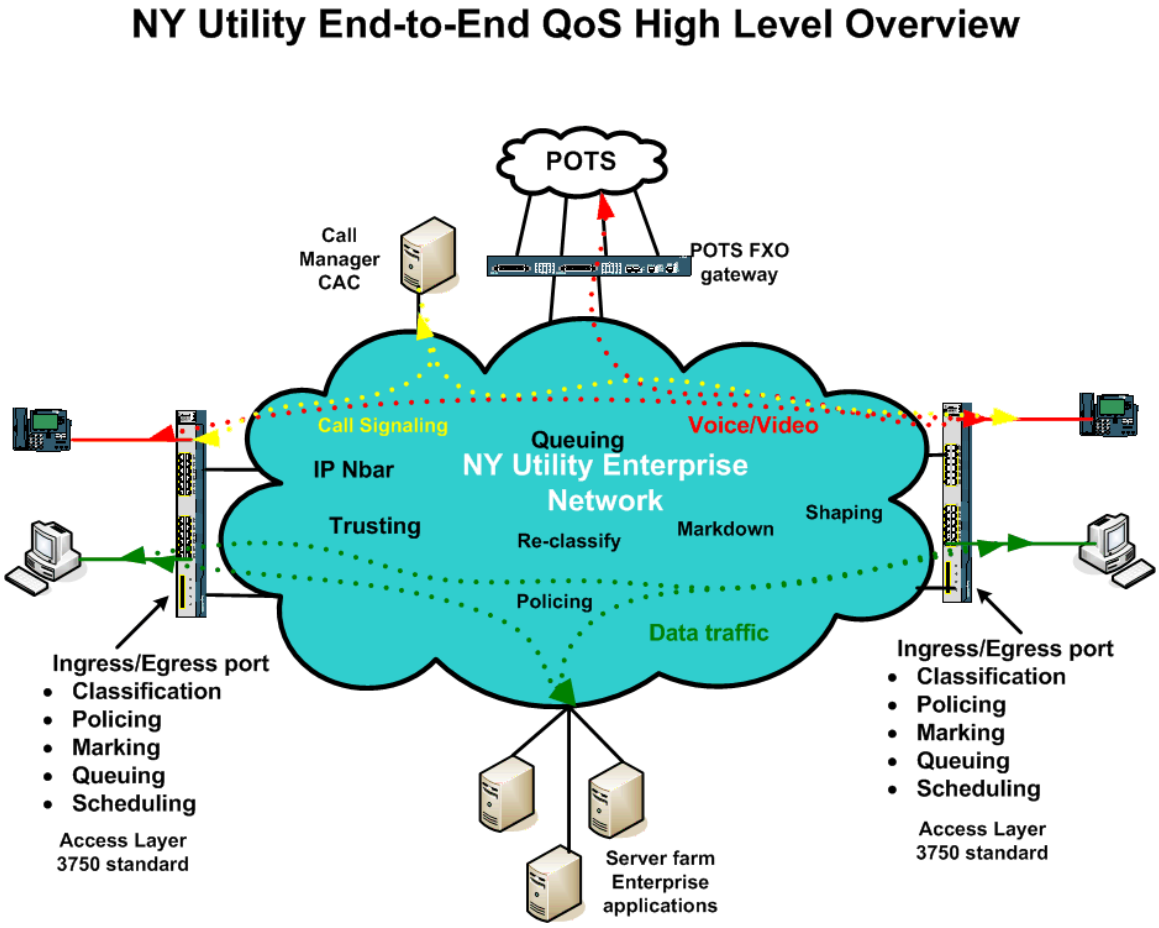
## *5.3 The Basic QoS Solution and Approach*

The basic approach to deploying QoS in NYC Utility's enterprise is to start small with a basic model that provides the foundation for a more granular model to handle additional classes of service. The basic foundation model will be applied against all infrastructure components. The basic concept of QoS in NYC Utility's context is to mark packets for different QoS levels at various edges of the network where user workstations, file servers, IP phones and other end point devices reside. These marked packets shall be "trusted" on QoS enabled devices across the enterprise. What trusting means is that the marked packets, as they cross a distribution or core layer router/switch, will be trusted(believed) for their markings (set at the access layer switch) and will not be changed to a default state. The router/switch shall then act on the markings of the packets for bandwidth allocation, queuing and transmit schedule forwarding to the next hop device. Each switch or router in between the communicating peers will invoke its configured PHB based on the packet's marking. PHBs were discussed at the beginning of this document in Section 1.2.2. The access layer switches comprise the "un-trusted domain", for packets need to be classified and marked, and any device between any access layer switch is considered part of the "trusted domain". End user packets shall originate at an un-trusted domain, cross a trusted domain for QoS, and arrive at another access-layer switch or server farm switch. The same applies for the return path packet.

The edges of the network, access layer switches shall be responsible for classifying traffic, marking traffic, policing traffic(if applicable), queuing and scheduling marked traffic for transit onto the network. All of the other infrastructure components(distribution and Core layer routers and switches) in between two end clients on the network will just trust the markings of such packets, act on its PHB and perform any queuing(if congestion is present) and scheduling to transmit the packet onto the next hop segment toward the other end access layer switch and corresponding end client. The response packets in the opposite direction shall exhibit the same behavior so packets are marked in both directions.

The core and distribution layer components will also have the capability to mark down or reclassify packets based on congestion level policies, police, and drop packets at different points in the enterprise. Denial of Service mitigation features are used to identify, markdown, police and drop illegal packets and will also be available to turn on and off as needed throughout strategic points in the network. These are additional features that can be applied at the basic level of NYC Utility's QoS solution or added later.

A conceptual diagram of this approach is depicted is on the following page.




## *5.3.1 General QoS Packet Marking and Handling Principals*

The following section provides important guidelines that NYC Utility should uphold when planning, deploying and maintaining its QoS models. These principals will be applied to the QoS models defined in this paper. They are listed here for reference and recommended best practices purposes as well.

### *5.3.2 Classification and Marking Principles*

When it comes to classifying and marking traffic, a general DiffServ design principle is to classify and mark applications as close to their sources as technically and administratively feasible. This approach promotes end-to-end DiffServ and per-hop behaviors (PHBs). Sometimes endpoints can be trusted to set CoS and DSCP markings correctly, but, in most cases, it is not a good idea to trust markings that users can set on their PCs (or other similar devices). This is because users easily could abuse provisioned QoS policies if permitted to mark their own traffic. For example, if DSCP EF receives priority services throughout the enterprise, a user easily could configure the PC to mark all traffic to DSCP EF right on the NIC, thus hijacking network-priority queues to service that user's non-real-time traffic. Such abuse easily could ruin the service quality of real-time applications (such as VoIP) throughout the enterprise. For this reason, the clause "as close as . . . administratively feasible" is included in the design principle. Also, this principal applies to rogue or faulty applications incorrectly setting their DSCP markings.

Following this rule, **it is further recommended** to use DSCP markings whenever possible because these are end-to-end, more granular, and more extensible than Layer 2 markings. It is also recommended to use DSCP marking for a consistent, standards based, set of packet class markings. This approach makes it easier to manage and scale the QoS environment in the future for NYC Utility is working off a consistent standards set of class markings.

### *5.3.3 Policing and Markdown Principles*

There is little sense in forwarding unwanted traffic only to police and drop it at a subsequent switch or router. This is especially the case when the unwanted traffic is the result of DoS or worm attacks. The overwhelming volumes of traffic that such attacks can create would drive network device processors to their maximum levels, causing network outages. Therefore, **it is recommended** to police traffic flows as close to their sources as possible. This principle applies to legitimate flows also because DoS and worm-generated traffic might be masquerading under legitimate, well-known TCP and UDP ports, causing extreme amounts of traffic on the network infrastructure. Such excesses should be monitored at the source and marked down appropriately. However, the use of aggregate policing at an uplink instead of access-layer ports may also be a more viable approach in NYC Utility’s case. One reason is to simplify the QoS configuration complexity by just having the aggregate policers on the uplink ports and they can be turned on and off when needed(if the platform supports aggregate policing). Another solution is to have DoS specific policers available to turn on and off at a moments notice on all router interfaces and switch uplink ports. These DoS features will be discussed in more detail shortly. The other reason is that NYC Utility has local servers on their access layer switches so policing at that level could affect inter-switch/stack traffic.

Whenever supported, markdown of traffic should be done according to standards-based rules, such as RFC 2597 ("Assured Forwarding PHB Group"). In other words, whenever supported, traffic marked to AFx1 should be marked down to AFx2 or AFx3. For example, in the case of a single-rate policer, excess traffic originally marked AF11 should be marked down to AF12. In the case of a dual-rate policer (as defined in RFC 2698), excess traffic originally marked AF11 should be marked down to AF12, and violating traffic should be marked down further to AF13. Following such markdowns, congestion-management policies, such as DSCP-based WRED, should be configured to drop AFx3 more aggressively than AFx2, which, in turn, is dropped more aggressively than AFx1. NYC Utility will not utilize marking down of its traffic in its Basic Voice QoS model but the QoS Models outlined for NYC Utility position it to use this feature in the future if required. Marking down traffic that is potential DoS traffic will also be available for NYC Utility to use. The principal here is that traffic identified as DoS traffic will be marked down to a lower class marking at different points in the network and will be treated to the smallest queues at each switch or router interface hope. This behavior in turn increases the marked down packet's chances of being dropped more often than other valid traffic and thus mitigating any affect on the limited amount of invalid traffic affecting any device.

### *5.3.4 Queuing and Dropping Principles*

Critical applications, such as VoIP, require service guarantees regardless of network conditions. The only way to provide service guarantees is to enable queuing at any node that has the potential for congestion regardless of how rarely, in fact, this might occur. This principle applies not only to campus-to-WAN or VPN edges, where speed mismatches are most pronounced, but also to campus interlayer links (where oversubscription ratios create the potential for congestion). **There is simply no other way to guarantee service levels than to enable queuing wherever a speed mismatch exists.** *Unless all the routers and switches in between supported an IntServ based reserved bandwidth on demand protocols.*

When provisioning queuing, some best-practice rules of thumb also apply. For example, the Best-Effort class is the default class for all data traffic. Only if an application has been selected for preferential or deferential treatment is it removed from the default class. NYC Utility has many data applications running over their networks, adequate bandwidth must be provisioned for this class as a whole to handle the sheer volume of applications that default to it. Therefore, **it is recommended** that at least 25 percent of a link's bandwidth be reserved for the default Best-Effort class. These percentages are also denoted on the NYC Utility QoS Model Platform Matrixes in ***Appendix G.***

### *5.3.5 DoS and Worm Mitigation Principles*

Whenever part of the organization's objectives is to mitigate Denial of Service(DoS) and worm attacks buy marking down packets to the Scavenger-Class the following best practices apply.

Profile the applications to determine what constitutes normal versus abnormal flows, within a 95 percent confidence interval. Thresholds differentiating normal and abnormal flows vary from enterprise to enterprise and from application to application. Caution must be extended not to over scrutinize traffic behavior because this could be time and resource exhaustive and easily could change from one day to the next. Remember, the presented Scavenger markdown strategy will not apply a penalty to legitimate traffic flows that exceed thresholds (aside from re-marking); only sustained, abnormal streams generated simultaneously by multiple hosts (highly indicative of DoS and worm attacks) are subject to aggressive dropping, and only after legitimate traffic has been serviced. NBAR can help in the classification and management by adding different illegal application “signatures” to the QoS model for DoS/Worm mitigation.

To contain such abnormal flows, **it is recommended** to deploy campus Access-Edge Policers to re-mark abnormal traffic to Scavenger (DSCP CS1). Additionally, whenever Catalyst 6500s with Supervisor 720s are deployed in the distribution layer, **it is recommended** to deploy a second line of policing defense, at the distribution layer via per-user microflow policing(if supported on the platform) Remember though that NYC Utility has servers on its access-layer switches so policing should be done either on a case by case switch basis or a limited policer is in affect so server traffic is not affected and if a Worm or DoS is present, some of its traffic will bleed out but not all or most of it.

To complement these re-marking policies, it is necessary to enforce end-to-end Scavenger-class queuing policies, where flows marked as Scavenger will receive a less-than best-effort service whenever congestion occurs. It is important to note that even when Scavenger-class QoS has been deployed end-to-end, this strategy only mitigates DoS and worm attacks and does not prevent them or remove them entirely.

### *5.3.6 A Further Word on DOS Mitigation*

Denial of Service(DoS) type of attacks have been around even before the modern day internet was available. The phone companies would have issues with call volumes overloading switches. One example was the Mothers Day overload of calls on the circuits. Whether malicious or not DoS can be present in different forms. For data networks there are several classes of DoS patterns that can be identified and mitigated. Some of these classes are outlined below.

1. A stream of packets that just uses up all bandwidth on a network link(s) thus denying any valid traffic access to the network. This type of attack pattern is analogous to the old broadcast storms of earlier networks.

2. A stream of packets that keep a device(router or an application server) busy at the application level. This type of attack, such as a SYN attack or higher up the OSI stack can cause a specific service on a server to be denied. The network is not overloaded and the server attacked is not overloaded but the specific service is "drowned" out.

3. A small stream of packets that cause a device(router or application server) to just overload its entire CPU, memory allocation, data bus or interrupt architecture and keep the entire device so busy that its function is akin to it being shut off. This type of attack is also deemed a CPU exhaustion attack.

The one thing that the above three types of DoS have in common is that a stream of packets is used, whether a large stream to use up link bandwidth or a small one to just kill a service, a stream or stream pattern is present. With advances in processing power on routers and switches the 3rd class of DoS listed is becoming more difficult to achieve. This is because the processors are so efficient that they can process even the bad traffic without utilizing too many CPU cycles. So, for the small stream of packets, such as an ARP attack, against newer devices the device will still function. Cisco also provides features such as ARP Throttling. ARP Throttling limits the rate at which packets destined to a connected network are forwarded to the route processor. Most of these packets are dropped, but a small number are sent to the router (rate limited). There are many other forms of DoS and distributed DoS (DDoS) not listed here but the same principal used to mitigate the basic forms of DoS listed earlier apply to those as well.

The QoS tools outlined in the NYC Utility QoS Toolset section later in this paper provide the tools to handle any of the three scenarios listed above. The goal is to markdown the offending traffic or to just police and drop either all of traffic or a classified type so that even a smaller stream of illegal packets cannot affect a device thus reducing the possibility of that device's service becoming busy or the CPU exhausted. This is done easily using QoS tools when just applying to aggregate traffic and not a specific type. All that has to be done on the device or its uplink or down link port is to turn on the QoS option to limit traffic and the offending traffic will be reduced and so will its attack prospects. This approach is easier to use for there is no need to constantly classify illegal traffic using Access Control Lists(ACLs) or other tools and then apply the policer. ACL management can become cumbersome to manage in pre and post DoS instances. The QoS tools for handling this will be deployed in strategic locations throughout NYC Utility enterprise to turn on and off when needed.

The following links provide additional information about such DoS issues and the QoS role in handling them.

http://www.cisco.com/en/US/about/ac123/ac114/ac173/ac253/about_cisco_packet_technology0900aecd800e0151.html

http://www.cisco.com/en/US/products/hw/switches/ps708/products_configuration_guide_chapter09186a008019c6e7.html

There is one final type of attack that QoS tools cannot mitigate. This type of attack can be considered the **"killer packet"** type. This type of attack does not require a stream of packets to exhaust a link, CPU or service. Though similar to a poison ARP or DNS packet attack it could comprise of just one packet sent to a device to utilize a very specific exploit. It could be an incrementing series of very specific exploit packets sent over days, weeks or months to cause a negative condition on a device. Such types of attacks are very difficult to characterize and monitor for. So, QoS tools do not offer much help where this type of attack is concerned. A good example of such an attack can be found in a report by Applied Methodologies at:
http://www.amilabs.com/ami-ciscoexploit.pdf

This type of attack requires the use of forensic protocol analysis, security intrusion detection/avoidance, firewall tools and the security support personnel's vigilance in tracking such exploits, to prevent them from occurring.

### Control Plane Policing

As of this writing Cisco does provide one additional DoS prevention tool for the **"killer packet(s)"** type of exploit. The tool is called **Control Plane Policing(CPP).** Control Plane policing is an internal IOS tool for routers to prevent packets from taking up control plane resources and exhausting the CPU. A router has several "planes" at different processing levels. One is the Management Plane(MP), another is the Data Plane(DP) and the third is the Control Plane(CP). The Data plane handles the packets arriving on the device's interface. The Management plane handles the management of the device, CLI, SNMP et. al., and the Control Plane handles the link management, routing protocols, signaling and path forwarding. All packets arrive on the same input interface regardless of which operational plane they are destined. Most DDoS attacks focus on the MP and the CP to interrupt the processing at these levels. Cisco now offers a feature to police the Control Plane to help ensure that packets do not overrun processes at this level. Control Plane Policing utilizes the same MQC commands that QoS does. However, instead of applying a MQC Policer to an interface it is applied to an internally defined Control Plane Virtual interface. What is advantageous with this feature is that the DoS policers already created for NYC Utility can apply to the Control Plane interface for an additional layer of protection or new ones for CPP can be created. CPP can then be turned on or off using the NYC Utility QoS Toolset commands just as any other DoS policer on any router it is supported. **Note:** *CPP was not tested in the QoS Lab due to time considerations.*

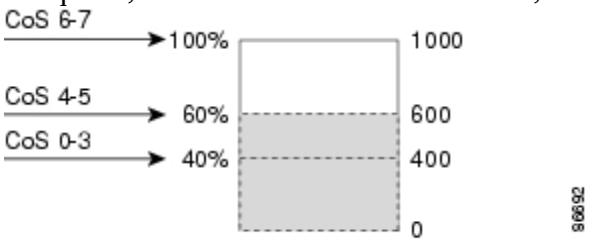
**It is recommended** that NYC Utility look into the use of CPP on all of its routers and switches. Since CPP is MQC based its configuration will have the same look and feel of other MQC commands and can be included into the NYC Utility QoS tool set.

## *5.3.7 Weighted Tail Drop*

Both the ingress and egress queues use an enhanced version of the tail-drop congestion-avoidance mechanism called weighted tail drop (WTD). WTD is implemented on queues to manage the queue lengths and to provide drop precedence for different traffic classifications. You will see this denoted on the NYC Utility QoS Base and Middle Model platform matrixes in Appendix G. under the column titled **"Thresholds Per Queue"** As a frame is sent to a particular queue, WTD uses the frame's assigned QoS label to subject it to different thresholds. If the threshold is exceeded for that QoS label (the space available in the destination queue is less than the size of the frame), the switch drops the frame. The following figure shows an example of WTD operating on a queue whose size is 1000 frames. Three drop percentages are configured: 40 percent (400 frames), 60 percent (600 frames), and 100 percent (1000 frames). These percentages mean that up to 400 frames can be queued at the 40-percent threshold, up to 600 frames at the 60-percent threshold, and up to 1000 frames at the 100-percent threshold. There will be different default allocations across different platforms.

In this example CoS values 6 and 7 have a greater importance than the other CoS values and they are assigned to the 100-percent drop threshold (queue-full state). CoS values 4 and 5 are assigned to the 60-percent threshold, and CoS values 0 to 3 are assigned to the 40-percent threshold.

Suppose the queue is already filled with 600 frames, and a new frame arrives. It contains CoS values 4 and 5 and is subjected to the 60-percent threshold. If this frame is added to the queue, the threshold will be exceeded, so the switch drops it.




The NYC Utility QoS models will employ these principals outlined earlier yet tuned to their goals and design considerations and the various model matrixes will show what features are in use. **It is recommended** that NYC Utility start out with the **default queue thresholds** for its lab testing and early deployments. Adjustments and tuning of the Queue thresholds can be performed in the future. It should be noted that adjusting and tuning Queue thresholds can be tedious across several different catalyst platforms so when the need does arise NYC Utility must carefully plan, test and lab their changes for all platforms affected first before making the changes on the production components. These tuned queue configurations will already be available for NYC Utility to turn on and off as part of its QoS Toolset. Ingress Queuing is not supported on most platforms(Only the 6500 currently supports it based on line cards used) and is rarely used so it will not be applicable to NYC Utilitys QoS needs.

### *5.3.8 SRR Shaping and Sharing*

Both the ingress and egress queues are serviced by Shaped Round Robin (SRR), which controls the rate at which packets are sent. On the ingress queues, SRR sends packets to the stack ring. On the egress queues, SRR sends packets to the egress port. You can configure SRR on egress queues for sharing or for shaping. However, for ingress queues, sharing is the default mode, and it is the only mode supported. In shaped mode, the egress queues are guaranteed a percentage of the bandwidth, and they are rate-limited to that amount. Shaped traffic does not use more than the allocated bandwidth even if the link is idle. Shaping provides a more even flow of traffic over time and reduces the peaks and valleys of *"bursty"* traffic. With shaping, the absolute value of each weight is used to compute the bandwidth available for the queues.

In Shared mode, the queues share the bandwidth among them according to the configured weights. The bandwidth is guaranteed at this level but not limited to it. For example, if a queue is empty and no longer requires a share of the link, the remaining queues can expand into the unused bandwidth and share it among them. With sharing, the ratio of the weights controls the frequency of de-queuing; the absolute values are meaningless.

The NYC Utility QoS model will employ the Sharing mode(where supported – at the 3750 access-layer) so queues can dynamically reallocate bandwidth amongst themselves when applicable. If more stringent guarantees of bandwidth are needed to be enforced in the future, Shaped Mode can be deployed.

Some platforms support different queuing structures than others. To ensure consistent PHBs, ***it is recommended*** to configure consistent queuing policies according to platform capabilities as much as possible to ensure that there is a priority queue for voice traffic from one end of the enterprise to the other.

For example, on a platform that supports only four queues with CoS-based admission (such as a Catalyst switch), a basic queuing policy could be as follows:

| Traffic Type | Bandwidth Allocation % |
|---|---|
| Real-Time | • 33 % |
| Critical Data | Remaining % |
| Best-Effort | 25 % |
| Scavenger/Bulk | < than 5 % |

The above percentages will be the default setting noted on the NYC Utility QoS Base Model platform matrixes outlined in ***Appendix G.***

However, on a platform that supports a full QoS Baseline queuing model such as routers or Catalyst 6500 switches, the queuing policies can be expanded in such a way that they provide consistent servicing to Real-Time, Best-Effort, and Scavenger class traffic. For example, on a platform that supports 11 queues with DSCP-based admission (such as a Cisco IOS router or some Catalyst 6500 line cards), an advanced queuing policy could be as follows:

| | |
|---|---|
| Voice | • 18 % |
| Interactive-Video | • 15 % |
| Internetwork Control | Remaining % |
| Call-Signaling | Remaining % |
| Mission-Critical Data | Remaining % |
| Transactional Data | Remaining % |
| Network-Management | Remaining % |
| Streaming-Video Control | Remaining % |
| Best-Effort | • 25 % |
| Bulk Data | 4 % |
| Scavenger | 1 % |

The above percentages can be applied to the router platforms and can also be ported over to the NYC Utility QoS Middle Model platform matrix.

In this manner, traffic will receive compatible queuing at each switch and or router, regardless of platform capabilities—which is the overall objective of the DiffServ per-hop behavior definitions and NYC Utility's design considerations.

**It is recommended** that the Catalyst line card queue setup should match that of the access-layer queuing model as best as possible for consistent traffic class to queue handling across the enterprise as mentioned earlier.

The access layer devices 3550, 3750 and the 4500 utilize the following minimum **1P3Q1T(1 priority queue, 3 regular queues and one threshold per queue)** port queuing structure. Keeping the queuing allocations across platforms is important to ensure that packets fall into the same queue across all routers and switches from end-to-end. This approach promotes consistent handling of traffic and understanding for troubleshooting and management of QoS operations across all platforms.

However, it is at the Catalyst 6500 that this practice becomes difficult to maintain. As per the Catalyst 6500 Line Card Port Queue Matrix outlined in ***Appendix F.*** shows that a majority of the current ports in place today do not follow the **1p3q1t** minimum. Most ports are either **1p2q2t** for Gbics or **2q2t** for the RJ-45 ports on the MSFC based platforms. This does not pose an issue as long as there is a priority queue available on all uplink ports, router ports and special server ports handling VoIP functions. Otherwise, all of the other ports that don't have a priority queue, such as the **RJ-45 2q2t** type, should only be used for basic servers that don't require a priority queue. These ports should also not be used for router uplinks as well. The goal here is to ensure that priority queue support is consistent and available end-to-end so priority traffic such as VoIP will always have a priority queue available for it to reduce jitter. VoIP traffic should not be serviced in a non-priority queue if it can be avoided as much as possible.

Whenever supported, **it is recommended** to enable WRED (preferably DSCP-based WRED)on all TCP flows. In this manner, WRED congestion avoidance will prevent TCP global synchronization and will increase overall throughput and link efficiency. Enabling WRED on UDP flows is optional. This is applicable to the router platforms.

# *6.0 NYC Utility's QoS Models*

The NYC Utility QoS models are outlined below. As you may see from reading the previous sections regarding Cisco's Baseline QoS model is that the NYC Utility model follows the Cisco model closely. The first model provides a basic entry point to QoS with a smaller number of classes defined than that of the Cisco Baseline Model.

The **DSCP AF21** range has been reserved in both NYC Utility models for further use in granular classifications or for future changes to the model or standards.

These application classes, marking and bandwidth allocations listed are not to be deemed **"in stone"** except for the Real-Time class with a PHB of EF. These allocations are considered de facto industry standard or best practices by Cisco. NYC Utility can change these allocations and markings if they wish to do so but must be very vigilant in documenting and keeping track of such changes. The NYC Utility QoS Toolset to be discussed shortly facilitates managing such changes.

Below is the general matrix for the base model of QoS for NYC Utility applications. There are more platform specific and detailed spreadsheets included in this paper that outlines what is below plus adds the queue and threshold information. Please refer to ***Appendix G.*** for the platform specific and detailed matrixes.

The NYC Utility QoS Basic VoIP Model is outlined below. This is the ground floor model of QoS for NYC Utility to implement. This model outlines the classification, marking and trusting of just VoIP related packets only. The way that this model is applied to NYC Utility's infrastructure is that all the access-layer switches will classify and mark voice and call signaling packets to the values according to the table below. Classifying and marking of the various VoIP CAC and media server ports will be the only "server" ports configured for QoS. All other server farm ports will not have QoS enabled, for it is not needed. The access layer switches and all core switches in between will just trust these markings and process the appropriately marked packets accordingly.

The VoIP packets marked EF(expedited forwarding) will use the priority queue at each switch or router hop across the enterprise until it reaches its destination. On the return trip, the same approach applies. Since the same model will be applied to the access layer switch at all ends between any two users the return voice packet will be marked and process accordingly on its return trip back. All the switches and routers in between will just look at the markings(trust them) and move them into the priority queue.

This behavior ensures that the voice traffic receives consistent end-to-end priority handling across the enterprise and across all platforms. All other non voice related traffic shall be treated as it is today, unclassified, unmarked and not using a priority queue. This basic model can be applied with the default queue and thresholds that are currently set on all switch and router interfaces.

Tuned queuing and thresholds, policing to limit voice traffic per user and DoS mitigation policing are some features that can be turned on and off when applicable for this model. For example NYC Utility can start out with its basic VoIP model with default queue allocations set across the enterprise and later turn on advance queuing/thresholds settings when needed to ensure call jitter budgets remain intact. This approach ensures a simpler deployment of QoS and helps NYC Utility determine if even any additional tuning is required.

**NYC Utility QoS Basic VoIP Model**

| Basic Voice Model<br><br>General QoS Classes | NY Utility Application Class | PHB | DSCP |
|---|---|---|---|
| Real-Time | Voice | EF | 46 |
| Call Signaling | Call Signaling | CS3 | 24 |
| All other traffic | Network control | CS0 | 0 |
| | Video all | CS0 | 0 |
| | Network Management | CS0 | 0 |
| | ConED General applications | CS0 | 0 |
| | Bulk data for FTP and backups | CS0 | 0 |
| | Scavenger | CS0 | 0 |
| | Best effort | CS0 | 0 |

The NYC Utility QoS Base Model below is used as a reference point for the various configurations on each platform to provide a consistent end-to-end marking, policing, queuing and scheduling solution for the candidate QoS applications listed earlier in Section 3.1. This is the next progression model that NYC Utility shall implement to scale its QoS solution.

**NYC Utility QoS BaseModel**

| Standard 5 Class Model<br><br>General QoS Classes | NY Utility Application Class | PHB | DSCP |
|---|---|---|---|
| Real-Time | Voice | EF | 46 |
| | Video all | AF41 | 34 |
| Call Signaling | Call Signaling | CS3 | 24 |
| Critical Data | Network control | CS6 | 48 |
| | Network Management | CS2 | 16 |
| | NYC Utility General applications | AF31 | 26 |
| | Bulk data for FTP and backups | AF11 | 10 |
| Scavenger | Scavenger | CS1 | 8 |
| Best Effort | Best effort | CS0 | 0 |

If NYC Utility needs to scale it QoS environment even further in the future then the NYC Utility QoS Middle Model, which is just a granular model, compared to the Base and Basic Voice models shown earlier is available to use. This model looks similar to the Cisco Baseline model and is presented here for future use. NYC Utility can scale from the Base model to this model in pieces, for example by just adding more specific markings in the Critical Data class or moving the Network Control items form the Base model's Critical Data class into its own class area as in the Middle model. This is dependent of course on a platform's ability to provide enough queues per class for additional granularity.

**NYC Utility QoS Middle Model**

| Standard 8 Class model | NY Utility Application Class | PHB | DSCP |
|---|---|---|---|
| General QoS Classes | | | |
| Real-Time | Voice | EF | 46 |
| Video | Video all | AF41 | 34 |
| Call-Signaling | Call Signaling | CS3 | 24 |
| Network Control | Network control | CS6 | 48 |
| | Network Management | CS2 | 16 |
| Mission-Critical Data | NYC Utility General applications | AF31 | 26 |
| Bulk Data | Bulk data for FTP and backups | AF11 | 10 |
| Scavenger | Scavenger | CS1 | 8 |
| Best Effort | Best effort | CS0 | 0 |

## *Cisco Catalyst Platform Specific NYC Utility Model Matrixes*

There are several pull out matrixes in ***Appendix G.*** that provide more detail about the NYC Utility models than listed here. These matrixes go a step further than the ones outlined earlier for they provide platform queuing and threshold information as well. These matrixes are a valuable source of reference and should be updated to reflect platform additions, deletion or updates. They are used for planning and provisioning QoS features now and in the future.

## *6.1 QoS End-to-End Architectures*

When one thinks of "End-to-End" one must understand that there may be many different end-to-end "scenarios" to be considered. In NYC Utility's case there are four.

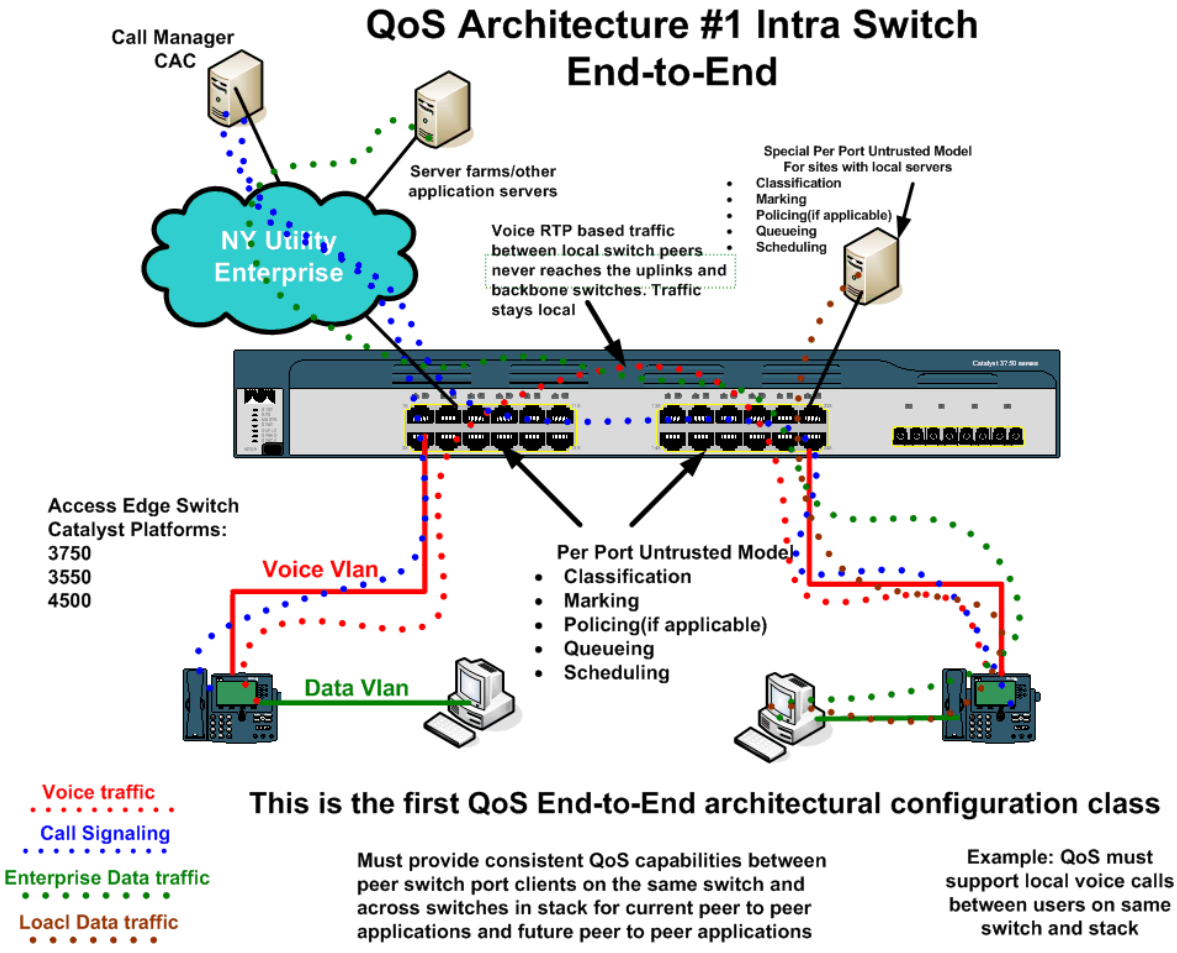
To facilitate these models across NYC Utility's enterprise requires different a "configuration" version of each model for each platform. This is because there are different Access layer platforms and different Core/Distribution platforms as well as different WAN platforms.

Smaller companies have one or two switching/routing platforms so most configuration options are consistent in contrast to NYC Utility. However, in most enterprises such as NYC Utility this is not the case. Nevertheless, NYC Utility has many different Access/Distribution/Core and WAN platforms that require a consistent operating QoS behavior for true anywhere end-to-end QoS. Cisco does facilitate this with its MQC discussed earlier in Section 4.1. Since there are several platforms to be considered a set of common denominator architectures that represent most of NYC Utility's enterprise end-to-end architecture have been defined for the purpose of QoS testing and planning. These are the different network architecture reference classes that NYC Utility's generic and specific cross platform QoS Toolset configurations shall be derived from.

The four different end-to-end architecture classes are outlined on the following pages.

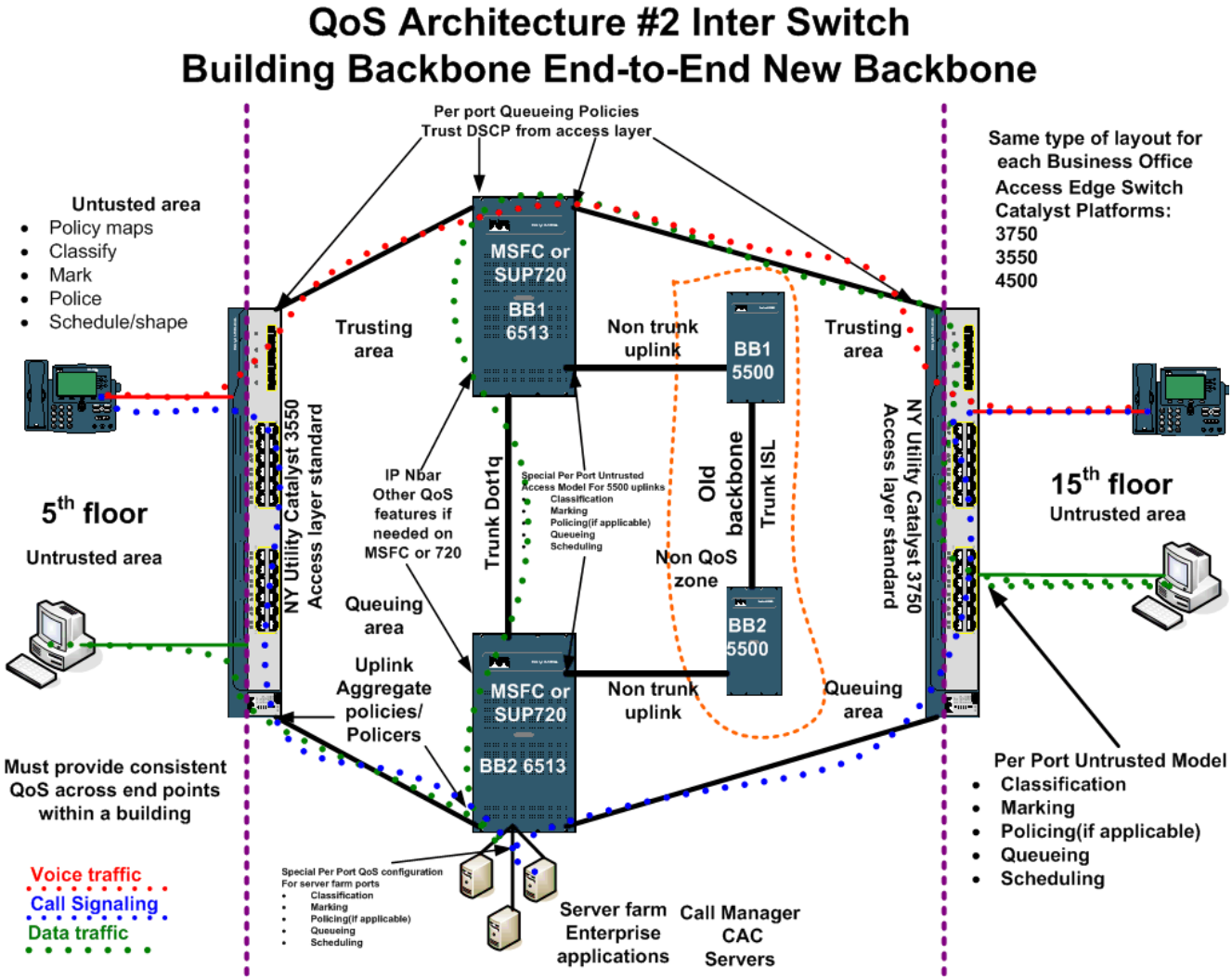
### *6.1.1 QoS Architecture #1 Intra Switch End-to-End*

This architecture diagram refers to providing QoS between data and voice applications between two clients on the same switch locally or same switch stack locally. The diagram provides additional information about QoS marking locations and trusting areas.



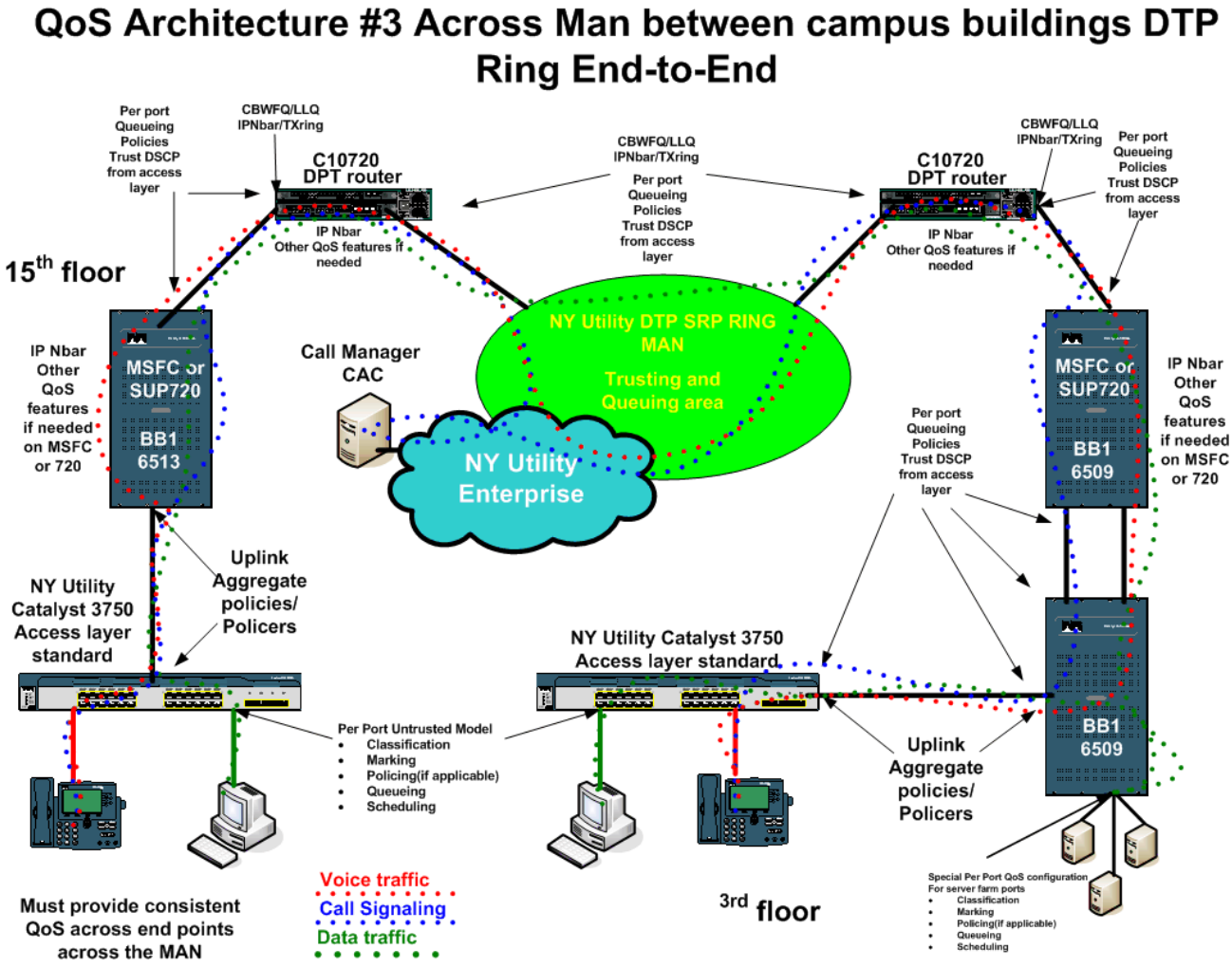
### *6.1.2 QoS Architecture #2 Inter Switch Building Backbone End-to-End*

This architecture diagram pertains to NYC Utility's new backbones in its headquarters and other major business office or any office where the access layer switches have uplinks to Catalyst 6500 switches. This diagram refers to providing end-to-end QoS between switches on different floors in a building across a building's switching backbone. The diagram provides additional information about QoS marking locations and trusting areas.



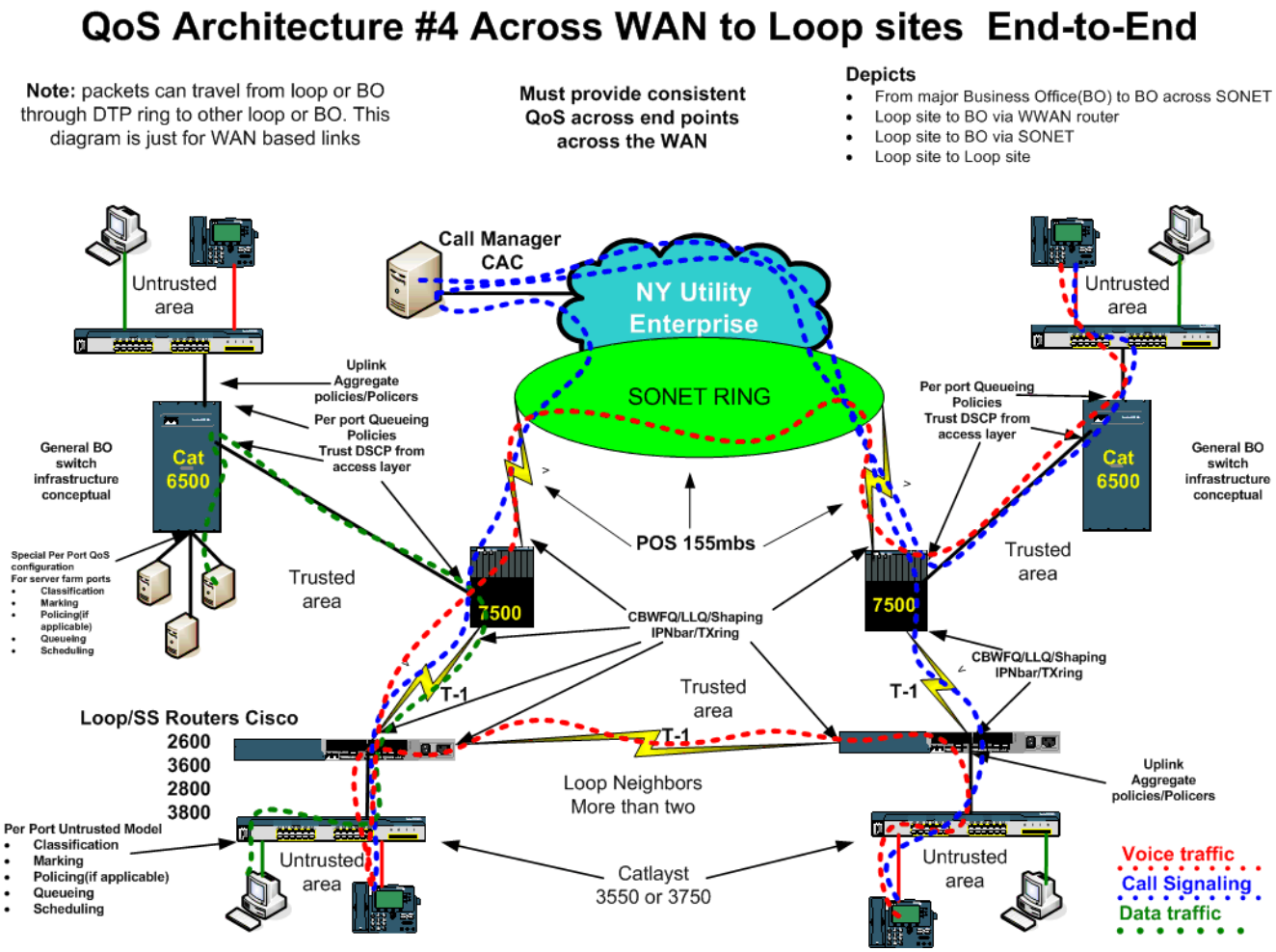
### *6.1.3 QoS Architecture #3 Across the MAN DPT RING End-to-End*

This architecture diagram pertains to NYC Utility's DWDM DPT high speed core backbone between its major Business Offices(BOs). This diagram refers to providing end-to-end QoS across the MAN to other BOs access layer clients. The diagram provides additional information about QoS marking locations and trusting areas.



### *6.1.4 QoS Architecture #4 Across the WAN To Loop Sites End-to-End*

This architecture diagram pertains to NYC Utility's WAN routers and to any site connected by a T-1, Multilink or lower than T-1 speed link such as loop sites. Also this diagram pertains to the loop sites connected via the SONET ring POS interfaces. This architecture references the end-to-end QoS needs of the loop sites connecting the loop sites via the WAN routers or though the SONET ring. The diagram provides additional information about QoS marking locations and trusting areas.




The four different End-to-End architecture considerations outlined earlier depict the level of complexity when planning for a true end-to-end QoS solution. As mentioned earlier in this paper the behavior of QoS must operate in a consistent manner regardless of the end-to-end architecture at an atomic level, per architecture, and across all architectures. This approach is needed to ensure that QoS operates and provides the service it was intended to provide between two client workstations regardless if they are on the same switch or in different boroughs/counties without encountering any QoS black holes(passing a non QoS device en route) To achieve this goal requires adherence to design considerations, best practices and rules.

## *6.2. NYC Utility QoS Design Rules:*

### *6.2.1 NYC Utility QoS Rule set#1*

1. All packets shall be marked with a DSCP value instead of CoS and IPP values unless otherwise required and already set by critical networking protocols such as EIGRP and HSRP. This approach ensures that NYC Utility knows what is marked what value and prevents any confusion in terms of CoS to DSCP translations.

2. The following table below is provided as a reference for configuration and supporting of QoS options. This table represents the initial DSCP values to IP precedence and COS values. This mapping provides a way for DSCP values to fall into the IPP levels when required for backward compatibility and for queue support. Most of the switches will have a unique mapping statement that will map the DSCP ranges to the IPP/COS values exactly as outlined below. The use of DSCP ranges for the DSCP to IPP/COS values will be utilized initially for NYC Utility but can be more specific in the future. The use of ranges does cause some confusion since a DSCP value assigned in a configuration may have been thought or planned to be for a particular class of service application that matches one IPP value but it is in a range that is mapped to another IPP value.

| DSCP Range numbers to IP Precedence or COS Map Reference | | |
|---|---|---|
| Range of DSCP values Decimal | Compatibable with these IP Precedence values | |
| 0-7 | 0 | Routine |
| 8-15 | 1 | Priority |
| 16-23 | 2 | Immediate |
| 24-31 | 3 | Flash |
| 32-39 | 4 | Flash Override |
| 40-47 | 5 | Critical |
| 48-55 | 6 | Internetwork Control |
| 56-63 | 7 | Network control |
| | | |

The design goal is to use DSCP values throughout NYC Utility for consistency but there are times when a platform will interpret the DSCP value as an IPP and COS value and act on its PHB accordingly, hence the need for such a table and mappings. For example the Spatial Reuse Protocol(SRP) protocol in the Core DPT/DWDM backbone will utilize the IPP markings so the IPP markings of packets need to reflect the proper DSCP values.

### *6.2.2 NYC Utility QoS Rule set #2*

1. Best Effort traffic is allocated 25% of a link's total bandwidth as per Cisco QoS Base Line best practices recommendations. For NYC Utility, 25% of 10mbs will be 2.5mbs and 25% of 100mbs will be 25mbs. The bandwidth allocations(depending on platform) may be specified in bit rates so NYC Utility must be cognizant of this when provisioning 100mbs links from 10mbs links.

2. Voice traffic shall be allocated 18% of the bandwidth on all non wan links.

3. Voice traffic shall be allocated 33% of the bandwidth on all T-1 and lower speed WAN links. This was already outlined in the previous sections but is listed here again for configuration purposes.

### *6.2.3 Classification and Marking Recommendations and Rules*

- **All access switches will implement a set of classification and marking configuration commands**

- **All user ports will implement classification and marking commands for Voice and Data traffic. This is the case even if an IP phone is present and marks the packets itself, for having the marking executed at the switch ports ensures that if a phone is moved, upgraded, changed to a non Cisco model or experiences operating issues in terms of marking packets the voice packets will still get marked regardless of phone related issues.**

- **User ports shall trust only Cisco IP phone devices since these phones already mark voice and signaling packets. No other trusting is done on user ports.**

- **All server access switches will implement only application based classification and markings(no need for extra voice QoS configuration commands on server only switches or local server ports on user access switches). This is needed to maintain the return trip marking consistency from the client.(Base and Middle Model implementations).**

- **All MSCF, Supervisor 720s and Core/Wan routers will implement NBAR for advanced classification needs when applicable for routed interfaces**

- **Additional Core layer router and switch classification and marking will only be utilized for policing and DOS uses**

- **Special servers and gateway devices will have their ingress ports configured for classification and marking of their outgoing enterprise facing traffic to ensure round trip consistency from user workstations.(Base and Middle Model implementations)**

### *6.2.4 Policing Recommendations and Rules*

- **Policing will NOT initially be performed on user ports to prevent inadvertent application performance issues. Policing of Realtime, Scavanger and Best effort traffic can be applied in the future when applicable.**

- **Policing will NOT initially be performed on any server ports to prevent configuration mistakes, or application port number changes from upgrades from causing valid traffic to suddenly get policed for markdown or dropping.**

- **Policing can be performed at the uplink port as an aggregator and extra line of defense for Worm mitigation. (See Section 7.0 for more details)**

- **Policing can be applied to control voice call bandwidth admission per user if a CAC based bandwidth control mechanism is not available**

### *6.2.5 Queuing Recommendations and Rules*

- **Queuing and queue allocations per DSCP value must remain as consistent across platforms as much as possible**
- **All Core and WAN routers should employ CBWFQ/LLQ capabilities**
- **A priority queue must be available on every uplink, router, Voice user and Voice server port**
- **All non Voice server switch ports do not need to have a priority queue allocated**

### *6.2.6 Trusting Recommendations and Rules*

- **All switch uplink ports will be set to trust DSCP values**
- **All router interfaces LAN and WAN will be set to trust DSCP values**
- **All switch to switch and router to router interfaces will be set to trust DSCP values**

### *6.2.7 Access Layer Switch QoS Required Policy Rules*

- **Appropriate (endpoint-dependant) trust polices can be used**
- **Classification and marking policies will be used**
- **Policing and markdown policies may be used when applicable**
- **Queuing policies will be used**
- **Uplink DOS mitigation policy will be available**

### *6.2.8 Distribution and Core Switch QoS Required Policy Rules*

- **DSCP-trust policies must be used**
- **Queuing policies must be used**
- **General Policing and markdown policies may be used when applicable**
- **DOS mitigation policy will be available**

### *6.2.9 Core and Wan Router QoS Required Policy Rules*

- **DSCP trust policies must be used**
- **Queuing Policies must be used**
- **General Policing and markdown policies may be used when applicable**
- **DOS mitigation policy will be available**

### *6.2.10 Additional Design Considerations*

Some additional design considerations and questions that NYC Utility must answer before scaling from its initial QoS model and deployment to a larger scale deployment:

1. For critical two or three tier Client/Server, CRM, ERP or custom applications NYC Utility must take into consideration that end-to-end QoS may not be complete between Application servers(proxy/content or front end types) and back end database servers such as Oracle or SQL server. The QoS markings are valid between the application server and the client. However from the application server to the database server residing on the same switch or in another location those types of transactions and traffic patterns must be considered before scaling QoS from its initial deployment.

2. Should HTTP based Intranet traffic fall under the Critical Application class or be thrown in as Best Effort? Should special HTTP based Intranet applications be given a higher class of service?

3. DNS and DHCP could be considered for Control Plane Network control markings to ensure in times of heavy congestion or congestion from an outage that these services are working.

4. Voice Vlan consideration should Voice traffic present on the Data Vlan be policed and dropped?

5. Voice Vlan consideration should data traffic found present on the Voice Vlan be policed and dropped?

6. Different Catalyst PFC and MSFC set L2 and L3 packet sizes inconsistently. ***It is recommended that*** NYC Utility should investigate this and ensure that all PFC and MSFC are consistent in hardware and software revisions.

7. The use of cRTP can be applied to heavily utilized T-1 links or links lower than T-1 speeds if applicable.

**8.** QoS across VPNs and firewalls require special considerations. There are methods to mark both the inner tunneled packet and the outer transport packet. NYC Utility should carefully review this method and test in a lab environment before implementing. If Voice is required to go over a VPN and or through a firewall then a separate "unique" QoS configuration set will be required on the VPN client edge and at the VPN's enterprise edge interfaces respectively. Also, the inward and outward firewall interfaces may need to reclassify and remark the packets as they enter or leave the NYC Utility enterprise. A complete set of MQC configurations are required to support QoS across VPNs.

**9.** QoS marking points for special services servers. If NYC Utility implements more than just the Basic Voice QoS toolset then special consideration must me made to other end devices that have to apply QoS marking principals. If NYC Utility implement the Base QoS model at the access layer for example then QoS marking configurations must be applied at the requisite end server or gateway to ensure that the return packets are marked and handled accordingly. This entails stripped down simpler marking configurations for each specific server. For example, if HTTP traffic is to be marked at the access-layer as best effort then the GOLRs must tag the internet traffic coming back into the NYC Utility network as best effort as well. A list of special server switch port and configurations names is outlined in Section 7.1.6 to facilitate in the planning of these serves and inclusion into the NYC Utility QoS Toolset.

**10.** Legacy switch uplinks - There are Catalyst 5500 series switches with user and possibly server populations residing off the 6500 based backbone in the Core sites. These switches do not have support for MQC and are thus QoS incapable. However, there are users that access the same set of services that other users do off the QoS capable access-layer switches. But, since these users are on 5500 series switches they have no way to classify and mark their packets. The only solution to this issue, other than upgrading the users from 5500 to a 3750 or 6500 series, is to provide a special uplink configuration on the 6500 switch that these 5500s connect to. This special configuration will classify and mark the packets from the 5500 ingress point into the 6500 domain from that point on. So, a packet from a 5500 based user will be marked at its 6500 uplink point and will receive QoS between the peer enterprise service and back to the uplink point. Packets between users within the same Vlan on the 5500 will not receive any QoS. A special QoS tool set command to turn on and off QoS for 55k uplink ports has been created and tested in the lab and is available as part of NYC Utilitys QoS Toolset.

**11.** Catalyst 2900 or older switch uplinks into Catalyst 6500 switches will have its MQC edge boundary at the Catalyst 6500 uplink interface. This ensures that applications are classified, marked, policed and scheduled in a consistent manner at the uplink point on the 6500 from the 2900s. Since the 2900 switches do not support MQC and are considered legacy the users located on the 2900s will not

have QoS within their switch cluster. Thus Catalyst 2900s to 2900s users will have no QoS between them.

**12.** Old NYC Utility Backbone and Etherchannel considerations: Since the 5500s do not support a consistent queuing and QoS configuration model as the approved QoS devices do, no QoS marking, queuing or policing policies will be applied to these backbone switches on end user ports nor on Etherchannels between Old Backbone switches.

**13.** DPT ring access or provide a T-3 to replace Multilink PPP. This approach ensures consistency with MQC configurations and functionality across DPT, T-3 and T-1 sites. The multilink PPP has different properties like LFI(link fragmentation and interleaving) that could help add to QoS functionality or possibly degrade it. It also presents NYC Utility with another "unique" QoS configuration set that has to be considered and managed. However, if the Multilink PPP links must still be supported than QoS features can be applied to the respective site interfaces. ***Note:*** *PPP Multilink/LFI configurations have not been created or tested and are not included in the NYC Utility tool set.*

**14.** There is no need to use QoS tools for broadcast based Virus/Trojans or worm traffic suppression. Instead, **it is recommended** to just use the switch's broadcast suppression features, thus improving performance and simplifying the QoS configuration. The switch ASIC will handle the broadcast suppression as apposed to using additional QoS tools.

**15.** IP based switch ports facing Fiber Channel Server connections should not have any QoS applied if it is not supported. Traffic from Fiber Channel Server ports can be classified and marked at the next uplink point when applicable.

**16.** 7200 Token/Ring Escon interfaces will not have any MQC configurations applied to their interfaces. Any IP based packets will be sent from these router interfaces will be trusted to the default class of service and no change is necessary nor an additional "unique" MQC configuration be required. However, until Token-Ring is completely phased out and there is an issue that requires IP based traffic originating from the remaining Token-Rings to have a class of service applied to them, then MQC configuration commands can be applied to the Token-Ring interface(if possible) to just classify and mark specific IP packets. The routers supporting such Token-Ring interfaces must be able to support CEF and MQC at a minimum. If not then it is up to NYC Utility to consider upgrading the router or remove the Token-Ring segment. **It is recommended** that Ethernet based OSA adapters be tested to see if QoS configuration commands can be applied to them.

**17.** All QoS configurations should be as consistent as possible amongst like platform ports. The use of different QoS queue and priority allocations for heavily utilized links should be a last resort. Such links should be investigated as to why the utilization is high and what traffic is causing such high utilization. Using QoS tools to **"gain"** bandwidth back or to **"wedge"** in traffic classes amongst such high utilization can cause additional problems when the utilization subsides and also increase the management and troubleshooting aspect of QoS by having yet another "unique" QoS configuration in the middle of an end-to-end path.

**18.** Aggregate Policers can be employed on uplink switch ports to police the entire Vlan(if supported), a specified amount of bandwidth or just a specific class of traffic. These Policers can be created and not applied until a specific need arises such as the condition when a worm creates more traffic or gets around the policies defined at the edge client device edge ports. Aggregate Policers should not be deployed initially for they may inhibit the natural flow of traffic from the uplink ports and may skew the behavior of the network's operation in regards to traffic flows.

**19.** Local non Server Farm servers pose a special concern. These are the file and print or any other type of local floor/department or field office server that resides on the same access layer edge switch as the clients. The presence of these servers requires another set of QoS configuration commands to support on the same edge access layer switch to ensure that those server ports mark their packets in a consistent manner, not only for intra switch/stack transport but for any file/print sharing or application access to such servers from other remote locations.

**20.** Pix 7.0 supports MQC and QoS - NYC Utility should consider looking into these features when applicable if QoS is to be applied to traffic crossing firewalls. Since version 7.0 of the PIX IOS supports MQC, many of the NYC Utility QoS Toolset MQC configuration files may be compatible. With PIX version 7.0 supporting MQC provides NYC Utility additional options with QoS in regards to firewall traversing and the benefit of a consistent CLI.

**It is recommended** that NBAR should be configured on all MSFC and Supervisor 720s running in Native mode and all routers in the Core and WAN.

NBAR provides the following features that would be advantageous to utilize

- **Advanced application and protocol identification and classification**
- **Simpler to configuration for Class maps, eliminates the need for access lists**
- **Provides application statistics in discovery mode**
- **Provides advanced and stateful DoS identification features**
- **Provides a plug in method of adding different protocol and virus/worm signatures without requiring an IOS upgrade**

NBAR operates in the IP Cisco Express Forwarding switching path, only the first packet within a flow requires stateful packet inspection, and the policy is applied to all packets belonging to the flow. NBAR stateful packet inspection requires more CPU processing power than simple access lists. However, on newer router platforms the overhead of enabling NBAR is quite minimal(typically 2 to 5%, depending on the traffic mix. **It is recommended** that NBAR be enabled on all Catalyst switch platforms running IOS and support it as well as all router platforms that can support it. NBAR can be used for worm mitigation through easy classification and can also be used as a protocol and traffic discovery tool on all links.

The QoS configurations for SRST routers should match that of the enterprise for these reasons:

- **Provide consistent operating and management behavior**
- **Ensure that return trip packets sourced from the SRST routers are marked appropriately**

# *7.0 Custom NYC Utility QoS Solution Built On the NYC Utility Models*

The previous sections of this paper outlined NYC Utility's recommended QoS models. This section covers how to implement the models within the NYC Utility enterprise.

## *7.1 Introducing the NYC Utility QoS Toolset*

The NYC Utility QoS Toolset or QoS Toolset is a set of basic to advanced platform based pre configured MQC QoS configuration command files that can be deployed and executed at any time. By utilizing the NYC Utility QoS Toolset in conjunction to following the recommendations listed in previous sections NYC Utility can meet the design considerations listed in section 5.2.1.

One can think of the NYC Utility QoS tool set as a set of macros similar to Cisco's Auto QoS macros, but tuned to NYC Utility's needs and more flexible in terms of changing and tuning specific QoS features.

The contrast to Auto QoS and NYC Utility's QoS Toolset is that with Auto QoS you execute one command and a whole series of macros are executed to provision a switch for QoS, regardless if the items provision are needed or tuned for the environment. If a QoS feature is not desirable after Auto QoS is executed one must search the configuration to disable the specific feature. Removing the QoS configurations items resulting from executing Auto QoS can be cumbersome and timely as well.

However, with the NYC Utility QoS Toolset one can just turn on a basic QoS function to start out and build on it by turning on and off various QoS functions as needed. For example, a switch might just need QoS enabled and trusting turned on for all of its ports. However, using the NYC Utility QoS tool set all that has to be executed is a ***truston*** command. If trusting were to be turned off a command ***trustoff*** can be executed thus removing the commands from all the interfaces.

This type of macro file approach provides NYC Utility the flexibility, to turn, test and **"feel"** various QoS knobs quickly without constantly oscillating between command line Configuration and Exec modes for changes. In deployments scenarios it enables NYC Utility to deploy specific QoS functions quickly and roll them back if the desired results are not achieved. This approach ensures that network devices do not end up with QoS **"spaghetti"** code that is unique per device which can become confusing and difficult to manage

### *7.1.1 NYC Utility QoS Toolset Advantages*

The NYC Utility QoS tool set provides many features and achieves the design consideration goals outlined earlier in this section. The QoS tool set provides the following benefits

- **Consistent look and feel of QoS implementation commands across all platforms**
- **Reflect the NYC Utility models in configuration and tuning options**
- **Plug and play capability for changing QoS options**
- **Consistency provides easier management and troubleshooting**
- **Follows Cisco recommended QoS practices support for MQC**
- **Enables NYC Utility to manage, tune and upgrade specific platform QoS classification markings, policing and queuing configuration commands easily by turning on and off features when needed**
- **Manageable through CiscoWorks QoS Policy Manager plug-in**
- **QoS configuration files can be centrally managed for changes and easy deployment**
- **QoS configuration files provide self documentation comments so, even if not used or about to be used, these files can be read on the device for tips and instructions**
- **Scales with QoS needs**
- **Easy to deploy and roll back**

The following section outlines an example of the QoS Toolset's contents and a basic implementation. A separate, formal, NYC Utility QoS Toolset User Guide for deployment and reference will be forthcoming to reduce the size of this paper.

### *7.1.2 NYC Utility QoS Toolset 3750 Example*

The NYC Utility QoS Toolset is comprised of a set of files with MQC commands in them. These files turn on and off various QoS features ("knobs") and for the most part are consistent across most platforms. These files will have the same names across platforms but some of the platform specific MQC commands within such files will be different. These differences are noted in comments in the files. The files use a nomenclature of the QoS features and the platform's model number. Some files are generic enough to just have its QoS feature in the name but no model number. Here are some examples using a 3750 platform and a 6500 platform.

**3750 access-layer platform Basic Voice model files:**

- **BASICVOICEQOS3750.cfg**
- **removeqos3750.cfg**

The two files above are used to turn on and remove QoS for this platform for just basic VoIP service. BASICVOICEQOS3750 just enables QoS on the switch, loads up the policy to mark just VoIP related traffic and to leave all other traffic at its default(unmarked). This file will also create the access-lists and setup the priority queues per port.

**All other queuing and queue thresholds are set at their defaults when QoS is enabled.** Another file for example can be used to implement the advanced queuing or threshold commands. The configuration command content of the file is in outlined in BLUE and comments are in GREEN.

```
!BASIC BARE BONES VOIP ACCESS LAYER QOS CONFIG 3750
!BUILT IN AMILABS SUNDAY MARCH 13 2005

!Using default QUEUE setup
!Priority Queue is Queue 1

Maps the COS to the appropriate DSCP values

mls qos map cos-dscp 0 8 16 24 32 46 48 56
mls qos map ip-prec-dscp 0 8 16 24 32 46 48 56
mls qos

!mls qos map policed-dscp 0 24 46 to 8
!excessive voice or general traffic can be remarked to scavenger

!AGGREATE POLICER FOR SWITH UPLINKS TO MITIGATE DOS TRAFFIC LEVELS
! Calculate aggregate based on total switch user ports at 10mb then
divide by half
! 24 user port aggregate example used below
mls qos aggregate-policer UPLINKPOLICER24 125400000 1000000 exceed-
action drop


This is the basic class map to match Voice traffic to a class for
marking

class-map match-all VOICEVLAN
  description ***VOICE RTP PACKET CLASS***
  match access-group name VOICE-VLAN  ACL listed at end of config

class-map match-all VOICESIGNALING
  description ***VOICE H323 SIGNALING PACKETS***
  match access-group name VOICE-SIGNALING

This is the policy map to set the match packets to the corresponding
DSCP value

policy-map NYCUTILITYBASICVOICE3750
  class VOICEVLAN
    set ip dscp 46      sets voice traffic to EF(expedited forwarding)
                        packet marked with value in uplink switches
                        will move this packet to the priority queue
  class VOICESIGNALING
    set ip dscp 24
  class class-default
    set ip dscp 0
```

```
!Policy map for DOS policer
policy-map POLICEDOS
  class class-default
    police aggregate UPLINKPOLICER24

Apply the policy to the end user interfaces and apply other QoS and
Voice Vlan commands if applicable

interface FastEthernet1/0/1
 description TEST VOIP QOS station
 switchport mode access
 switchport access vlan 10
 !switchport voice vlan 20  This is used if a voice vlan is utilized
 no ip address
 duplex half
 speed 10
 priority-queue out  Turns on the priority queue for EF marked packets
 service-policy input NYCUTILITYBASICVOICE3750  see the nomenclature?
On a 3550 switch for example, the policy map will be NYC
UTILITYBASICVOICE3550.


!mls qos trust device cisco-phone
Used if voice vlan and phones are trusted

 no mdix auto
 spanning-tree portfast

The "Interface Range" command can be used to apply these settings and
will be different from switch or stack configuration. You can just set
a range of ports to have QoS applied for users with IP phones and leave
other ports non QoS configured. Just edit the file to reflect your
switch port configuration and upload to the switch.

Remember when the file is uploaded to the switch QoS is not enabled or
executed. The related QoS Toolset command will be used to enable or
disable these MQC features. Also, if a majority of the switch port
profiles are similar than the configurations are even easier to setup
and manage.

!set trusting on uplink ports or any router ports connected to switch
!switches 1 and 4 per stack as per NYC Utility standard
interface range gigabitEthernet 1/0/1 - 2
mls qos trust dscp
priority-queue out

!may need to specify voice vlans subnet address in ACL
```

```
This is the access list used to identify the VoIP related traffic on
the 3750.

ip access-list extended VOICE-VLAN
 remark ***VOICE RTP PACKETS IP PHONE SETS DSCP***
 permit udp any any range 16384 32767
ip access-list extended VOICE-SIGNALING
 remark ***VOICE CALL SIGNALING PROTOCOLS***
 permit tcp any any range 1000 4000  NYC Utility specific Signaling
protocol    port range goes here
```

The NYC Utility QoS Toolset MQC configuration files utilize Named Extended ACLs to facilitate the consistent look and feel of the configurations and self documentation of the QoS Toolset configuration options.

That is all there is to setting up QoS at the access-layer for the Basic Voice model. This file can be loaded on the switch and a command can be given to load this configuration onto each switch's running configuration during deployment cycles. Another file to remove these configuration changes and a command to execute the file to remove commands will also be available…

The removal of QoS configuration file is outlined below. This configuration will remove all aspects of QoS on a switch, even disabling QoS if NYC Utility needs to. NYC Utility can leave QoS enabled but remove all the commands and turn on and off what it needs and when.

```
!removal of qos commands
!Using default QUEUE setup
!Priority Queue is Queue 1


Can use interface range for all interfaces in switch
 interface FastEthernet1/0/1
 no switchport voice vlan 20
 no priority-queue out    Turns off the priority queue
                          Since other queues are at their defaults
                          no other changes are necessary
 no service-policy history
 no service-policy input NYCUTILITYBASICVOICE3750 Removes any of the
models in use
 no service-policy input NYCUTILITYBASEQOS3750
 no service-policy input NYCUTILITYADVQOS3750
 no mls qos trust device cisco-phone
```

```
interface range gigabitEthernet 1/0/1 - 2
no mls qos trust dscp    Turns off trusting on the uplink interface
no priority-queue out    Turns off the priority quuee

no mls qos map cos-dscp 0 8 16 24 32 46 48 56  Removes QoS DSCP
                                               mappings
no mls qos map ip-prec-dscp 0 8 16 24 32 46 48 56
no mls qos  Disables QoS globally. Optional. Can leave it on and just
remove the configuration options or disable QoS completely on the
switch.

no policy-map NYCUTILITYADVQOS3750        Removes all policy maps
regardless which model is used
no policy-map NYCUTILITYBASICVOICE3750
no policy-map NYCUTILITYBASEQOS3750
no policy-map POLICEDOS             Removes advanced features as well

Remove all models QoS definitions
no class-map match-all VOICEVLAN
no class-map match-all VOICESIGNALING
no ip access-list extended VOICE-VLAN
no ip access-list extended VOICE-SIGNALING
no class-map match-all BESTEFFORT
no class-map match-all REALTIME
no class-map match-any CRITICALDATA
no class-map match-all CALLSIGNALING
no class-map match-all SCAVENGER
no class-map match-all BULKDATA
no class-map match-all VIDEO
no class-map match-all NETWORKCONTROL
no ip access-list extended BEST-EFFORT
no ip access-list extended CALL-SIGNALING
no ip access-list extended GENERAL-APPS
no ip access-list extended REAL-TIME
no ip access-list extended SCAVENGER-TRAFFIC
no ip access-list extended BULK-DATA
no ip access-list extended NETWORK-CONTROL
no ip access-list extended VIDEO-TRAFFIC
no mls qos map policed-dscp 0 24 46 to 8

no mls qos aggregate-policer UPLINKPOLICER24 125400000 1000000 exceed-
action drop   Remove 3750 platform uplink policer
```

Once the above two files have been configured for their respective platform dynamics in terms of number of ports and bandwidth allocated for the uplink policer they can be copied to the device via TFTP or using CiscoWorks for mass deployment.

When all devices have their appropriate QoS files loaded and deployment is ready to begin all that has to be done is execute the appropriate NYC Utility QoS Toolset command to enable or disable a QoS feature. This can be done one device at a time via the CLI, or through QPM or CiscoWorks Net Job.

In the example given here to just enable the 3750's Basic Voice QoS model the command to be entered at the switch would just be ***"basicvoice".*** Afterwards, if QoS needs to be rolled back all that would be entered at the switch would be ***"remqos"*** to remove QoS and return the switch back to its pre QoS state. With this approach if QoS needs to be removed due to another issue it can than be turned back on at a later date.

Let's take another example of what would be on the Core switches in between our 3750s in a single building for example…

Once again a set of files for this platform will be available. In this case for the 6500's with MSFC running in native the files needed for just the Basic Voice Model would be:

Loads the Basic Voice model on the 6500 switch

- **BASICVOICEQOSMSFC.cfg**
- **removeqosMSFC.cfg**

Adding and removing interface trust MQC command files

- **notrustdscpinterface.cfg**
- **trustdscpinterface.cfg**

### *7.1.3 NYC Utility QoS Toolset 6500 Example*

Below is the Basic Voice model QoS configuration for a 6500 switch running the Supervisor II/MSFC in Native mode. Many of the commands are present for future use of routed interfaces so the only ones of importance are the commands to just enable QoS on the device and map out the CoS to DSCP values. All of the Policy and Class maps are defined for future routed interface and NBAR use but will not be used initially.

```
!BASIC BARE BONES VOIP FOR MSFC SERIES ROUTER
!BUILT IN AMILABS SUNDAY MARCH 31 2005
!VOICE ONLY QOS FEATURES
!NON NBAR VERSION
!TO MATCH UP WITH BASIC SWITCH QOS CONFIGURATIONS
!12.3 version config
!***FOR ROUTED INTERFACES****

!ip cef  !*** IMPORTANT*** CEF should already be running
!mls qos map policed-dscp max-burst 0 24 46 to 8
!excessive voice or general traffic can be remarked to scavenger
!mls qos aggregate-policer uplink 400000000 12500000 12500000 conform-
action transmit exceed-action drop
!to be used on routed ports when needed

!Sets up mappings and turns on qos globally
mls qos map cos-dscp 0 8 16 24 32 46 48 56
mls qos map ip-prec-dscp 0 8 16 24 32 46 48 56
mls qos
```

```
Not used in a core switch that is just trusting but is here for future
uses in this model. Notice how it is similar to the 3750s
configuration.

class-map match-all VOICE
  description ***VOICE RTP PACKET CLASS***
  match ip dscp ef
class-map match-any VOICESIGNALING
  description ***VOICE H323 SIGNALING PACKETS***
  match ip dscp cs3
  match ip dscp 24

policy-map NYCUTILITYBASICVOICEMSFC notice the nomenclature from the
3750.
  class VOICE
   priority percent 33
  class VOICESIGNALING
   bandwidth percent 5
  class class-default
   fair-queue  !basic wfq on remaining traffic
   random-detect dscp-based
   !set cos dscp
   !optional if switch does not support
   !internal dscp ***ONLY FOR LAN INTERFACES***

!router type percentage version depending on IOS version on msfc
!policy-map policedos   same as on the 3750
 !class class-default
   !police cir percent 50 !rate limit up to 50% of links bandwidth
     !conform-action transmit !anything within 50% gets transmitted
     !exceed-action drop      !anything over 50% gets dropped
     !service policy for DOS is to be applied to the INPUT
     !direction of an interface

!MFSC bit rate version
policy-map policedos
  class class-default
     police 400000000 31250000 31250000 conform-action transmit exceed-
action drop
!anything over 50mb drop when applied to an interface.

!FOR ROUTED INTRFACES ONLY
!interface range gi2/1 - 16
!service-policy output NYCUTILITYBASICVOICEMSFC

!Test with defaults first
!interface range gi2/1 - 16
!priority-queue cos-map 1 5
!assigns VoIP traffic to Priority queue 3
!Line card dependant on where priority queue is located
```

The MQC configuration file to remove these commands from the switch is outlined below:

```
!removal of 6500 MSFC qos commands script
!built in AMILABS march 20 2005

Removes mappings and disables QoS globally on the switch
no mls qos map policed-dscp max-burst 0 24 46 to 8
no mls qos map cos-dscp 0 8 16 24 32 46 48 56
no mls qos map ip-prec-dscp 0 8 16 24 32 46 48 56
no mls qos

Removes all other models configuration's if used.
no policy-map NYCUTILITYBASICVOICEMSFC
no policy-map NYCUTILITYBASEQOSMSFC
no policy-map NYCUTILITYADVQOSMSFC
no policy-map policedos
no policy-map hack

no class-map VOICE
no class-map VOICESIGNALING
no class-map REALTIME
no class-map CALLSIGNALING
no class-map CRITICALDATA
no class-map SCAVENGER
no class-map BESTEFFORT
no class-map BULKDATA
no class-map NETWORKCONTROL
no class-map VIDEO

!No need to do NO commands on interfaces
!removal of policy automatically removes from interfaces

!interface GigabitEthernet1/1
! no service-policy output NYCUTILITYBASICVOICEMSFC
```

As you can see the removal of a QoS configuration file for a platform not only removes the Basic model but any and all models in use. So if you moved from the Basic model to the Middle model for example and wanted to remove QoS the same command and configuration file will remove the middle model that was used to remove the basic one. NYC Utility can edit it's QoS Toolset removal configuration files to just remove one model at a time.

Another example to complete an end-to-end QoS configuration is to enable trusting on the switches in between the access-layer 3750s.

The file loaded up on the 6500 called **trustdscpinterface.cfg** loads the following MQC configuration commands.

```
!sets trusting of dscp values from access layer switches
!6500msfc native
!or use range

interface GigabitEthernet1/1
mls qos trust dscp    turning on trusting for DSCP marked packets
interface GigabitEthernet1/2
mls qos trust dscp   Same for a range of interfaces.
interface range gi2/1 - 16
mls qos trust dscp     turning on trusting for DSCP marked packets
```

The MQC configuration file to remove trusting commands from interfaces is **notrustdscpinterface.cfg**

```
!removes dscp trusting from interface 6500msfc native
!or use range

interface GigabitEthernet1/1
no mls qos trust dscp   removes trusting from the interface and the
interface is back to its default of untrusted mode.
interface GigabitEthernet1/2
no mls qos trust dscp
interface range gi2/1 - 16
no mls qos trust dscp
```

Trusting can be turned on by entering at the CLI ***"truston"*** or to turn off trusting just enter ***"trustoff"***

Now all that has to be done on a Catalyst 6500 switch to support the Basic Voice model, for example, is run the following commands:

***basicvoice***
***truston***

And to remove or rollback all that has to be done is:

***trustoff***
***remqos***

By using the NYC Utility QoS Toolset in the example above, we have just enabled QoS on the access-layer end user switches(3750s) and the switches in-between the users building Core 6500s with the same set of commands and utilizing a consistent MQC configuration commands set and naming system.

This is a basic default example, the queues allocations are all left to their defaults for NYC Utility can deploy the Basic Voice model and the defaults may be more than adequate for the initial deployment. Using a separate configuration file for advanced queuing provides NYC Utility to step up and try advanced features when needed and at its convenience. Features can be turned on and off whenever necessary provided the configuration file is resident on the device. By leaving devices at their defaults for queue allocations ensures that an advance tuning option is not used if it is not necessary.

### *7.1.4 QoS Toolset Usage Summary Using the QoS Toolset Commands*

So, for end-to-end QoS using the NYC Utility QoS Toolset to activate the Basic Voice model within a building comprising of 3750s and 6500s, the only commands necessary would be:

***basicvoice* – for 3750 and 6500**
***truston* – for 6500**

This is the basic premise of the NYC Utility QoS Toolset, a set of files with similar naming and suffixes to denote the platform all using MQC commands provides a configuration and management system that is consistent. There will be some unique platform specific configurations in some of the different files but all of the Toolset files will have the same naming and contents structure. For example the file for advance queuing on the 3750 switch will have its specific commands for that platform, but the files for the 6500 will have different commands. However, the files will utilize a similar naming and CLI name to load up when needed. So, to turn advanced queuing on both platform from the previous example all one has to do at the CLI is execute ***"basicqueue"*** and the appropriate tuned queuing configurations for that platform will be loaded to replace the defaults already in place. If you wanted to turn off the tuned queuing features and return to the default queue setup on the platform one would just enter at the CLI ***"removequeue"*** and the switches queuing configuration will return to its defaults. The word "basicqueue" was selected to denote a step up from the defaults. If this name is confusing NYC Utility can change this to something like ***"advqueue***". If any of the QoS Toolset command and file names are confusing NYC Utility can change it to whatever they wish as long as the Toolset is consistent and self documenting.

That is the advantage of the QoS Toolset, all the work is done upfront and once deployed NYC Utility can turn on and off features when needed. Advanced features can be loaded and ready so if NYC Utility wanted to turn on a feature like policing in the future it can by just entering a command at device it wants policing to run on. There are other files that can be uploaded to the switch to facilitate the toolset's operation and most are static for the platform and will never change.

### *7.1.5 QoS Toolset File Structure*

Below is a list of files on a typical switch 3750 switch. Again, all devices will have the same files, with similar MQC configurations but with different platform suffixes. Some platforms will have additional files for some additional features and others will have fewer files for simpler features. Nonetheless, all platforms shall utilize a set of commands that will be consistent across them all.

3750 alias and macro configs.txt **- loads the QoS toolset CLI commands**
BASICVOICEPOLICE3750.cfg **- the Basic Voice Model's Policing MQC configuration set**
BASICVOICEQOS3750.cfg **- the Basic Voice Model main MQC configuration set**
NYCUTILITYADVQOS3750.cfg **- the Middle Model main MQC configuration set**
NYCUTILITYADVQOSPOL3750.cfg **- the Middle Model's Policing MQC configuration set**
NYCUTILITYBASEQOS3750.cfg **- the Base Model main MQC configuration set**
NYCUTILITYBASEQOSPOL3750.cfg **- the Base Model's Policing MQC configuration set**
NYCUTILITYLOCALSRVQOS3750.cfg **- the Local File Server MQC configuration set**
NYCUTILITYQUEUE3750.cfg **- the tuned Queuing MQC configuration set for all models**
NYCUTILITYSERVERFARMPORT.cfg **- the Aggregate server farm port MQC configuration set**
NYCUTILITYTHRESH3750.cfg
dos.cfg **- rate based DOS mitigate MQC configuration set**
NOBASICVOICEPOLICE3750.cfg **- removing policing from the Basic Voice model MQC set**
NONYCUTILITYADVQOSPOL3750.cfg **- removing policing from the Middle model MQC set**
NONYCUTILITYBASEQOSPOL3750.cfg **- removing policing from Base model MQC set**
noqoscommands.cfg **-removes QoS Tool set commands from device**
qoscommands.cfg **-used to load QoS Tool set commands on device**
remdos.cfg **- MQC commands to turn off DOS features**
removeqos3750.cfg **- removal of all MQC commands on the 3750**
removeqossrv3750.cfg **-removes MQC server specific QoS configuration set**
removequeue3750.cfg **-removes advanced MQC queuing configuration set**
removethresh3750.cfg **-removes advanced MQC queue thresholds configuration set**

Below are the typical files associated with a Catalyst 6500 utilizing a Supervisor II/MSFC in Native mode. There are additional files for the use of turning on and off specific features related to the platform. Also, notice that the naming structure is the same and will be the same across all NYC Utility router and switch platform devices.

BASICVOICEPOLICEMSFC.cfg
BASICVOICEQOSMSFC.cfg
NYCUTILITYADVQOSMSFC.cfg
NYCUTILITYADVQOSPOLMSFC.cfg
NYCUTILITYBASEQOSMSFC.cfg
NYCUTILITYBASEQOSPOLMSFC.cfg
NYCUTILITYQUEUEMSFC.cfg
dos.cfg
hackattackoff.cfg **- removes NBAR based attack signature MQC configuration**
hackattackon.cfg **- enables NBAR based MQC attack mitigate configuration**
loadattackpolicy.cfg **-pre loads the NBAR based MQC DOS attack configuration**
nbaroff.cfg **- turns off NBAR on routed or Vlan interfaces**
nbaron.cfg **- turns on NBAR on routed or Vlan interfaces**
5500uplink.cfg **- applies MQC for any model to 55k uplink interface**
NO5500uplink.cfg **- removes 55k uplink MQC configuration set from uplink ports**
NOBASICVOICEPOLICEMSFC.cfg
NONYCUTILITYADVQOSPOLMSFC.cfg
NONYCUTILITYBASEQOSPOLMSFC.cfg
noqoscommands.cfg
notrustdscpinterface.cfg **- removes MQC trusting configuration from interfaces**
qoscommands.cfg
remattackpolicy.cfg **-removes the NBAR based MQC policy from the switch**
remdos.cfg
removeqosMSFC.cfg
removequeueMSFC.cfg
trustdscpinterface.cfg **- applies DSCP based trusting MQC configuration to interfaces**

Storage requirements for these files are very small. These files range in the 1 to 2kb in size. They can be stored in a QOS directory in flash or in the root directory.

### *7.1.6 NYC Utility QoS Toolset Port Classification Matrix*

The following matrix outlines the port marking, queuing and trusting state for the various types of interfaces/ports utilized by NYC Utility. This matrix is for reference purposes.

There may be special or stripped down configurations files to support certain servers or device ports that don't require a full model's worth of packet marking. These files can be loaded on the switch or router to support the server or appliance device and not require a full model's QoS MQC configuration set. Or these files can be loaded in conjunction with the Basic Voice, Base or Middle model to support a specific function. For example there is a Catalyst 5500 uplink MQC configuration file that can be applied to any of the NYC Utility models to support the non MQC QoS users on the Catalyst 5500. This file can be loaded and a QoS Toolset CLI command can be run to enable or disable on the appropriate uplink interface when applicable. A majority of the Toolset files has been created to reflect NYC Utility's inventory of switches and routers.

QoS End Device Port Matrix
http://www.amilabs.com/NYutility/enddevportmatrix.htm

### *7.1.7 NYC Utility's Custom QoS Tool Kit Command List*

The following matrix outlines the NYC Utility QoS Toolset CLI set to be used across all platforms.

| ***NYC Utility's Custom QoS Toolkit*** | |
|---|---|
| ***shproc*** | shows all non idle process utilization |
| ***remqos*** | removes the QoS configuration from the device |
| | |
| ***basicvoice*** | loads the NYC Utility VoIP only QoS model |
| ***basicvoicepol*** | loads the policer for the Basic VoIP model |
| ***basicvoicepoloff*** | turns off the policer for the Basic VoIP model |
| | |
| ***baseqos*** | loads the NYC Utility base model QoS configuration |
| ***baseqospol*** | loads the NYC Utility base model QoS policer |
| ***baseqospoloff*** | turns off the base model QoS policer |
| | |
| ***advqospol*** | turns on the advance model QoS policer |
| ***advqospoloff*** | turns off the advance model QoS policer |
| ***advqos*** | loads the NYC Utility Advance QoS Model |
| | |
| ***srvqos*** | turns on QoS policy for local file and print server ports |
| ***remsrvqos*** | removes QoS policy for local file and print server ports |
| ***srvfrmqos*** | loads QoS policy for server farm ports |
| ***remsrvfrmqos*** | unloads QoS policy from server farm ports |
| | |
| ***basicqueue*** | loads tuned queue configuration for all QoS models |
| ***removequeue*** | removes tuned queue configuration |
| | |
| ***thresh*** | loads advance QoS queue thresholds onto ports |
| ***remthresh*** | unloads advance QoS queue thresholds from ports |
| | |
| ***dos*** | loads Denial of Service policer on uplink interfaces |
| ***remdos*** | removes the Denial of Service policer on uplink interfaces |
| | |
| ***nbaron*** | turns NBAR on all interfaces |
| ***nbaroff*** | turns NBARoff on all interfaces |
| | |
| ***hackon*** | loads NBAR based DOS policer |
| ***hackoff*** | unloads NBAR based DOS policer |
| | |
| ***loadattack*** | loads NBAR based DOS policy on device |
| ***unloadattack*** | unloads NBAR based DOS policy on device |
| | |

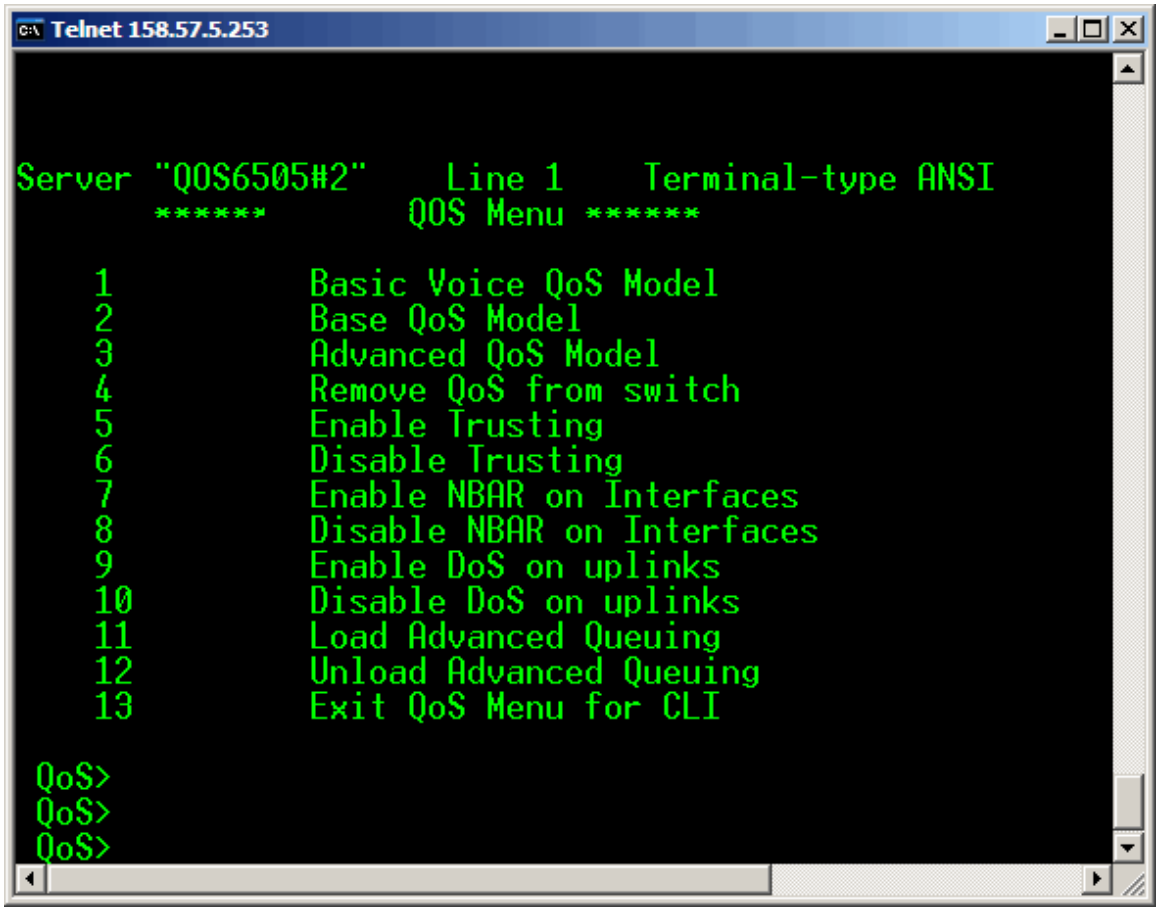
Toolset commands(continued)

| | |
|---|---|
| ***truston*** | turns on DSCP based trusting on all interfaces |
| ***trustoff*** | turns off DSCP based trusting on all interfaces |
| | |
| ***NYC Utilityqos*** | shows above command aliases on devices |
| | |
| ***qoscommands*** | loads the above command set into the router or switch |
| ***qoscommandsoff*** | Unloads the above commands from the router or switch |
| | |
| ***more*** | used to view configuration files on device for changes or online documentation. |
| | |

***Note:*** *many of the QoS Toolset configurations have been tested in a lab environment for syntax and loading accuracy but not all of the features have been tested across all of NYC Utility's platforms for there were some line cards and devices not available in the lab. DPT router platforms and POS interfaces were not tested yet as of this writing. Also, initial testing was performed against all QoS models to ensure that packet classification and marking work correctly. Queuing and policing options were tested at a basic level to ensure that the mechanics for a particular platform worked. However, in-depth stress and performance testing was not conducted due to time constraints. These tests can be conducted after this paper is reviewed to provide any additional pre deployment information and changes to the QoS Toolset or during a small initial deployment. A detailed QoS test criterion in Appendix E. has been drafted and is available to use.*

### *7.1.8 NYC Utility's Custom QoS Toolset Menu*

If using the CLI is considered difficult a menu system has been created for NYC Utility to load up on all their QoS capable devices to use. The QoS menu is a standard Cisco IOS configuration file and supports all routers, Catalyst 6500 MSFC based and Supervisor 720 models. Further support for other switches is pending. The menu just uses the same alias commands defined for the QoS Toolset so changing menu items to reflect what device options you need is just a matter of referencing the QoS Toolset CLI commands listed in section 7.1.7. Below is the output of the NYC Utility QoS Menu on a Catalyst 6500 MSFC based switch running in native mode.




***Note:*** *a discussion on MQC configuration command structure in IOS is beyond the scope of this paper. It is assumed that the reader understands what Policy Map and Class Map structures referenced are in the context of IOS and QoS. Applied Methodologies can provide a tutorial in lecture format about such constructs if applicable.*

# *8.0 Deployment of QoS in NYC Utility*

The previous sections in this document have outlined NYC Utility's QoS capable platform inventory, the models NYC Utility can implement today and in the future, design considerations and goals, design rules and recommendations and an introduction to the NYC Utility QoS Toolset.

By utilizing the NYC Utility QoS Toolset NYC Utility has a flexible toolset to facilitate the deployment of QoS in the enterprise. When it comes to deploying QoS in the enterprise careful planning and consideration must be applied to what the initial goal is and what are the end devices and applications that QoS will be applied against. QoS, depending on the model used, may need to touch every device in the enterprise so it is imperative that all of the matrixes, tools and links provided in the previous sections be updated to reflect the goals of the deployment. If the Basic Voice model is to be deployed for example, then a simpler approach and plan can be drafted for only two types of packets will be marked and trusted. If the NYC Utility Base or Middle QoS models are to be used initially, or in the future, additional applications will be classified and marked and that in turn means more end devices must be taken into consideration for the deployment. It would be detrimental as an example to enable the Base model at the access-layer but not apply a Base Server port model at the other end. Packets will only be marked at the user side but not at the server or application service side. So, when considering deployment of QoS features always think ***End-to-End.***

When deployment of QoS is discussed in the context of this section we are referring to any of the models, that is all of the NYC Utility QoS models Basic Voice, Base and Middle can all utilize the deployment approaches outlined in this section.

## *8.1 Deployment Approaches*

In NYC Utility's case there are two approaches for deploying QoS.

- **Holistic per site or entire enterprise**
- **Surgical per "section" of the enterprise**

### *8.1.1 The Holistic Approach*

The Holistic per site or enterprise approach entails selecting the site or all devices in the enterprise to have QoS deployed(become QoS capable), uploading the platform specific QoS Toolset files and schedule turning on the QoS features across the enterprise over the course of a weekend or several days.

This approach has the following benefits:

- **The site or enterprise become QoS capable in a shorter time**
- **Identify any platform or performance issues and anomalies quickly within the deployment timeframe**

This approach has the following disadvantages:

- **Issues can arise to affect other devices enterprises wide**
- **Issues in deployment or QoS anomalies may be present in more than one location in the enterprise**
- **What is learned from the deployment in one section of the enterprise may not be applied in time at a different section since all sections were deployed during the same time frame.**
- **If a configuration change is needed, though easy to do using the QoS Toolset, but not discovered until the entire deployment is complete then the change will have to be applied to all the devices again, thus causing a "re-visit" to the device.**
- **Stresses deployment and support staff, especially if multiple issues arise since many sections of the network were changed in such a short time.**
- **Difficult to correlate change related issues due to the short time frame and the amount of devices touched.**

Some corporations do prefer to just **"get it over with"** and deploy their changes over a weekend for example. This is usually a management decision due to many other factors.

In NYC Utility's case the critical nature of its enterprise network's role in the corporation and the sheer number of devices and services that its network comprise of makes the holistic approach unrealistic to accomplish.

**It is recommended** that NYC Utility utilize the second deployment approach the **Surgical per "section" of the enterprise**

### *8.1.2 The Surgical Approach*

The second **"Surgical"** approach entails selecting a section of the network between two end-to-end devices that will utilize QoS. This "section" can comprise of a switch on a floor, several switches between floors in one building, an entire building's switching infrastructure and or could expand to a specific network path to another building's end devices. NYC Utility can work its way outward from one location and deploy QoS in small sections where needed and the end-to-end path is known. This approach could be analogous to deployment of a NYC Utility circuit section in NYC Utility's electrical delivery business.

This approach has the following benefits:

- **Easy to start and rollback**
- **Can proceed to deploy and test in small steps**
- **Can learn from observation and apply lessons to future deployments/next steps**
- **Less stress upon deployment or support staff**
- **Minimizes impact to other devices and services across the enterprise**
- **Any issues or anomalies related to QoS are localized**
- **Provides a platform for staff to learn QoS and VoIP technologies**

This approach has the following disadvantages:

- **Time consuming, may take weeks to months to complete enterprise wide end-to-end deployment**

***Note:*** *Keep in mind that deployment does not have to mean enabling QoS on the device. The NYC Utility QoS Toolset files and commands can be loaded to all devices in the network section to be tested first and then enabled at each device in a sequence at NYC Utility's convenience. Once QoS has been enabled compliance testing can commence against the QoS based applications for performance results.*

### *8.1.3 Pre-deployment Planning Tasks*

The following are recommendations and tasks that must be completed before deploying QoS:

**It is recommended** that NYC Utility participate in a custom QoS education and training class from Applied Methodologies Inc, if applicable. This class can cover the basics of QoS concepts, administrating the NYC Utility QoS Toolset, and walk through mock deployments in a lab environment. The mock deployment training can be gained during a pilot deployment as well instead of in the lab or classroom to save time.

**It is recommended** that NYC Utility start out with the Basic Voice Model first for the reasons outlined Section 6.0.

**It is recommended** that NYC Utility should select a small section of its enterprise that will host the intended QoS application. A set of switches on a floor or building that provides an end-to-end path between IP Phone users and various Call Manger servers. This keeps the initial entry into QoS simple and easy to mange, test, change and rollback. Building on the initial deployment is then just a matter of deploying the QoS Toolset and turning on the features needed.

Trying a larger deployment for an entire building at first can be difficult and risky as the first step. Also, if the initial QoS path must be across buildings over the MAN/WAN then additional time will be required to plan for all possible end-to-end paths between IP Phones. Since NYC Utility's network is very resilient and provides path diversity all of those paths will have to be identified and QoS applied to them to ensure that any return trip QoS based packet does not cross a non QoS enabled device. If this happens then performance results will be skewed.

**It is recommended**, if possible, that any pilot VoIP deployment proceed without QoS applied first. This approach will provide NYC Utility with critical operational and behavior information about VoIP in general and will also provide a "before" snapshot of the VoIP's QoS before QoS is deployed. MOS(Mean Opinion Score) and PESQ(Perceptual Evaluation of Speech Quality) scores can be compiled for "before" and "after" QoS analysis. This approach will also help NYC Utility determine when QoS is applied, whether the defaults provided a positive, negative or no change in VoIPs operation/behavior. Plus, this approach gives the VoIP support team to test and identify its final production based solution in terms of signaling protocols, codec etc. This information can then be used to finalize the QoS Toolset configuration files for pilot deployment.

If VoIP is to be fully tested in a separate VoIP lab then **it is recommended** that the QoS Toolset be tested along with the VoIP solution. These tests can be conducted in a before and after approach and the detailed test criterion outlined in ***Appendix E.*** can also be used in this lab.

### *8.1.4 General VoIP Deployment Tasks*

1. **Ensure Codec selection and bandwidth per codec is know/defined**
2. **Ensure jitter and PCL budget is know/defined**
3. **Ensure end-to-end delay budget is know/defined**
4. **Identify if any features such as VAD and echo cancellation are in use on IP phones**
5. **Identify the Signaling protocol(s) and all of its associated ports or range of ports(these will be added to the QoS Toolset Basic Voice model's ACL)**
6. **Identify if CAC features to restrict calls and call bandwidth will be used or if QoS will provide(this determines if the QoS Toolset end user per port policing MQC is required)**
7. **Pre test voice calls for packet response times, MOS, PESQ and Jitter statistics before QoS is enabled on the path. This is the before snapshot.**

### *8.1.5 General QoS Pre-deployment Tasks*

1. **All interfaces on routers must have the correct bandwidth statement. QoS utilizes the Bandwidth statement and not the line/clock rate for bandwidth percentages. This should have been completed from the EIGRP migration project.**

2. **Ensure that CEF is on first for all routers and switches running in Native mode**

3. **Ensure that all routers and switches in the network section to be tested are compliant to the minimum recommended IOS outlined in Section 4.3.2**

4. **Ensure that the access-layer switches are compliant with minimum recommended hardware platforms outlined in Section 4.2**

5. **Ensure that the Catalyst 6500 platforms in between any designated pilot QoS section are upgraded to Native mode at a minimum with the Supervisor module MSFC-PFC2 or a Sup720.**

6. **Ensure that the Catalyst 6500 platforms in between any designated pilot QoS section have line cards that support a priority queue. Refer to Appendix F. for the Catalyst 6500 line card inventory.**

7. **For the platforms identified in the selected deployment section's end-to-end path, please review the platform's related QoS testing notes in Appendix C. for information on issues and behavioral characteristics for deployment and operation pertinent to that platform. This section will outline if the platform's interface resets when QoS is deployed or enabled or any issues that the installers should be aware of when dealing with the platform.**

8. **Conduct a basic traffic utilization baseline analysis on the QoS path before and after QoS is deployed.**

9. **Update the QoS Toolset configuration files for the model used for the test. In this case the Basic Voice model will be used so configurations files for the appropriate platform's Basic Voice model will have its ACL configuration updated to reflect the signaling protocol used.**

10. **Update the QoS Toolset configuration files for the switches in between the end user access-layer switches. In this case if the 6500s are used in between 3750s then the Basic Voice model configuration files need to be reviewed to ensure that all interfaces and line cards definitions are accounted for. This applies to the Basic Voice main QoS file and to the "truston" and "trustoff" files at a minimum.**

11. **Upload the QoS Toolset files and load the QoS alias command set file into the device.**

### *8.1.6 Deploying QoS for the Basic Voice Model local Within a Building Example*

Once all of the steps outlined in the earlier section have been completed all that remains to do is the following:

1. **Schedule the change window to enable QoS on the devices**

Devices should be QoS enabled in sequence to provide a window of time to identify if a devices's changed state(from non to to QoS enabled) had an affect on any other functions(routing protocols etc). This approach facilitates quick rollback by using the **"remqos"** on the device last changed.

2. **On each device, in this case for the Basic Voice model on the 3750 and 6500 switches the following commands are entered**

**3750 switches**

***basicvoice***

**for any 6500s in between**

***basicvoice***
***truston***

### *8.1.7 Rollback Changes*

There are two methods to rolling back the changes to the devices.
One is to just issue the QoS Toolset command ***"remqos"*** and this will disable QoS globally on the device and remove all QoS MQC configuration commands.

The other option is to go into configuration mode and just enter the ***"no mls qos"*** command. This will "disable" QoS **BUT** leave the QoS Toolset's MQC configuration commands in place. So, if QoS is to be enabled at a later date and the configuration options are still in place all that has to be done is execute the "mls qos" command on the platform.

### *8.1.8 Post QoS Deployment Tasks*

1. **Conduct post deployment compliance testing using the show commands outlined in the QoS Toolset documentation.**

2. **Conduct necessary VoIP testing to determine if default QoS settings are adequate to maintain the defined VoIP quality levels.**

3. **Update the necessary post deployment documentation**

4. **Apply lessons learned to the next network section that QoS will be deployed to.**

### *8.1.9 Additional Deployment Tasks(future)*

Additional tasks for future deployments across the enterprise and for the NYC Utility Base and Middle models. The same tasks outlined above apply here as well but were not repeated.

1. **Ensure that the core network routing and switching platforms are compliant with minimal hardware and software requirements for QoS**

2. **Ensure that applications ports and flows have been defined(matrixes are filled out)**

3. **Conduct a Concord or a traffic/protocol based analysis of the sections that will have QoS deployed to before changes are made to see if there is a difference in traffic and utilization behavior.**

4. **Baseline of pre QoS deployed applications such as a voice call and some other traffic to see the difference when QoS is deployed across the enterprise**

5. **Identify if critical enterprise wide servers are located on non QoS capable switches. If this is the case either upgrade the switch where the servers reside, move the ports to a new switch(6500 or 3750 aggregate) or utilize the QoS Toolset uplink commands to implement Marking and Classification of return path server traffic on the non QoS capable uplink port. This means that traffic between peers on the non QoS capable server do not have QoS.**

6. **All server farm and server ports must have QoS enabled so packets leaving the server farms also have QoS applied**

7. **Conduct post deployment routing protocol convergence test for voice(if applicable)**

8. **Possible testing of DPT or backbone switch failure to see if voice traffic continues (Schedule off hours) during a major outage(if applicable)**

9. **If SRST compliance testing is required then ensure that router ports on SRST routers acting as gateways must remark packets and have a service policy for outgoing bandwidth allocation for normal serial interface processing. Plus, if voice packets are sourced at the router from SRST then they must be tagged going back into the ingress local site network.**

### *8.1.10 Post Deployment Documentation*

This document and The QoS Toolset document provide a majority of the documentation regarding the QoS tools used in general, but if the deployment is going to be done in sections some form of accounting must be kept to list what devices are already QoS enabled, have the QoS Toolset deployed, but not enabled, and a raw non QoS device on the network. This is critical for troubleshooting and future deployment activities. NYC Utility must keep track of all QoS enabled switches and routes either in a matrix, table or in a diagram showing the QoS enabled path.

NYC Utility should investigate if it is at all possible on the NMS OpenView map icons the ability to put a Q on the device Icon to indicate that it is QoS enabled. This is also very helpful in identifying QoS enabled end-to-end paths and for troubleshooting.

CiscoWorks can keep a historic record of the NetJob jobs, if used, of executing the QoS Toolset commands.

## *9.0 NYC Utility QoS Don'ts*

1. **Do not enable QoS on a device that is not intended for QoS or does not have the QoS Toolset uploaded to the device**

2. **Do no enable Auto QoS in general or enable it and then enable a QoS Toolset model on top of it**

3. **Do not switch user and server port devices across switch ports on a QoS enabled switch or switch stack. You may inadvertently move a user from a QoS defined port to a local server port on the same switch or switch stack that has its own QoS definitions applied or none**

4. **Do not enable QoS on SPAN ports**

5. **Do not remove/disable QoS on a switch in the middle of a QoS end-to-end path. This creates a QoS black hole where the packets will be untrusted crossing the QoS disabled switch and thus have its DSCP markings rewritten. When the packets enter the QoS enabled domain their markings are not reinstated and thus will not utilize the QoS options on the rest of their journey to their destination. See appendix C.**

6. **Do not mix and match different NYC Utility QoS Toolset MQC configuration sets. For example, do not have the NYC Utility Middle model configured on one set of switches and the Basic Voice or Base model on other switches.**

7. **Do not load multiple NYC Utility QoS Models on the same switch**

8. **Do not move up to the Base or Middle Models without planning for and applying the appropriate models against general server farm ports and specialized application server ports.**

***Note:*** *A switch in the middle of a QoS end-to-end path can have the QoS Tool set uploaded and QoS enabled but even if none of the QoS Toolset features are used(truston for example) the packets will cross the switch and their DSCP markings will remain intact but QoS features will not be in affect. Refer to Appendix C. for the 6500MSFC testing notes for more details.*

## *10.0 Additional Testing Tools*

NYC Utility can utilize Cisco's Service Assurance Agent(SAA) features embedded in routers with IOS release levels over 12.1 as an additional testing and troubleshooting tool. SAA provides an IOS based testing suite for different applications and some specific tools for testing VoIP in general. The following link is from Cisco's Packet Magazine and provides an excellent tool that NYC Utility can utilize in testing its VoIP and QoS deployments.

**PACKET VOL. 16, NO. 3, THIRD QUARTER 2004**

**Tech Tips and Training: Is Your Network Ready for Voice? Measuring Delay, Jitter, and Packet Loss for Voice-Enabled Data Networks**

http://www.cisco.com/en/US/about/ac123/ac114/ac173/Q3-04/dept_techtips.html

SAA can be utilized on production devices or a pair of older 2600 routers with a 12.1 or above IOS can be deployed as just "end point" client devices on either side of a VoIP and QoS deployment section and then use SAA to generate VoIP related traffic before QoS is deployed and after to identify any performance differences relating to Codec used, TOS and Jitter.

## *11.0 Managing and Tuning QoS in NYC Utility*

There are various tools that NYC Utility can use to manage its QoS environment. When referring to managing QoS there are two areas that fall under the management of QoS:

- **Managing the QoS Toolset**
- **Managing the production QoS environment**

## *11.1 Managing the NYC Utility QoS Toolset*

Managing the QoS Toolset entails administration of the NYC Utility QoS Toolset MQC files per platform. These files may change in a minor way from deployment specific platform and IOS version changes. Subtle but different versions of the same files may be created for different platform profiles or deployment considerations. For example, Catalyst 3750 24 port or 48 port switches. There can be a QoS Toolset file using the same suffix **BASICVOICEQOS3750.cfg** but one version may have an interface range for 48 ports and another for 24 ports because a different member in the network support team has changed the interface range in the original file and saved a copy of the whole directory for his specific deployment. This is acceptable to do but it is from these files that NYC Utility can customize its QoS environment as its needs change in the future and have them reflected in the QoS Toolset.

It is the responsibility of the network support team to ensure that these files are up to date, not mixed up with other deployment files and always available. These files are the foundation of the NYC Utility QoS environment and can be expanded upon from their initial configuration. So, version control is an important consideration when managing these files.

The initial set of QoS Toolset files currently resides in the QoS Lab server under the following directory structure. These files should be stored in a production server central location and backed up regularly.

**\NYC UTILITYQOSCONFIGS**
- 2600qosconfigs
- 2800qosconfigs
- 3550qosconfigs
- 3700qosconfigs
- 3750qosconfigs
- 3750srvconfigs
- 3800qosconfigs
- 4500qosconfigs
- 7200qosconfigs
- 7500qosconfigs
- CATMSFCqosconfigs
- CATSUP720qosconfigs
- DPTqosconfigs
- Generic Router IOS QoS Toolset

## *11.2 Managing the production QoS environment*

Even though Cisco has made strides in providing a consistent configuration solution for cross platform QoS – MQC, there is very little in terms of effective management tools available beyond the CLI to monitor and make changes to a post deployment QoS environment. Plus. there were different tools created at different times for different platforms so a consistent single source tool for managing QoS is still evolving. The ideal management tool should complement the NYC Utility QoS Toolset and make the administration and deployment of the QoS Toolset files more efficient.

The following tools are available and are discussed in the context of NYC Utility's use for managing its QoS environment.

### *11.2.1 Cisco's CiscoWorks QoS Policy Manager(QPM)*

QPM provides many of the features that you need when you get serious about deploying QoS across an enterprise. This product is the single source QoS management tool for cross platform QoS environments. The following list outlines some of its features

- **Enables you to define a QoS policy based on business goals**
- **Automatically configures some or all network devices with QoS features, based on the QoS policy described to QPM. However, it should be noted that initial testing of applying the same QoS commands to different devices resulted in different results, such as interfaces resetting, thus causing a link to go down for a moment. When using QPM to deploy or change QoS functions be aware that QPM may mask this behavior.**
- **Loads the correct configurations automatically**
- **Enables you to monitor the device configuration to make sure no one has made changes to them. If the configurations have been changed, you can use QPM to restore the original configuration.**

It should be noted that as of this writing initial testing of QPM proved to be less than satisfactory. The product had issues with manually discovering some QOS enabled devices and further testing on its capabilities were not conducted due to time considerations. **It is recommended** that additional time is allocated for the testing of QPM to determine if such a product can help facilitate NYC Utility with its QoS deployment and management and complement the NYC Utility QoS Toolset.

### *11.2.2 Cluster Management Suite(CMS)*

Cisco CMS provides an integrated management interface for delivering intelligent services, such as multilayer switching, quality of service (QoS), multicast, and security Access Control Lists (ACLs). Thus, Cisco CMS allows administrators to take advantage of benefits formerly reserved for only the most advanced networks without having to learn the command-line interface (CLI) or even the details of the technology.

On certain classes of switches such as the 3550 and 3750 Cisco offered with its IOS license an optional web based management system for managing switch clusters, hence the name CMS. CMS consists of a set of html and other java files located on the switch's flash subsystem that provides a very good and functional GUI for managing the switch in general. What was ideal about CMS is that it had a QoS management and wizard GUI feature built into it so one can administer Policy and Class maps, ACLs, and see what is configured for each interface. Combine this tool with the QoS Toolset files loaded up and administering the files on a local switch is simpler and reading the QoS Toolset files is easier as well. All that had to be done was enable HTTP on the switch, then browse to the switch, use CMS to administer your QoS Toolset feature, disable HTTP and you are finished.

However, CMS is only available on older versions of the switches IOS 12.1(19) and below and was not available for different platforms such as the 6500. It appears that Cisco is moving away from CMS and does still provide a GUI HTTP based tool for the switches but the QoS features are no longer present, just tools for measuring performance.

### *11.2.3 SNMP*

Cisco provides a couple of proprietary SNMP Management Information Bases(MIBS) that provide some good information about QoS operating in a router. The Class-Based QoS MIB (CBQoSMIB) contains variables that describe the MQC configuration commands in a router. This MIB also includes statistical variables, which are essentially the same statistics seen from executing the show policy-map interface command.

The ***CBQoSMIB*** provides statistics for packets before a policy map has been processed, and afterwards. You can see statistics about packets before the PHBs have been applied or treated and after. This is helpful in troubleshooting QoS related issues. QPM uses this MIB for many of its reports and management functions.

**It is recommended** that NYC Utility look into this particular MIB and any other Cisco based QoS MIBs per platform to see if they can be integrated into the HP OpenView management system.

### *11.2.4 PFC QoS Statistics Data Export*

**Note:** Release 12.1(11b)E or later supports PFC QoS statistics data export.

The PFC QoS statistics data export feature generates per-LAN-port and per-aggregate policer utilization information and forwards this information in UDP packets to traffic monitoring, planning, or accounting applications. You can enable PFC QoS statistics data export on a per-LAN-port or on a per-aggregate policer basis. The statistics data generated per port consists of counts of the input and output packets and bytes. The aggregate policer statistics consist of counts of allowed packets and counts of packets exceeding the policed rate.

The PFC QoS statistics data collection occurs periodically at a fixed interval, but you can configure the interval at which the data is exported. PFC QoS statistics collection is enabled by default, and the data export feature is disabled by default for all ports and all aggregate policers configured on the Catalyst 6500 series switch.

**Note** The PFC QoS statistics data export feature is completely separate from NetFlow Data Export and does not interact with it.

Using the PFC QoS export options is for the 6500 series only.

### *11.2.5 Quality of Service Device Manager(QDM)*

QDM was an http based add-on for 7500 routers similar to CMS on the switch. This product is no longer supported and not available for download.

### *11.2.6 Access Control Lists(ACL)*

Depending on the platform ACLs are used to identify packets for service class matching so packets can get marked or policed. For the Basic Voice model there will only be two ACLs, one for the RTP VoIP traffic port range and one for the signaling protocol port range NYC Utility utilizes. Managing the ACLs for the Basic Voice model and any policing for DOS is not very time consuming or difficult task. The same applies for utilizing NBAR for any switch routed interface or router interface for the Basic Voice Model.

It is when the other models are used do the ACLs become larger and more granular and thus some time is required to ensure that ACLs are accurate and consistent across all the files in the QoS Toolset. There is currently no tool to manage ACLs from a GUI perspective to inspect and easily make insertions and deletions on a specific ACL. Also, newer versions of Cisco IOS does provide support to insert ACL entries in-between others. Cisco did have a CiscoWorks ACL manager plug-in for Cisco Works but this product is no longer supported or available. QPM is supposed to have an ACL management element to it but this has not been tested of this writing.

### *11.2.7 Command Line Interface(CLI)*

The NYC Utility QoS Toolset provides a consistent set of commands to turn on and off specific features at the CLI and these commands are consistent across all platforms. But these commands are for the deployment, removal and turning on and off QoS features only. To observe actual QoS related statistics there are a series of CLI ***show*** commands available.

Most of the MQC CLI show commands are similar and consistent across switch and router platforms but there are some, discovered from lab testing, that show different output from executing the same command. Please refer to ***Appendix C.*** for the particular platform's testing notes on any issues with CLI show commands. There are many commands and their pre and post execution output nuances to cover in this paper so these commands may be covered in detail in the QoS Toolset user guide and in any QoS training classes Applied Methodologies conducts for NYC Utility personnel.

**3550 and 3750 switches**

show mls qos
show mls qos interface
show mls qos map
show mls qos interface policers
show mls qos statistics
show class-map
show policy-map
show policy interface
show mls qos interface (interface number) statistics
show mls qos interface queueing
show mls qos interface buffers
show msls qos queue-set
show mls qos maps cos-output-q
show mls qos maps dscp-output-q

**4500 switches**

For the 4500 switch the above commands from the 3550 and 3750 apply but are used without the mls keyword. For example show qos map, show qos interface.

The unique commands for the 4500 would be:
show qos dbl
show qos maps dscp tx-queue

**6500 switches**

For the 6500 switches the show commands are similar to those of the 3550 and 3750

The unique commands for the 6500 different show commands:
show queueing interface
show queueing interface (interface number)
sh tcam ?(advanced )
sh tcam counts
sh tcam interface gi3/1 qos type1 ip det

It is interesting to note that the one Native mode command "show queuing interface" commands provides the output that several CatOS commands would have been required to use.

**For the router platforms**
The router platforms all utilize a common CLI command set for monitoring QoS, and NBAR features.

show policy
show policy interface
show policy interface (interface number)
show class
show ip access-list
show ip NBAR port-map
show ip NBAR protocol-discovery
show interface

QoS at a basic level should be a feature that can be turned on and left alone. However, that is not always the case as you scale its use. As more applications are classified and fall into different classes of service additional administration time will be required by the network support team to maintain the configurations and ensure that matrixes are up to date. This is also the case for updating any DOS ACLs or NBAR signatures. Also, having the various QoS related matrixes and the QoS Toolset as tools do make things easier and can help reduce administration time. The time required to maintain QoS should not be very large on a monthly basis even if the Middle model were used with advanced features. The benefit of the QoS Toolset is to have the configurations done ahead of time and also distributed so you can turn them off and on as needed and once they are in place and operating, minimal additional time is needed to maintain them.

There will be IOS changes that may provide enhancements or syntax changes to the MQC per platform so the network support team must be cognizant of this before upgrading a device or placing a new device with a different IOS version and applying the QoS Toolset features. **It is recommended** that the QoS Toolset files be loaded on the new device and IOS before production deployment to ensure that they load and work properly.

# *12.0 Troubleshooting QoS in NYC Utility*

Troubleshooting QoS entails an understanding of how QoS works and the applications that QoS is based upon. The reason that this understanding is required is to help the troubleshooter become more productive in quickly identifying whether the problem is QoS related or not. For example in the case of VoIP an issue could occur where the voice quality suddenly becomes poor for a user months after QoS and VoIP have been deployed. This could be the result of an echo cancellation or jitter buffer issue on the user's phone, a codec issue or CAC issue with regard to allocating call resources for this user, if used. All of these issues and others can appear to be QoS related and may be masked or intensified by QoS.

The worst issues that can happen resulting from a QoS problem is that when congestion is present and QoS is not tuned or operating properly some critical packets such as VoIP may get dropped or the call quality suffers. Another issue could be a policer that is left on or improperly configured and it is policing to drop or markdown packets from a section of the network when it shouldn't. QoS because of its per device and PHB nature cannot cause network wide meltdowns, unless it was specifically configured to do that and that is not the case here. In most cases a QoS issue is localized to the device or upstream/downstream from it.

Routing protocol packets will always have a PaK_Priority assigned to them on the local per hop router and are only significant across a neighboring link segment so a QoS related issue affecting routing protocols is not very common. It is only when policing an aggregate of traffic where neighbor state protocols may be affected but again this is an extreme case and is caused by poor configuration planning.

Another thing to keep in mind is that the access-lists used in QoS are just for marking packets and not for blocking them and thus never applied in a manner to an interface to block traffic. Incorrectly marked packets will just be handled at a different class level and may become a candidate to be dropped but only when congestion is present.

Traffic policing is also something to keep in mind. The policer's goal is to just mark down traffic or just drop traffic based on an ACL based Class/Policy map or just drop a certain percentage of bandwidth. Depending on the policer's use for DoS mitigation or CAC this operation is expected.

Troubleshooting QoS related or non related issues becomes trickier when many different platforms are in between any two communicating peers. In this case, as is in NYC Utility's environment, knowledge of not only the basic aspects of QoS is helpful but also the platform's specifics of implementing and managing QoS. Keeping the QoS configurations and functions as consistent as possible will help mitigate cross platform discovery and troubleshooting time but there will always be some level of platform related QoS detail that may have to be identified to troubleshoot the problem further.

By classifying the more common causes of issues related to QoS we have a reference point to follow when problems do arise. The next session covers this.

## *12.1 Common QoS Related Issues*

### *12.1.1 Issues Caused from QoS*

**Caused from QoS – invalid configuration**

An improper use of a configuration command or a typo such as an access-list typo or the incorrect port number was defined in the ACL. Even though the QoS Toolset helps prevent this by having the files created and tested prior deployment sometimes typos and other mistakes do occur. Fortunately if a typo or mistake is present in a QoS Toolset file the same issue will be repeated in the devices it is deployed to thus provide a quick pattern to identify the issue and trace the problem back to its source. In this example if an ACL issue were the cause then packets would not be matched and marked properly. The CLI commands and a sniffer trace can verify this quickly.

**Caused from QoS – Default queues and thresholds outlived their allocations or additional level of congestion is present.**

In this case the default (per port) queue allocations used are experiencing issues due to the increased volume or the initial volume of traffic when QoS was first deployed. The traffic level crossing the port is just too much for the defaults. If this were the case the CLI commands can show if packets are being dropped on the output queues or if queue exhaustion is present. Regardless of whether WRED or WTD is in affect the queuing algorithms only take affect if congestion is present. So, for an interface(fast or gigabit) to be heavily congested in the first place for a user or uplink port regardless if QoS is available would suggest another issue is the root cause. However, if the congestion was always there in the form of valid traffic levels but just too much for the initial defaults or became available at a later time post deployment, the QoS Toolset already has a pre tuned recommend queue and threshold MQC based command to turn on for the platform experiencing the congestion. This approach should be considered the last resort approach for a general traffic and protocol analysis should always be conducted first to determine if the congestion is a natural result of user traffic volumes or from a component failure issue. This type of issue also applies to slower serial links.

Also, DoS issues can mask this type of behavior as well, so it is imperative that the traffic analysis is always conducted first to determine if any QoS Toolset features are needed like DoS tools or to turn on the tuned platform features.

Always check the traffic levels and types before changing or adding QoS features. Always remember that QoS tools are not meant to **“find”** additional bandwidth on already congested or heavily utilized links.

**Caused from QoS – Platform issues, future IOS bug or version limitation**

These issues could result from an IOS bug, for example, removing a policy map command causes a switch to crash(see ***Appendix C.*** for an example from a 3750), or a bug in the packet marking function causes a packet to exit an interface with the wrong marking. Other issues to consider are version or platform limitations. Router platforms support NBAR but DPT router platforms do not. The ***"sh policy-map"*** command shows output statistics on router platforms but on switch platforms the same command provides the same output but no statistics due to a know hardware ASIC limitation(see ***Appendix C.*** for an example from a 3750). These examples must be kept in mind when analyzing QoS related issues. Such issues can be mitigated by testing IOS and platform versions of QoS Toolset options before future deployments to identify any platform or version idiosyncrasies.

**Caused from QoS – Deployment of QoS affecting network operation**

This issue is a platform specific issue. Some platforms reset the interface when a QoS Policy map or other feature is applied or removed. Others do not. The QoS Toolset commands have been tested against as many platforms NYC Utility supports as possible and a majority of the platforms allow for real time non intrusive or reset of an interface when turning on and off QoS Toolset features. The details of the platforms that do incur an interface reset are outlined in the individual platform testing notes in ***Appendix C.*** and should be reviewed for pre deployment planning. These platforms were not listed here in a matrix for this document has too many matrixes already and the reader may go nuts.

**Caused from QoS – QoS polices affecting application performance**

This type of issue could be the result of bandwidth and queue allocation policies specified in the Base and Middle model not working properly or incorrectly planned. For example if the business rule is restrict call volume using QoS and not CAC and to allow only 2 calls per user port and a per port policer could already be in place to allow only 2 calls, but for two G.711 64kbs calls. Then at a later date NYC Utility changes its codec to G.729a or 723. The change in codec results in calls amounting to much less bandwidth per call and now the user can make more than two simultaneous calls out of his port. VoIP and QoS are working properly now but the business rule has been violated. The same applies in the reverse where an uplink policer could have been improperly planned for aggregate uplink traffic levels and when turned on does not provide any relief for it is still allowing too much traffic or in the reverse scenario too little traffic through.

Another example is packets marked at the source user port for one marking, lets say **AF31** and at the server's port for the round trip reply the response packet is marked **CS0**. The return packet will have a lower class of service applied to it on the return path and if congestion is present anywhere on the path back to the user's workstation this packet has a better chance of becoming a candidate to be dropped.

This is where all of the matrixes and tables in this document come into play to ensure that defined application flows, bandwidth allocations, queue allocations, class markings and platform specific configuration items are all reviewed so consistent configurations and tuning features are applied and such issues will not or never be present.

The example given of the problem outlined in the last paragraph can look like an application or application server related issue. It is always prudent when such an issue occurs in a post deployment QoS environment is to take the following measures:

1. **Identify the source and destination end switches and user workstation or IP Phone(IP and subnet pairs) of the problem – the source user port switch and corresponding destination server or user IP Phone switch.**

2. **Verify that the path taken between the peers is the same from both ends. This is required, in a partially QoS enabled environment, to ensure that packets are not going through a non QoS enabled section of the network.**

3. **Verify the QoS configurations on both peer end switches to ensure the QoS configuration in use is in compliance. The correct QoS Tool set MQC policy is in place, ACL port ranges etc. For example, a user on one switch is running the Basic Voice Toolset MQC configuration and another user on another switch is running the NYC Utility Base Model Toolset MQC configuration set. This would still work but the packets are marked the same for voice and call signaling across the NYC Utility models but should be checked nonetheless.**

4. **Check to see if a policer is turned on at any point between the peer's paths.**

5. **Utilize the CLI commands to check QoS operation- see if packets are marked going out and marked packets are coming in. Check interface utilization and interface queuing information for any dropped packets.**

6. **If all looks compliant at the end switches proceed from the user end switch to the next hop switch and conduct the same tests listed above against all the switches in between the user and server pair or user to user IP Phone pair. Also check for trusting status and priority queue allocation on the switches in between the VoIP peers.**

***Note:*** *this section covered QoS specific issues and troubleshooting methods. VoIP based systems have their own set of troubleshooting methods and approaches and these should be considered and used in conjunction with the QoS troubleshooting details outlined in this section.*

### *12.1.2 Issues Caused From External Events*

**Caused from external events – Interface faults or flapping**

Interface faults or flapping interfaces do have some relevance to the local(on the local switch) QoS behavior in a switch in terms of any resulting congestion on an uplink or the actual port when the interface comes back up. QoS can be an asset in this scenario for if an interface just came up and it was flooded with pent up traffic from an application(s) retransmission the QoS mechanisms in place can smooth out the bulge of traffic and the higher class traffic will still be able to get through, the lower classes will also get through when the bulge ceases. ***Note:*** *this condition has not been tested and should be tested before deployment to see if such a benefit can be available.*

**Caused from external events – Device performance or hardware issue**

If a device, router or switch is experiencing a problem the practices that NYC Utility currently follows to rectify such conditions apply. Where QoS is related in this area is that if traffic was flowing through the device and it is no longer available then the converged paths around the device should be QoS enabled. This is to be kept in mind when conducting the "surgical approach" of deploying QoS to sections of the network. If a QoS enabled device in between two IP Phones faults and is out of service the paths around the device may not yet have QoS applied so a residual issue of call quality may ensue from the initial device failure. So, if a QoS enabled device is experiencing an issue that has nothing to do with QoS, still consider the end-to-end ramifications for the QoS based applications.

**Caused from external events – Application server issue**

Any type of application issue is always somewhat more difficult to troubleshoot and resolve for there are many factors involved. QoS does add another factor to application related issues. Whether applications are QoS based(marked) and utilize a QoS enabled path or if the application is not QoS based(not marked) but it still uses a QoS enabled path the accepted practice of application analysis that NYC Utility utilizes today still applies when issues are present. There are some additional steps to keep in mind:

1. **Ensure that the QoS based application issue is not the result of QoS handling of the application across the network. This has relevance for TCP based applications as well. For TCP analysis, retransmissions, slow start, and other flow behavior may result from server congestion, network issue or QoS marking down the traffic. The analyst looking into the application performance issue must keep this in mind.**

2. **Did the performance issue appear immediately after QoS was deployed to the section?**

3. **Using the QoS Toolset the analyst can turn off QoS for the section and monitor the application and then turn QoS back on to see if a change in the application's performance behavior was resulting. (dependant on other traffic and off hours scheduling)**

4. **The same approach as above applies to non QoS based applications traversing a QoS based path. Yet in this case these applications would just belong in the Best Effort handling class and if no congestion is present then QoS is not contributing to the application's performance issue.**

5. **The analyst must always keep QoS in mind when reviewing application logs, protocol analyzer traces to either eliminate QoS from the variables of possible contributors to the performance issue.**

6. **There has been no documentation or documented results showing that QoS causes such problems to date. This does not mean it could not happen but it also means that not every problem in a post QoS environment is the fault of QoS. Application related issues that have QoS had nothing to do with may appear to be QoS related and QoS issues mask an application problem could also occur. The analyst must be aware of this when troubleshooting.**

### Caused from external events – Workstation or IP Phone issue

A faulty workstation or its NIC is self evident

### A faulty IP Phone may exhibit these some of these characteristics:

- **The Cisco IP Phone's Trusted-device packets are not getting marked(hence the practice of marking at the switch) so everything works fine but the RTP packets in one direction are not getting marked and hence will not have the priority queues available to them.**
- **Jitter buffer**
- **Echo cancellation codec issue**
- **Delay in codec performance causing jitter**
- **VAD(if used) and PCL(if applicable to Cisco)**

These issues, though rare, could crop up from a firmware upgrade, poor lot of phones manufactured, or just Murphy's Law becoming applied to a single phone.

The point here is that when troubleshooting a QoS issue, don't presume that QoS is broken immediately network wide and troubleshoot from there. Start at the source, eliminate all the QoS and non QoS components and work outward, for sometimes the problem may be right at the source or peer end and thus time can be saved from not looking at QoS section wide before narrowing down the problem to a workstation or phone issue. QoS related issues can be atomic as well. Such issues can be local to the device and affect only devices originating at the device or crossing the device.

**Caused from external events – Call Manager server issue**
There are issues relating to Call Manager servers that must also be considered when troubleshooting QoS. The relevance here is related to VoIP based applications.

Some immediate things to check:

- **Ensure that QoS is applied to the ports where the Call Manager server's reside**
- **Verify that the signaling protocols used port ids match that of the QoS Model in use and configurations applied to the user access-layer switch**
- **Packets marked leaving the Call Manager server are marked properly even if QoS is applied to the port**
- **Check that the path signaling packets take to and from IP Phones follow a QoS enabled path.**
- **There are other Call Manager specific issues not covered here and left to the Call Manager installation engineers and administrators to identify and document prior any QoS or VoIP deployment.**

### *12.1.3 Troubleshooting Tools required*

- **Knowledge of QoS specific CLI tools**
- **QoS Toolset files – to quickly and easily turn on and off QoS features**
- **Protocol analyzer**
- **For voice related issues an RTP or Jitter analysis tool**
- **Codec bandwidth calculator – mentioned in Section 2.2.1**
- **SAA tools – used to reproduce issue**
- **Traffic generation tools for stimuli and lab testing**

# *13.0 Scaling and future proofing QoS in NYC Utility*

Scaling QoS in the enterprise comprises of two components, QoS Model scaling and platform scaling

## *13.1 QoS Model Scaling*

Scaling upwards or vertical scaling from one QoS model to another is available as more application become QoS aware. As discussed in Section 6.0 there are several NYC Utility QoS models that NYC Utility has available to utilize based on its current and future applications needs. As more applications require the services of QoS the additional models can be turned on and utilized. If a QoS model needs to be changed the models can be easily edited and applied(this is where all the matrixes in this document come into use) The NYC Utility QoS Tool Set enables NYC Utility to easily move up from the Basic Voice Model to the Base Model and then up to the more granular Middle Model. These models provide a vertical hierarchy to enable additional applications to utilize QoS.

For example, when the time comes for additional applications that require QoS such as Video, the Base Model can be utilized. Subsequently, when additional applications and further granularity is needed beyond the Base Model the Middle Model can be implemented. By utilizing a best practice foundation of models based on Cisco's Base Line Model and RFC based PHB standards ensures that NYC Utility can scale its QoS needs in a vertical manner and the practices used are supported industry wide. The NYC Utility QoS Toolset already provides all the tools needed to implement all the NYC Utility QoS models for today and in the future with advanced tuning options for all models as well.

## *13.2 Platform Scaling*

The other aspect of scaling and future proofing QoS is in the platform arena or horizontal scaling. NYC Utility must ensure that its vendor(Cisco) remain consistent in terms of its QoS offerings. New devices and IOS versions must continue to use the MQC and queuing structures for line cards and devices are compatible and act consistently in behavior from what is installed currently. This ensures that future QoS enhancements will be available on current and future platforms and upgrades for a platform do not disrupt the QoS architecture. By utilizing vendor baseline models and a vendor that supports RFC internet PHB standards helps to ensure that NYC Utility's investment in QoS tools, features and methodologies will remain intact for the duration of NYC Utility's platform hardware and software lifecycle.

## *14.0 Wireless*

Wireless users and phones were not directly covered in this strategy as an initial end-to-end deployment consideration. However, it is listed here as a consideration since its use is growing due to the changes in wireless technologies at the time of this writing. There is currently no operating QoS mechanism in the 802.11 series of standards. There is a draft specification called 802.11e which provides QoS bits to be used for applications such as VoIP. However, at the time of this writing 802.11e has not been ratified.

There are several ways to achieve a small level of QoS on a wireless cell but not guaranteed service. This is due in part to the fact that the protocols used for wireless access are contention based(CSMA/CA) and the timing standard used in access points is Distributed Coordinated Function(DCF). There are several ***"tweakable"*** methods to provide a way to prioritize voice traffic on the wireless cell to get transmitted over the other cell users.

- Adjusting the Interframe spaces when certain frames are to be transmitted(like Voice)
- Adjusting the cWinMin and cWinMax for smaller values for frames like Voice.
- Utilizing the Order bit of the Frame Control portion of the 802.11 header to give the packet a strict priority over others when transmitting. However, using this bit disables power save mode.
- See if PCF mode is allowed on the Access points for Voice users.
- Research into 802.11e Hybrid Coordination Function's(HCF) two approaches: Enhanced Distributed Channel Access (EDCA) or Hybrid Controlled Channel Access(HCCA)

Also, packets leaving the wireless distribution system services and integrated LAN interfaces can be classified, marked, policed and queued at the access-point's LAN switch port. So, as packets leave the wireless cell, they can be marked for a certain QoS and handled accordingly through the NYC Utility enterprise but once sent back onto the wireless cell it is up to CSMA/CA and DCF to schedule packets if none of the previously mentioned 802.11 QoS mechanisms are in place.

## *15.0 Next Steps*

- **Review this paper with appropriate staff and management**
- **Conduct any additional lab testing for pre-deployment answers**
- **Conduct any staff training courses**
- **Complete any matrixes and fulfill any recommendation tasks**
- **Select model and deployment approach**
- **Plan for deployment**

# *Appendix A. Application Inclusion Policy*

**New application scenario**

Business unit has a new or an update of a critical business application and wants to have it deployed on the network.

1. Business unit should follow the already outlined procedures for requesting deployment of the application if any.
2. Business unit must contact either the Network Suppport staff directly to inform them that this application needs to run over the network.
3. Network Support Staff will conduct an Application Impact Analysis against the application to determine the impact to the network and other applications if deployed and used over the network. **Note:** *NYC Utility already has a set of application impact analysis procedures which were drafted in 2001 by Applied Methodologies. Inc.*
   The analysis will also help determine what class in the QoS model the application should or must fall under. The impact analysis should provide the needed information in terms of bandwidth, delay and flow information for Network Support staff to determine the appropriate class level.
4. If the application requires a higher class of service and the impact analysis shows that it cannot fit into its appropriate class then Network Support staff must document and present the reasons why this application cannot receive the higher class of service that the business unit is requesting it should have.
5. If the application passes thought the Impact Analysis without any issues and its flows/bandwidth and delay requirements can be met with the class requested then Network Support staff shall make the appropriate changes to the QoS model(if any) and add the application.
6. A test of the application under the QoS model will be conducted to ensure that the QoS model is working properly with the new application and that the new application is operating in the correct manner.
7. Network Support staff will update its QoS application matrix to list the new application under the appropriate class with others in the same class.

**Application Promotion scenario**

Some applications may grow or change in functionality and importance over time. For example a SQL database application that was once just used for a small business function and was subject to the Best Effort class of service has now been changed in functionality and need to the business that it now needs to be moved into the Transactional or Mission-Critical classes of the QoS model. A procedure needs to be in place to promote the application upwards in class and ensure that its transactions meet the completion times required of the business units using it.

The promotion scenario is similar to the new application with the following changes.

1. Business unit should follow the already outlined procedures for requesting a change of the application if any. ***Note:*** *this need to upgrade in class could come from empirical results by the application already being changed and its performance is suboptimal over the network.*

2. Business unit must contact either the Network Support staff directly to inform them that this application needs to be upgraded in class.

3. Network Support staff will conduct an Application Impact Analysis against the application to determine the impact to the network and other applications if deployed and used over the network. ***Note:*** *NYC Utility already has a set of application impact analysis procedures which were drafted in 2001 by Applied Methodologies. Inc.*

4. The analysis will determine if the application can fit into its requested class in the QoS model. The impact analysis should provide the needed information in terms of bandwidth, delay and flow information for Network Support staff to determine the appropriate class level.

5. If the application for promotion to a higher class of service shows that it cannot fit into its appropriate class then Network Support staff will document and present the reasons why this application cannot be promoted to a higher class of service that the business unit is requesting it should have or it needs.

6. If the application passes thought the Impact Analysis without any issues and its flows/bandwidth and delay requirements can be met with the class promotion requested then Network Support staff shall make the appropriate changes to the QoS model(if any) and promote the application.
7. A test of the application under the QoS model will be conducted to ensure that the QoS model is working properly with the application in its new class and that the new application is operating in the correct manner.
8. Network Support staff will update its QoS application matrix to list the new class that the application now falls under.

**Application Demotion scenario**

The demotion scenario is similar to the new application with the following changes.

1. Business unit should follow the already outlined procedures for requesting a change of the application if any. ***Note:*** *this need to demote in class could come from empirical results by the application already being changed and its performance is impacting other applications in its current class.*

2. Network Support staff or the business unit must contact either each other or the Network Support staff directly to inform them that this application needs to be demoted in class. The Network Support staff can perform this if the application is impacting other applications over the network.

3. Network Support staff will conduct an Application Impact Analysis against the application to determine the impact to the network and other applications if deployed and used over the network. ***Note:*** *NYC Utility already has a set of application impact analysis procedures which were drafted in 2001 by Applied Methodologies. Inc.*

4. The analysis will determine why the application is affecting other applications in its class in the QoS model. The impact analysis should provide the needed information in terms of bandwidth, delay and flow information for Network Support staff to demote the application to the appropriate class level.

5. If the application for demotion to a lower class of service shows that it cannot fit into its appropriate class then Network Support staff must document and present the reasons why this application cannot be demoted to its chosen class of service that the would be optimal for its continued use.

6. After the impact analysis is conducted and it shows that the application's flows/bandwidth and delay requirements can be met within a demoted class then Network Support staff shall make the appropriate changes to the QoS model(if any) and promote the application.

7. A test of the application under the QoS model will be conducted to ensure that the QoS model is working properly with the application in its new(demoted) class and that the demoted application is operating in the correct manner.

8. Network Support staff will update its QoS application matrix to list the new class that the application now falls under.

# *Appendix B. Bibliography*

**RFC**

**3261 The Session Initiation Protocol (SIP)**
**2474 Definition of Differentiated Services Field in the IPv4 and IPv6 headers**
**2475 An Architecture for Differentiated Service**
**2481 A Proposal to add Explicit Congestion Notification (ECN) to IP**
**2597 Assured Forwarding PHB Group**
**2598 An Expedited Forwarding PHB**
**3175 Aggregation of RSVP for IPv4 and IPv6 Reservations**
**3246 An Expedited Forwarding PHB**

**End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs**
Cisco Press, Published: Nov 9, 2004; Copyright 2005; Pages: 768; Edition: 1st.

**Cisco Catalyst® QoS: Quality of Service in Campus Networks**
Cisco Press, Published: Jun 6, 2003; Copyright 2003; Pages: 432; Edition: 1st.

**Cisco QOS Exam Certification Guide (IP Telephony Self-Study), 2nd Edition**
Cisco Press, Published: Nov 18, 2004; Copyright 2005; Pages: 768; Edition: 2nd

**Inside Cisco IOS Software Architecture (CCIE Professional Development)**
Cisco Press, Published: Jul 28, 2000; Copyright 2000; Pages: 240; Edition: 1st.

**Packet Magazine archives**

http://www.cisco.com/en/US/about/ac123/ac114/about_cisco_packet_magazine.html

**Internet Protocol Journal archives**

http://www.cisco.com/en/US/about/ac123/ac147/about_cisco_the_internet_protocol_journal.html

**Network Magazine archives**

http://www.networkmagazine.com/pastIssues.jhtml;jsessionid=TCHMT4CC5JHSCQSNDBCCKH0CJUMEKJVN?year=2005

Applied Methodologies, Inc, Lab resources

WWW.AMILABS.COM

# *Appendix C. General Findings and Research Notes*

**Platform specific notes from initial QoS testing activities and QoS Toolset compilation**
These are the original notes from testing and have not been formatted/proofed.

**2600 router testing notes**

differences in versions from 12.2 to 12.3 for bandwidth percent feature in policy map. recommend NYC Utility go to 12.3 on all 2600s. Most 2600xms can do this with 64mbs and 16mbs flash min running enterprise basic

As for the 2600 non XMs can run latest 12.2 but policy map commands for priority and voice allocation will need to be specified in bps.

To verify if IOS is auto tuning tx ring for serialization delay issue "sh controllers s0/0 | include tx_limited" should see **"tx_limited=1(2)"**

WHY!!!! does the class-default show up as 0% bw??? cannot allocate 25% bw to the class default as in the research text? On 12.3 can use ftp-server option to load all files at once and at will change them on 12.2T still trying to get ftp to run.

**3800 router testing notes**

When applying a policy to the interface the interface resets.

**3550 Switch testing notes**

Trusting on server ports will not ensure return trip marking must add policy Bandwidth percent is for CBWF queuing only and for output which 3550 does not support. All classification and markings work perfectly and so does the show commands tested on 12.1(22)EA1

Able to add and remove service policies while call is running without interruption to ports or call. **The dscp markings changed on the fly RA!!**

**3750 Switch testing notes**

Trusting on server ports will not ensure return trip marking must add policy Bandwidth percent is for CBWF queuing only and for output which 3550 does not support. All classification and markings work perfectly and so does the show commands when running QoS commands on interface for first time interface does cycle down and up there is a difference in the way versions portray the dscp markings in the policy map.

**The same show commands have different output between 3550 and 3750**

**sh mls qos int stat --- the output on the 3750 Its very poor!**

sh mls qos interface buff

span ports don't work the same on 3750 as on 3550. Having trouble seeing both sides of a conversation's packets marked. Individual analyzers on the ws show the packets are marked though.

It appears that the sh mls qos interface statistics is misleading on the input counters. It does not show the packets dscp incoming correctly where traces on the actual end workstation show the received packet marked.

the output stats of the command is fine

mls qos map cos-dscp 0 8 16 24 32 46 48 56
mls qos map ip-prec-dscp 0 8 16 24 32 46 48 56

Verison 12.1(19)EA1d issue???
receive funky output of following command
QOS3750#1#sh policy-map interface
FastEthernet1/0/1
service-policy input: NYC UTILITYBASEQOS3750
class-map: REALTIME (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
match: access-group name REAL-TIMEqm_inform_features_ps_action: CLASS_SHOW

Unfortunately you are hitting a known issue with this platform. The show policy-map interface command is not supported on this platform. This is due to hardware implementation. The reason the command is still in the IOS is from when this code was ported over from a pervious version of code that runs on routers that use different hardware.

The limitation is described in this bug:
**CSCdy50035**
Externally found cosmetic defect: Closed (C)
show policy interface command doesn't show statistics

Therefore the workaround, as you have already determined is to use the show mls qos interface stats command.

Let me know if I can provide any more assistance regarding you service request.

AMIQOSSW1(config)#no policy-map dirty

Unexpected exception to CPUvector 1100, PC = 7E56CC
-Traceback= 7E56CC 7EBEF8 7E87D8 264428 26E0E0 26E1AC 180CC4 193C14 260658 25AB04

Buffered messages:
00:00:43: %STACKMGR-6-SWITCH_ADDED: Switch 1 has been ADDED to the stack
00:00:50: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:50: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type vlan
00:00:53: %SYS-5-CONFIG_I: Configured from memory by console
00:00:53: %STACKMGR-6-SWITCH_READY: Switch 1 is READY
00:00:53: %STACKMGR-6-STACK_LINK_CHANGE: Stack Port 1 Switch 1 has changed to state DOWN
00:00:53: %STACKMGR-6-STACK_LINK_CHANGE: Stack Port 2 Switch 1 has changed to state DOWN
00:00:54: %STACKMGR-6-MASTER_READY: Master Switch 1 is READY
00:00:54: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) C3750 Software (C3750-I5-M), Version 12.2(20)SE4, RELEASE SOFTWARE (fc1)Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Sun 09-Jan-05 00:09 by antonino
00:00:55: %LINK-3-UPDOWN: Interface FastEthernet1/0/1, changed state to up
00:00:55: %LINK-3-UPDOWN: Interface FastEthernet1/0/24, changed state to up
00:00:56: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0/1, changed state to up

## Catalyst 6500 Switch MSFC and Supervisor 720 in native mode testing notes

### WS-6748-GE

**Note:** This module is supported only on the Cisco Catalyst 6500 Series Supervisor Engine 720 running Cisco Catalyst OS Version 8.1.2 or Cisco IOS® Software Release 12.2(17a)SX (available in the next few months) and future software releases.

WS-X6724-SFP for Redundant Uplinks to Distribution Layer, GE Aggregation, Core and Data Center Deployments

**Note:** This module is supported only on the Cisco Catalyst 6500 Series Supervisor Engine 720 using Cisco Catalyst OS Version 8.1.2 or Cisco IOS Software Release 12.2(17a)SX (available in the next few months) or subsequent software releases.

**6416-gbics work in msfc2 in native mode only**

3750 type configs do work on the 6500 mfsc native mode. But some bit rates may have to be changed for the policer plus the interfaces do not have a "priority-queue out" function like the 3750 does. The priority queue is specified in the policy map on this switch so it is better to use the router IOS type of configs on the msfc instead of the 3750 IOS types. However, for MSFC with server farms the 3750 server farm classification and marking commands and access-list can work fine on the 6500 msfc. Plus you get the NBAR for easier classification of server applications. **NOT TRUE FOR SWITCH PORTS**

Uplinks in native mode work fine can add trust dscp to uplink ports
However when adding nbar on uplink ports the switch will revert to software switching
QM-6-NBAR_ENABLED: Packets will be software switched.

May limit NBAR to just router ports for performance reasons.
Same for sup720 in native mode

**Note:when ports are in switchport mode NBAR is not available**

Ingress LAN Port Trust States
The trust state of an ingress LAN port determines how the port marks, schedules, and classifies received Layer 2 frames, and whether or not congestion avoidance is implemented. You can configure the trust state of each ingress LAN port as follows:

•Untrusted (default)
•Trust IP precedence (not supported on 1q4t LAN ports except Gigabit Ethernet)
•Trust DSCP (not supported on 1q4t LAN ports except Gigabit Ethernet)
•Trust CoS (not supported on 1q4t LAN ports except Gigabit Ethernet)

**sh queueing interface gi1/1**

**mls qos queueing-only**
**Disables marking and policing globally**
**Configures all ports to trust Layer 2 CoS not to be used**

**Line Card queue notes**

When setting queue buffer allocations on one interface it adds to all interfaces on line card
WS-X6416-GBIC
QOS6505#2(config-if)#wrr-queue queue-limit 40 30
queue-limit configured on: Gi2/1 Gi2/2 Gi2/3 Gi2/4 Gi2/5 Gi2/6 Gi2/7 Gi2/8

May simplify commands but may not do the same on other line cards thus having to specify queue limits on each port or range of ports. Some cards may apply to first 10 ports then you have to apply for the next 10 this is because on some line cards the buffers are shared amongst x numbers of ports on the line card.

Removing qos automatically removes the advanced wrr queuing configs on the supervisor ports BUT leaves any configurations left on the line cards. Must then remove line card ports separately with "removeqos" command.

**Adding and removing a policy from a routed interface does not reset the interface.**

Adding and removing mls qos trust dscp command does not affect interface.
Removing mls qos trust dscp from routed interface does affect end-to-end marking so routed interfaces must have mls qos trust set on them.

Can add nbar to vlan interface but still get message of software switching
Routed interface policy maps work on vlan interface
Aggregate policer works on vlan interface.

qos on vlan interfaces just for intervlan policing
but not really needed according to Cisco

55k uplink configurations work fine. Must emphasize that mls trusting be **turned off** for 55k uplink interface to properly use service policy to mark packets at that point, otherwise the trusting will inadvertently trust non marked packets to begin with.

**QoS applied to Trunk ports**
**Trunk testing between two 6500 switches**

- Setting MSFC and 720 links to trunk
- Both end 3750 switches on either end of each trunked 6500 switch has QoS enabled
- MSFC switch is set for QOS and its trunk interface has QoS trusting enabled
- The 720 switch does not have QoS enabled.

First test of pings shows that the DSCP markings stay intact from end-to-end through the trunk even though one of the 6500s in the middle(720) does not have QoS enabled at all.
Removing mls QoS trusting on the QoS enabled switch trunk port then shows in a ping test that the markings do not survive across the non QoS trunked switch.

What this shows is that for QoS deployments a switch in between cannot be QoS enabled but as long as one of its links between QoS enabled end switches is connected to a QoS enabled switch that has its link set to trust the DSCP markings shall survive.

Reversing the configurations – having the msfc qos disabled and 720 qos enabled with trust on its trunk port results in the same behavior as above. The markings survive when crossing a qos black hole. Turing off trusting on the 720 trunk port to the non qos switch shows that the dscp markings do not survive.

Enabling qos on both switches but having qos trusting off on both sides of trunk links
Only one side of the packets get marked. Half duplex marking. It depends on which side of the end-to-end conversation you place is where you will see the direction of the marking.

Enabling qos on both switches but having qos trusting on on just one side of a trunk link.
Only one side of the packets get marked. Half duplex marking. It depends on which side of the end-to-end conversation you place is where you will see the direction of the marking.

Enabling qos on both switches and having qos trusting turned on both sides of a trunk link.
Provides the desired results of full duplex dscp marking. The markings survive in both directions.

End result here is that trusting must be enabled on both ends of a trunk link

Adding and removing policies on 720 interfaces did not result in any interface resets.
Turning on trusting and queuing features on and off also did not result in any interfaces resetting.

Running IP nabar on the vlan interface does not work. The command sticks but no statistics are shown. This is the opposite of the MSFC in native mode where nbar works on the vlan interface.
The vlan interfaces are configured identical for cef and route caching. Maybe an IOS issoue.

**Port channel testing results for both MSFC and SUP720**

When setting up a port channel running trunking between two switches dscp trusting commands can only be applied to the port channel and not the member interfaces.

mls qos trust command can be entered only on the Port-channel and not on it's members

Having no trusting on either side of the switch's port channel interface only provides half duplex marking.

Having trusting of dscp added to one side's port channel interface provides only half duplex marking.

Adding trusting to both sides of the port channel provides the desired results of keeping the markings intact across the portchannel and trunk.

Add trusting to port channel interface automatically adds it to channel member interfaces

Removing trusting from a port channel automatically removes the trust commands from the group members.

Queuing commands cannot be applied to the port channel they still must be applied to the individual portchannel member interfaces.

Adding and removing queuing on the port channel member interfaces does not affect operation, no reset

Adding and removing queuing on the port channel member interfaces does not affect port channel operation.

Random detect - WRED can be applied to the Portchannel in conjunction with the queuing on the member ports to offer true class based service scheduling of packets out of the interface.
**mls qos channel-consistency** – brings down port channel interface, not used in NYC Utility's case

Random-Detect for distributed WRED can be applied to the Port channel interfaces and any/all Catalyst switch ports for additional congestion avoidance capabilities.

Hey AMILABS,

**NBAR is not supported on the SUP720 on LAN interfaces**. Check out the release note here:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/ol_4164.htm#wp25624

?Network-based application recognition (NBAR) for LAN interfaces

Please let me know how you would like to proceed. Thanks!

Regards,

Cisco support

# *Appendix D. Spatial Reuse Protocol 802.17 RPR Notes*

Information regarding the tuning of SRP on 10720 DPT platforms for QoS.

**set srp-priority**

To set the priority level of Spatial Reuse Protocol (SRP) Layer 2 packets, use the MQC **set srp-priority** command. To remove the **set srp-priority,** use the **no** form of the command.

**set srp-priority** *value*

no set srp-priority *value*

**Syntax Description**

| | |
|---|---|
| *Value* | Specifies the priority assigned to SRP packets, where a higher number represents a higher priority. The valid values are from 0 to 7. |

**Defaults**

The default value is 0.

**Command Modes**

Modular QoS class-map configuration

**Command History**

| Release | Modification |
|---|---|
| 12.0(18)ST | This command was introduced. |
| 12.0(22)S | This command was integrated into Cisco IOS Release 12.0(22)S. |

**Usage Guidelines**

For the Cisco 10720 Internet Router, data packets that are mapped with SRP priority 6 and 7 are, by default, mapped to the SRP high-priority queue. Any data packet that is not mapped to the SRP high-priority queue has its data packets set to SRP priority 0 and is mapped to the SRP low-priority queue.

To change the priority value between the high- and low-priority queue, use the **set srp-priority** command with the **srp priority-map transmi**t command so that the high-priority queue uses a different slicing value. The **set srp-priority** command applies only to outbound traffic exiting the SRP interface.

For example, to change the mapping so that data packets with SRP priority 5 go to the SRP high-priority queue, use the **set srp-priority 5** command to set the SRP priority bits to 5. Then use the **srp priority-map transmit 5** interface command to change the mapping of SRP queues so that queues 0 to 4 are mapped to the SRP low-priority queue, and queues 5 to 7 are mapped to the SRP high-priority queue.

**Examples**

The following example sets the SRP priority to 5 and the IP high priority to SRP high priority:

```
Router# configure terminal
Router(config)# class-map match-any high_priority
Router(config-cmap)# match ip precedence 5
Router(config-cmap)# match ip precedence 6
Router(config-cmap)# match ip precedence 7
Router(config-cmap)# exit
Router(config)# policy-map IP_to_SRP
Router(config-pmap)# class high_priority
Router(config-pmap-c)# bandwidth percent 80
Router(config-pmap-c)# set srp-priority 5
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface srp 1/1
Router(config-if)# service-policy output IP_to_SRP
Router(config-if)# srp priority-map transmit 5

Router(config-if)# end
```

**Related Commands**

| Command | Description |
|---|---|
| **bandwidth** | Specifies or modifies the fair-queue committed bandwidth allocated for a traffic class that belongs to a service policy. |
| **Class** | Specifies the name of the class whose policy you want to create or change, and the default class (commonly known as the class-default class) before you configure its policy. |
| **Class-map** | Creates a class map to be used for matching packets to a specified class. |
| **interface srp** *port/slot* | Selects the SRP interface. |
| **match ip precedence** | Identifies a specific IP precedence value as the match criterion. |
| **policy-map** | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| **service-policy** | Attaches a policy map to an input or output interface to be used as the service policy for that interface. |
| **srp priority-map** | Sets priority mapping for transmitting packets. |

Hi AMILABS

Sorry to ge back to you, but its been crazy :)

Here is a link that hopefully helps you out:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newf
t/120limit/120s/120s24/10720.htm#wp1563488

In the above, for DSCP points you can match on them just the same as
you would IP Precedence values. Let me know if this does it for you.
Otherwise, feel free to let me know if you have any remaining
questions, comments or if this can be closed.

Best regards,
Cisco Support

AMILABS wrote:
> Yes this does help very much. Is there a link for QoS for the 107200
and srp specifically? In the policy map config I need to know what type
of set command(if there is a special one for srp) to ensure that EF or
dscp 46 traffic gets cos 5 or above for srp transmit priority queue
usage?
>
> I beleive that is all I will need and I am good to go.
>
> Regards..

AMILABS,

One other thing I found that may be even more helpful:

- The DEFAULT behavior of the c10720 QoS treatment is to set the
  SRP COS value to 0 for all traffic. This subsequently means that
  the c10720 will only forward low-priority traffic in default mode.
  The exception to this behavior are routing protocols (pak priority
  in IOS terms) and L2 protocols (IPS and topology packets).
  Please note that this default behavior is independent of layer 3 QoS
  values and in default mode the layer 3 QoS values are not touched.
- The c10720 offers via MQC the knob to explicitly assign a distinct
SRP COS
  value to a given output queue config via the MQC bandwidth statement.
  If the configured COS value is above the SRP discriminator (default =
5)
  the traffic goes out a high otherwise it is low.
- If traffic is assigned via MQC to a "priority" queue the COS value
will
  be 7 and will go out as high priority.

Regards,
Cisco support

le wrote:

> AMILABS,

> On c10720 there is no automatic mapping from IP precedence to srp-priority. On the GSR it's different, but on c10720 you need a policy-map. PAK_Priority packets will be sent with
> srp-priority = 6 which is the high transmit queue (6-7 by default). The DSCP just like IP Precedence will also need to be specified in a policy-map.
> Let me know if this help answer your questions or not.
> Best regards,
> Cisco support
>
Hi AMILABS,

On c10720 there is no automatic mapping from IP precedence to srp-priority. On the GSR it's different, but on c10720 you need a policy-map. PAK_Priority packets will be sent with
srp-priority = 6 which is the high transmit queue (6-7 by default). The DSCP just like IP Precedence will also need to be specified in a policy-map.

Let me know if this help answer your questions or not.

> "AMILABS wrote:
> > Thanks Cisco support, the set commands are helpful and may be utilized. But I just need to know the default behavior. How does a packet that is using a DSCP value end up in the priority queue or it does not because the router is only looking for the IP precedence or class selector? What is the default behavior regarding dscp so I can determine if I even have to use the use the set priority commands?
> >
> > See the table below this is a default standards based table
> > DSCP Range numbers to IP Precedence or COS Map Reference
> > Range of DSCP values Compatibility with these
> > DSCP
> > Decimal IP Precedence values
> >
> > 0-7 0 Routine
> > 8-15 1 Priority
> > 16-23 2 Immediate
> > 24-31 3 Flash
> > 32-39 4 Flash Override
> > 40-47 5 Critical
> > 48-55 6 Internetwork Control
> > 56-63 7 Network control
> >
> >
> > if a packet comes in with a dscp of 46(for EF for voice)= IP precedence 5, does the router (SRP)automatically know to assign it to the priority queue by default or do I have to set this and maybe a MQC mapping table on the router? Is the default functionality just IP Precedence and if so what IP precedence values are the ones that end up in the SRP priority queue.
> > I just need to make sure and clarify..
> >
> > Thanks...

## Testing notes of the 107200 DPT routers for QoS features

# AMILABS July 30th 2005

Loading and unloading QoS policy maps does not reset interfaces
QoS Menu works
QoS CLI works
Policer is in bit rates
Loading and unloading of policers work fine
Needed separate policy map for gigabit interfaces versus srp interface to accommodate spr priority transmit command.
Show output of policy maps works

As of version 12.0(31)S, still no NBAR

Bandwidth percentage for policer is available in version 12.0(28) and above.
Support for Classification of Locally Sourced Packets – HW dependant.
Note above is not true tested with 12/0.(31)S still not percentage based policer for use in class maps.

Support for locally defined SNMP dscp setting  snmp-server ip dscp

Having problems with loading priority percentages for voice and video at same time. Works fine on regular router platforms so it is either a 10700 platform issue or IOS version issue.

Seems to work with just one priority percent per policy map and the video one just has bandwidth percent allocated. However I can assign a separate srp priority statement to different classes in the policy map.

http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a008009d848.html

http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a008009d848.html#wp1417317

DSCP-based WRED and IP Precedence-based WRED are not supported in the same policy map.

•DSCP-based WRED does not support IPv6 traffic.

WRED will use defaults and can be tuned in each mode’s main file policy maps.
default WRED min and max thresholds can be tuned
example  random-detect dscp 1 3000 3500 1

Special considerations towards bandwidth allocations and policer allocations when dealing with side A and side B RPR (VCCI)links to ensure that the allocations are consistent per side and not over or under allocated to reflect the aggregate of both sides.

The VCCI, or Virtual Circuit Connection
Identifier, is a variable that identifies a

virtual circuit connection between two nodes.
A virtual circuit connection, or VCC, consists of
one virtual circuit link or a series of concatenated
virtual circuit links. In its most common usage,
the value of the VCCI is unique between the nodes
at the extremities of the virtual circuit
connection, but not on a network-wide basis. Hence,
its value needs to be qualified by the ATM
addresses of these end nodes. At one of these
end nodes, its value needs to be qualified by the
ATM address of the far-end node. Some applications
can extend this definition to make the VCCI value
unique on a network-wide basis. This is specially
possible when VCCIs are administered from a management system and not locally assigned by a node.

Currently baked DPT router configs for REMANING BANDWIDTH type of configuration with the priority queue allocating all bandwidth when needed.
Will test for functionality when second DPT router is available so packet flows, matching and allocations can be verified.

Will also test the use of just bandwidth percent and not priority queue on the DPT routers to see which version works best when second router arrives.

MQC Strict Priority Queue on the Cisco 10720 Internet Router
Starting in Cisco IOS Release 12.0(26)S, the Cisco 10720 router does not support the priority percent percentage and priority bandwidth-kbps commands in policy-map class configuration mode, except as a hidden command to preserve backward compatibility. These commands allow you to create a queue to handle low-latency traffic.

The percentage and bandwidth-kbps arguments specify the bandwidth that is guaranteed to the queue. However, when the excess bandwidth on a link is consumed by other queues, the priority queue receives only the amount of bandwidth that you configured. As a result, if there is a burst of traffic into a priority queue, the packets in the burst are shaped and transmitted at the configured rate. Because a temporary burst often exceeds the amount of configured bandwidth, the packets in the burst are queued with an increased latency. This impacts performance because the increased latency occurs for bursts that have a long-term traffic rate that is much lower than the configured bandwidth.

To guarantee latency for any packet that enters the priority queue regardless of the current congestion level in the link, strict priority mode is supported as the only mode of operation for a priority queue in Cisco IOS Release 12.0(26)S and later releases.

To configure the strict priority queue, you use the priority command in policy-map class configuration mode with no arguments; the percentage and bandwidth-kbps arguments are not supported. When you create a strict priority queue using the priority command, the committed information rate (CIR) of the queue is set to 99 percent of the link bandwidth.

Because the priority queue consumes almost all the link bandwidth when packets are transmitted from it, there is no way to guarantee bandwidth to other queues on the link. Therefore, if you use the priority command for a traffic class in a policy map configuration, only bandwidth remaining commands are allowed in other class configurations in the same policy map.

To prevent the priority queue from starving other queues on the link, you can use the police command in conjunction with the priority command. In this case, the CIR is set to the bandwidth specified in the police command. Also, you can use the bandwidth command on the other queues in the link to create one or more queues with guaranteed bandwidth. In either case, you must set the exceed and violate actions of the police command to drop for the CIR of the priority queue to be affected.

**Random Detect DSCP Defaults**

In a class map, the default minimum and maximum threshold values and the mark probability denominator for a DSCP value not configured with the random-detect dscp dscp value command are set by the values configured with the random-detect dscp default command.

If you do not enter the random-detect dscp default command in a class map (or if the default profile has been removed by entering the no random-detect dscp default command), the default minimum and maximum threshold values and mark probability denominator for a DSCP value are as follows:

•The default minimum and maximum packet threshold values are automatically determined by queue size.

•The default mark probability denominator is 10.

Testing conducted on 12.0(27) most of NYC Utility’s DPT routers are 12.0(26) only  a couple are 12.0(29).

**Bandwidth remaining percent**

Use the MQC bandwidth remaining percent command to specify how the "remaining" bandwidth is distributed among the output queues on a Cisco 10720 router interface or subinterface. "Remaining" bandwidth is the available bandwidth left on an interface or subinterface after all guaranteed traffic is accounted for.

The bandwidth remaining percent command allows you to configure the remaining bandwidth for output queues. The percentage parameter specified with the bandwidth remaining percent command is translated into an internal excess information rate (EIR) value between 0 and 255. The aggregate of all user-configured EIR bandwidth percentages cannot exceed 100 percent.

If the aggregate of all remaining bandwidth is less than 100 percent, the remainder is evenly split among user queues (including the default queue) that do not have a remaining bandwidth percentage configured. The minimum EIR value of each output queue is 1.

The EIR parameter for the network control queue is fixed at 128 and is not configurable.

If you have not configured a committed information rate (CIR) value for the default queue and it is the only user queue, the default queue receives half of the remaining bandwidth percentage of the network control queue.

CSCeb67098
Symptoms: A memory leak may occur in the Parallel Express Forwarding (PXF) interprocess communications (IPC) buffer on a Cisco 10720, as may be seen in the "toaster IPC buffer" counter in the output of the show buffers EXEC command.
When the buffer pool is empty, the following error messages may appear, you may no longer be able to Telnet to the router, and the router may reload unexpectedly:
%CAMR_QUEUE_CFG_GENERAL-3-EREVENT: Error @
../toaster/camr_rp/camr_tt_queue_cfg.c:463 -Traceback= 500DB204
500DB2BC 503954D8 503986EC 50330A58
%SYS-2-MALLOCFAIL: Memory allocation of 18196 bytes failed from
0x502C5BD0, alignment 32 Pool: I/O Free: 552 Cause: Not enough free
memory Alternate Pool: None Free: 0 Cause: No Alternate pool
-Process= "Pool Manager", ipl= 0, pid= 5 -Traceback= 50308EEC 5030A8E8
502C5BD8 5031DD3C 5031DE7C
Conditions: This symptom is observed on a Cisco 10720 when a policy map with a Weighted Random Early Detection (WRED) configuration that is enabled by
using the random-detect policy-map class configuration command is applied to any interface of the router.
The higher the rate with which the Route Processor (RP) sends packets to PXF, the faster the PXF IPC buffer leaks. However, the buffer may leak very slowly, and it may takes weeks before the buffer pool is empty.
Workaround: Remove the policy maps with the WRED configuration from all interfaces of the router.

Running H323 protocols and ports for voice and signaling

dpt testing notes on SRP fail over for voice Qos

Enabling and disabling qos on DPT SRP interface does not reset interface

on dpt rtr 1 running version 12.0(31)S the sh policy-map int srp 1/1 does show the mac layer SRP dscp to srp priority matching and handling of voice bearer and signaling frames working properly
gi2/1 does show voice signaling packets but no voice packets matched

on dpt rtr 2 running version Version 12.0(27)S5 the sh policy-map int srp 1/1 does not show any qos statistics
regardless of direction of call originated flow, no voice bearer
But signaling packets are seen and matched

gi2/1 does show voice packets matched and signaling packets matched

The above issue was from a higher port number used on the softphone than configured on the 3750's marking access-list. Traces verified this. Adjusted access-lists and re-ran marking tests over the DPT ring all works perfectly BOTH SRP interfaces matched packets for SRP priority for Voice and signaling traffic in both directions.
Both Gigabit interfaces on both dpt routers also matched the packets correctly in both directions.

Both routers class-default policy maps work properly

The three color policer works fine on Gigabit interfaces and also on the SRP interfaces. Interesting side not for SRP interfaces and policer. The policer will work on just one VCCI side see output below. Curious to see how this is affected in a wrapped ring situation.

```
DPT Router#1
police (VCCI 2->side B):
        1000000000 bps, 1300000000 limit, 1300000000 extended limit
        conformed 0 packets, 0 bytes; rate 0 bps; action: set-dscp-
transmit 8
        exceeded 0 packets, 0 bytes; rate 0 bps; action: drop
        violated 0 packets, 0 bytes; rate 0 bps; action: drop
      police (VCCI 3->side A):
        1000000000 bps, 1300000000 limit, 1300000000 extended limit
        conformed 206 packets, 19158 bytes; rate 0 bps; action: set-
dscp-transmit 8
        exceeded 0 packets, 0 bytes; rate 0 bps; action: drop
        violated 0 packets, 0 bytes; rate 0 bps; action: drop


DPT Router#2

police (VCCI 2->side B):
        1000000000 bps, 1300000000 limit, 1300000000 extended limit
        conformed 0 packets, 0 bytes; rate 0 bps; action: set-dscp-
transmit 8
        exceeded 0 packets, 0 bytes; rate 0 bps; action: drop
        violated 0 packets, 0 bytes; rate 0 bps; action: drop
      police (VCCI 3->side A):
        1000000000 bps, 1300000000 limit, 1300000000 extended limit
        conformed 240 packets, 24003 bytes; rate 2000 bps; action: set-
dscp-transmit 8
        exceeded 0 packets, 0 bytes; rate 0 bps; action: drop
        violated 0 packets, 0 bytes; rate 0 bps; action: drop
```

In a wrapped situation the router where the failure occurred the policer will show that the three colors conformed etc packets have moved to the other VCCI. The far end router's policer will show packets still on the original VCCI.

**Failure testing notes**
**Goal:** try to see if a ring wrap will impact a call, cause jitter, break a call or cause longer delays. Even though this is only a two node ring, the production ring is longer and has longer fiber distances. A wrap may impact call quality for users communicating on the ends of the wrapped DPT routers.

**Testing of failures with no general traffic generated, just one plus voice calls.**

Show if SRP packet counters does show that the QoS voice tagged packets are given high priority SRP mac layer priority treatment and this fsm is working properly.

Failure of side A interface
No change in in voice call no dropped voice packets
Restore of interface caused no issues with voice call

Failure of side B interface
No change in in voice call no dropped voice packets
Restore of interface caused no issues with voice call

Interface policy maps still work if one side is down/wrapped state
No voice call interruption noted or lost packets when switching from idle to wrapped back to idle states

**Failure of side A completely srp interface goes down**
Remove fiber from both dtp routers side A
Call fails goes dark cannot hear other side packets no
Restore one dpt side A call restores without hanging up as soon as SRP interface goes back up.
Ring in wrapped state
Bring up other dpt side A no change in voice call

**Failure of side B completely srp interface goes down**
Remove fiber from both dtp routers side B
Call fails goes dark cannot hear other side packets
Restore one dpt side B call restores without hanging up as soon as SRP interface goes back up.
Ring in wrapped state
Bring up other dpt side B no change in voice call

**Failure of a single TX side A**
No change in voice call
Failure of TX on both dtp routers side A
interface goes down no voice
Recover by inserting cable on on Side A TX brings back voice call but in wrapped state
Wrap clear process does not affect voice call

**Failure of a single RX on side A**

No change in voice call
Failure of RX both dpt routers side A
interface goes down no voice
Recover by inserting cable on on Side a RX brings back voice call but in wrapped state
Wrap clear process does not affect voice call

**Failure of a single TX side B**
NO change in voice call
Failure of TX on both dpt routers side B
Interface goes down no voice
Recover by inserting cable on on Side B TX brings back voice call but in wrapped state
Wrap clear process does not affect voice call

**Failure of a single RX side B**
NO change in voice call
Failure of RX on boh dpt routers side B
Interface goes down no voice
Recover by inserting cable on on Side B RX brings back voice call but in wrapped state
Wrap clear process does not affect voice call

Recovery back to idle state does not cause an voice issues

**Dual fiber failure test**
**Fiber cut test**
**Signal degrade test**

WTR timer resumption observations
No change in voice call from when a ring is wrapped to an unwrapped state.
Change WTR to 10 seconds from 60 so interface can go back to idle state much faster
no difference to call

**IPS manual switch test**
Same as forced switched but a lower priority

**IPS forced test**
Forced ips request to switch side a on DPT router 1 no changed in voice call
Forced ips request to switch side a and b causes interfaces to go down
Unforcing ips request switch b brings interface back up and call continues.
Jump to remote dpt router and force switch side b while a was switched on dpt 1 no change to voice call.
Having both dtp router forced switched side a causes interface to go down.
IPS forced switch mechanisim can be used to add new nodes and test wrapped states on a dpt ring.
The voice call and qos operated properly during these tests

**IOS option SRP SRR test**
SRR recovers quickly

Without SRR and remove cable from both DTP routers side A TX causes ring to break thus interrupting call

Having SRR enabled on both SRP interfaces provides an extra level of redundancy by having everything run on a single ring. (default SRR bandwidth allocated is 400mbs.)

The call is interrupted briefly as the srp interface comes back up on a single ring and eigerp neighbors re-esablish less than 2 seconds.

```
QoS10720DPT1#
5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 wrapped on side B (side A Span
Neighbour Signal Fail)
QoS10720DPT1#sh srp srr srp 1/1

 SRR Information for Interface SRP1/1
  SRR version: 0
  Current node Info:
  Node MAC address: 0008.a35f.8a00
  State           : Idle
  Outer In Use    : Yes
  Inner In Use    : Yes
  Announcing      : No
  Outer Fail      : IDLE
  Inner Fail      : IDLE
  SRR Prevention  : No (Idle)
  Periodic SRR packet sent every 10 sec. (next pkt. after 6 sec.)
  Single ring bandwidth set to 400 Mbps.
  SRR WTR timer set to 60 sec.
  Side A WTR countdown is inactive.
  Side B WTR countdown is inactive.
  Name                      MAC       Outer fail Inner fail Announce
Last received
  QoS10720DPT1         0008.a35f.8a00 IDLE       IDLE       No
00:00:04
                       0008.e321.3280 IDLE       SF         Yes
00:00:00

QoS10720DPT1#
5d00h: %SRP-4-ALARM: SRP1/1 Side B Failure: Layer 1(SF)
5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 wrapped on side A (side B Self
Detect Signal Fail)5d00h: %LINEPROTO-5-UPDOWN: Line protocol on
Interface SRP1/1, changed state to down
5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 unwrapped on side A (node is
locked out)
5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 unwrapped on side B (node is
locked out)
5d00h: %SRP-4-SRR_STATE_CHANGE: SRP1/1 SRR usage changed (Outer ring is
used, BW=400 Mbps)5d00h: %SRP-4-ALARM: SRP1/1 Side B Failure: MAC
Keepalive(SF)
```

```
5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 wrapped on side A (side B Self
Detect Signal Fail)5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 wrapped on
side B (side A Span Neighbour Signal Fail)
5d00h: %SRP-4-SRR_STATE_CHANGE: SRP1/1 SRR usage changed (Both rings
are used)
5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 unwrapped on side A (node is
locked out)
5d00h: %SRP-4-WRAP_STATE_CHANGE: SRP1/1 unwrapped on side B (node is
locked out)
5d00h: %SRP-4-SRR_STATE_CHANGE: SRP1/1 SRR usage changed (Outer ring is
used, BW=400 Mbps)5d00h: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor
200.1.1.2 (SRP1/1) is up: new adjacency
5d00h: %SONET-4-ALARM:  SRP1/1: SLOS Side B
QoS10720DPT1#
QoS10720DPT1#
5d00h: %LINEPROTO-5-UPDOWN: Line protocol on Interface SRP1/1, changed
state to upsh srp srr srp 1/1

 SRR Information for Interface SRP1/1
  SRR version: 0
  Current node Info:
  Node MAC address: 0008.a35f.8a00
  State           : Discovery
  Outer In Use    : Yes
  Inner In Use    : No
  Announcing      : Yes
  Outer Fail      : IDLE
  Inner Fail      : SF
  SRR Prevention  : No (Idle)
  Periodic SRR packet sent every 1 sec. (next pkt. after 1 sec.)
  Single ring bandwidth set to 400 Mbps.
  SRR WTR timer set to 60 sec.
  Side A WTR countdown is inactive.
  Side B WTR countdown is inactive.
  Name                     MAC       Outer fail Inner fail Announce
Last received
  QoS10720DPT1         0008.a35f.8a00 IDLE       SF         Yes
00:00:00
                       0008.e321.3280 IDLE       SF         Yes
00:00:00
```

IOS option SRP Fast convergence test

- WTR wrap clearing does not affect voice calls
- Calls do resume quickly when interfaces come up.
- No increase in jitter was experienced when interfaces and fiber failed and when ring was wrapped

# *Appendix E. In-depth QoS testing criterion*

The following links provide the testing criterion that NYC Utility can utilize to help understand and ensure the operational behavior of QoS tools and features prior any major deployment in the enterprise. The following testing criterion was designed to test the various QoS features across all platforms for compliance to NYC Utility's needs and to determine operational behaviors and breaking points of the features involved. The criterion is included for NYC Utility to utilize after reviewing this paper.

Main QoS Testing Criterion
http://www.amilabs.com/NYutility/maintest.htm

Qos Testing Criteria Architecture #1
http://www.amilabs.com/NYutility/test1.htm

Qos Testing Criteria Architecture #2
http://www.amilabs.com/NYutility/test2.htm

Qos Testing Criteria Architecture #3
http://www.amilabs.com/NYutility/test3.htm

Qos Testing Criteria Architecture #4
http://www.amilabs.com/NYutility/test4.htm

QoS Management and Troubleshooting Tool testing Matrix
http://www.amilabs.com/NYutility/tooltesting.htm

# *Appendix F. Catalyst 6500 Platform Line Card Port/Queue Matrix*

The following matrix outlines the Current NYC Utility Catalyst 6500 inventory of line cards and their interface queuing allocations. This information is to be used for QoS planning and provisioning purposes.

NY Utility Catalyst 6500 series switch platform inventory QoS Matrix

http://www.amilabs.com/NYutility/6500queuematrix.htm

# *Appendix G. Cisco Catalyst Platform Specific Model Matrixes*

NY Utility Basic Voice Model for all platforms QoS Matrix
http://www.amilabs.com/NYutility/basicvoicematrix.htm

NY Utility Base and Middle Model Access Layer QoS Switch Matrix
Excel sheet for your use

http://www.amilabs.com/NYutility/acclayerswmatrix.xls

NY Utility Base and Middle Model Catalyst 6500 Core/Access Layer QoS Matrix
Excel sheet for your use

http://www.amilabs.com/NYutility/coreswmatrix.xls

QoS End Device Port Matrix
http://www.amilabs.com/NYutility/enddevportmatrix.htm