CHAPTER 1

Quality of Service Design Overview

This document provides an overview of Quality of Service (QoS) tools and design and includes high-level answers to the following questions:

- Why is Quality of Service Important for Enterprise Networks?
- What is Cisco's Quality of Service Toolset?
- How is QoS Optimally Deployed within an Enterprise?
- How can QoS Tools be used to Mitigate DoS/Worm Attacks?

QoS has already proven itself as the enabling technology for the convergence of voice, video and data networks. As business needs evolve, so do demands on QoS technologies. The need to protect voice, video and critical data via QoS mechanisms in an enterprise network has escalated over the past few years, primarily due to the increased frequency and sophistication of Denial of Service (DoS) and worm attacks. This document examines current QoS demands and requirements within the enterprise and presents strategic design recommendations to address these needs.

QoS Overview

This section answers the following questions:

- What is QoS?
- Why is QoS Important for Enterprise Networks?

What is QoS?

QoS is the measure of transmission quality and service availability of a network (or internetworks).

Service availability is a crucial foundation element of QoS. The network infrastructure must be designed to be highly available before you can successfully implement QoS. The target for High Availability is 99.999 % uptime, with only five minutes of downtime permitted per year. The transmission quality of the network is determined by the following factors:

- Loss—A relative measure of the number of packets that were not received compared to the total number of packets transmitted. Loss is typically a function of availability. If the network is Highly Available, then loss during periods of non-congestion would be essentially zero. During periods of congestion, however, QoS mechanisms can determine which packets are more suitable to be selectively dropped to alleviate the congestion.

- Delay—The finite amount of time it takes a packet to reach the receiving endpoint after being transmitted from the sending endpoint. In the case of voice, this is the amount of time it takes for a sound to travel from the speaker's mouth to a listener's ear.
- Delay variation (Jitter)—The difference in the end-to-end delay between packets. For example, if one packet requires 100 ms to traverse the network from the source endpoint to the destination endpoint and the following packet requires 125 ms to make the same trip, then the delay variation is 25 ms.

Each end station in a Voice over IP (VoIP) or Video over IP conversation uses a *jitter buffer* to smooth out changes in the arrival times of voice data packets. Although jitter buffers are dynamic and adaptive, they may not be able to compensate for instantaneous changes in arrival times of packets. This can lead to jitter buffer over-runs and under-runs, both of which result in an audible degradation of call quality.

Why is QoS Important for Enterprise Networks?

A communications network forms the backbone of any successful organization. These networks transport a multitude of applications, including realtime voice, high-quality video and delay-sensitive data. Networks must provide predictable, measurable, and sometimes guaranteed services by managing bandwidth, delay, jitter and loss parameters on a network.

QoS technologies refer to the set of tools and techniques to manage network resources and are considered the key enabling technology for network convergence. The objective of QoS technologies is to make voice, video and data convergence appear transparent to end users. QoS technologies allow different types of traffic to contend inequitably for network resources. Voice, video, and critical data applications may be granted priority or preferential services from network devices so that the quality of these strategic applications does not degrade to the point of being unusable. Therefore, QoS is a critical, intrinsic element for successful network convergence.

QoS tools are not only useful in protecting desirable traffic, but also in providing deferential services to undesirable traffic such as the exponential propagation of worms. You can use QoS to monitor flows and provide first and second order reactions to abnormal flows indicative of such attacks, as will be discussed in additional detail later in this document.

What is the Cisco QoS Toolset?

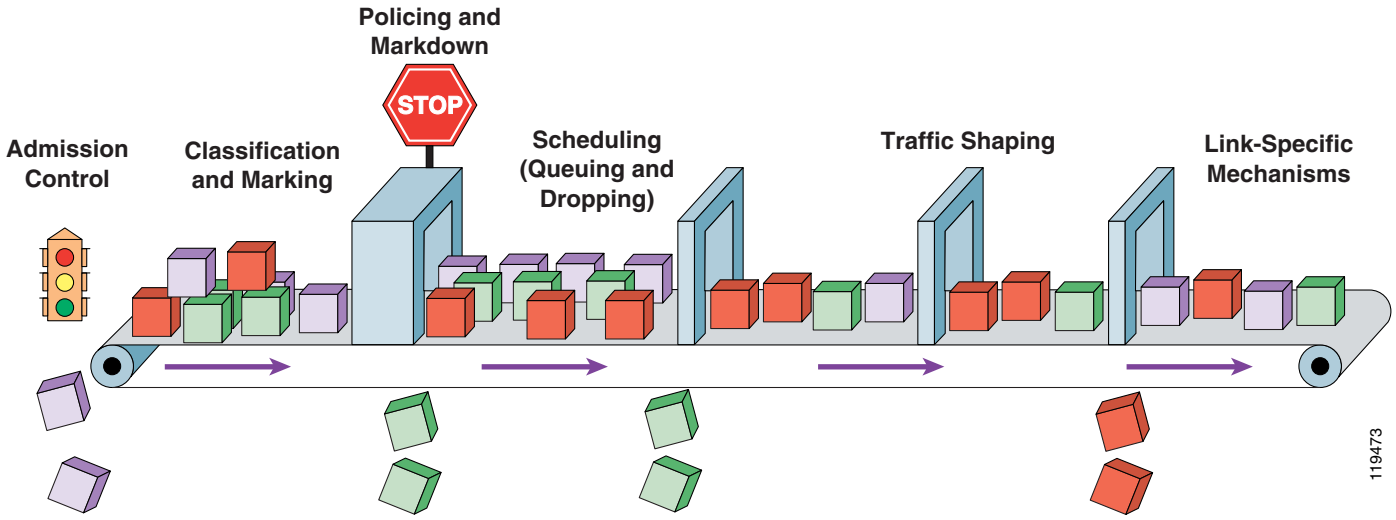
This section describes the main categories of the Cisco QoS toolset and includes the following topics:

- Classification and Marking tools
- Policing and Markdown tools
- Scheduling tools
- Link-specific tools
- AutoQoS tools
- Call Admission Control tools

Cisco provides a complete toolset of QoS features and solutions for addressing the diverse needs of voice, video and multiple classes of data applications. Cisco QoS technology lets complex networks control and predictably service a variety of networked applications and traffic types. You can effectively control bandwidth, delay, jitter, and packet loss with these mechanisms. By ensuring the desired results,

the QoS features lead to efficient, predictable services for business-critical applications. Using the rich Cisco QoS toolset, as shown in Figure 1-1, businesses can build networks that conform to the Differentiated Services (DiffServ) architecture, as defined in RFC 2475.

Figure 1-1 The Cisco QoS Toolset



Classification and Marking Tools

The first element to a QoS policy is to classify/identify the traffic that is to be treated differently. Following classification, marking tools can set an attribute of a frame or packet to a specific value. Such marking (or remarking) establishes a trust boundary that scheduling tools later depend on.

Classification and marking tools set this trust boundary by examining any of the following:

- Layer 2 parameters—802.1Q Class of Service (CoS) bits, Multiprotocol Label Switching Experimental Values (MPLS EXP)
- Layer 3 parameters—IP Precedence (IPP), Differentiated Services Code Points (DSCP), IP Explicit Congestion Notification (ECN), source/destination IP address
- Layer 4 parameters— L4 protocol (TCP/UDP), source/destination ports
- Layer 7 parameters— application signatures via Network Based Application Recognition (NBAR)

NBAR is a Cisco proprietary technology that identifies application layer protocols by matching them against a Protocol Description Language Module (PDLM), which is essentially an application signature. The NBAR deep-packet classification engine examines the data payload of stateless protocols against PDLMs. There are over 98 PDLMs embedded into Cisco IOS software 12.3 code. Additionally, Cisco IOS software 12.3(4)T introduces the ability to define custom PDLMs which examine user-defined strings within packet payloads. PDLMs can be added to the system without requiring an IOS upgrade because they are modular. NBAR is dependent on Cisco Express Forwarding (CEF) and performs deep-packet classification only on the first packet of a flow. The remainder of the packets belonging to the flow is then CEF-switched.

You can only apply policies to traffic after it has been positively classified. To avoid the need for repetitive and detailed classification at every node, packets can be marked according to their service levels. An analogy: imagine that each individual in the postal system would have to open up each letter to determine the respective priority required and service it accordingly. Obviously it would be better to

have the first mail-clerk stamp something on the outside of the envelope to indicate the priority level that would be applied during each phase of processing and delivery. Similarly, marking tools can be used to indicate respective priority levels by setting attributes in the frame/packet headers so that detailed classification does not have to be recursively performed at each hop. Within an enterprise, marking is done at either Layer 2 or Layer 3, using the following fields:

- 802.1Q/p Class of Service (CoS)—Ethernet frames can be marked at Layer 2 with their relative importance by setting the 802.1p User Priority bits of the 802.1Q header. Only three bits are available for 802.1p marking. Therefore, only 8 classes of service (0-7) can be marked on Layer 2 Ethernet frames.
- IP Type of Service (ToS) byte—Layer 2 media often changes as packets traverse from source to destination, so a more ubiquitous classification occurs at Layer 3. The second byte in an IPv4 packet is the ToS byte. The first three bits of the ToS byte are the IPP bits. These first three bits combined with the next three bits are known collectively as the DSCP bits.

  The IP Precedence bits, like 802.1p CoS bits, allow for only the following 8 values of marking (0–7):

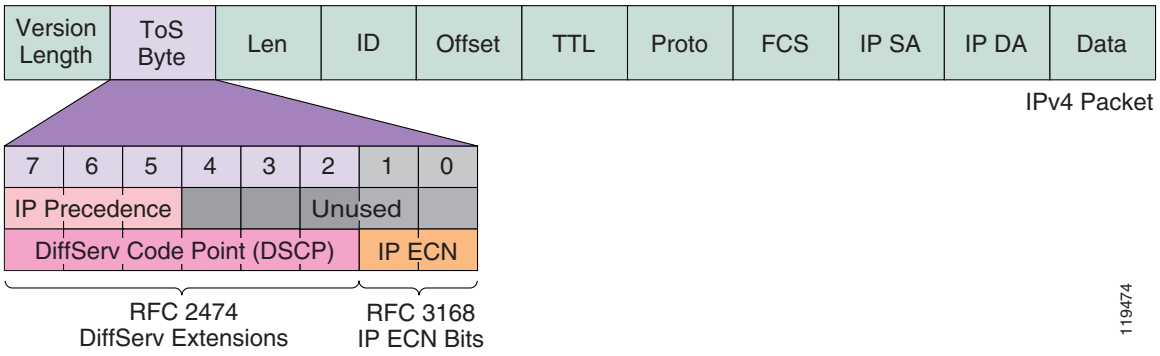
  - IPP values 6 and 7 are generally reserved for network control traffic such as routing.
  - IPP value 5 is recommended for voice.
  - IPP value 4 is shared by videoconferencing and streaming video.
  - IPP value 3 is for voice control.
  - IPP values 1 and 2 can be used for data applications.
  - IPP value 0 is the default marking value.

  Many enterprises find IPP marking to be overly restrictive and limiting, favoring instead the 6-Bit/64-value DSCP marking model.
- DSCPs and Per-Hop Behaviors (PHBs)—DSCP values can be expressed in numeric form or by special standards-based names called Per-Hop Behaviors. There are four broad classes of DSCP PHB markings: Best Effort (BE or DSCP 0), RFC 2474 Class Selectors (CS1–CS7, which are identical/backwards-compatible to IPP values 1–7), RFC 2597 Assured Forwarding PHBs (AF*xy*), and RFC 3268 Expedited Forwarding (EF).

  There are four Assured Forwarding classes, each of which begins with the letters “AF” followed by two numbers. The first number corresponds to the DiffServ Class of the AF group and can range from 1 through 4. The second number refers to the level of Drop Preference within each AF class and can range from 1 (lowest Drop Preference) through 3 (highest Drop Preference).

  DSCP values can be expressed in decimal form or with their PHB keywords. For example, DSCP EF is synonymous with DSCP 46, and DSCP AF31 is synonymous with DSCP 26.
- IP Explicit Congestion Notification (IP ECN)—IP ECN, as defined in RFC 3168, makes use of the last two bits of the IP ToS byte, which are not used by the 6-bit DSCP markings, as shown in Figure 1-2.

Figure 1-2 The IP ToS Byte (DSCP and IP ECN)



These last two bits are used to indicate to TCP senders whether or not congestion was experienced during transit. In this way, TCP senders can adjust their TCP windows so that they do not send more traffic than the network can service. Previously, dropping packets was the only way that congestion feedback could be signaled to TCP senders. Using IP ECN, however, congestion notification can be signaled without dropping packets. The first IP ECN bit (7th in the ToS byte) is used to indicate whether the device supports IP ECN and the second bit (last bit in the IP ToS byte) is used to indicate whether congestion was experienced (0="no congestion"; 1= "congestion was experienced"). IP ECN can be marked through a congestion avoidance mechanism such as weighted early random detection (WRED).

Policing and Markdown Tools

Policing tools (policers) determine whether packets are conforming to administratively-defined traffic rates and take action accordingly. Such action could include marking, remarking or dropping a packet.

A basic policer monitors a single rate: traffic equal to or below the defined rate is considered to *conform* to the rate, while traffic above the defined rate is considered to *exceed* the rate. On the other hand, the algorithm of a dual-rate policer (such as described in RFC 2698) is analogous to a traffic light. Traffic equal to or below the principal defined rate (green light) is considered to *conform* to the rate. An allowance for moderate amounts of traffic above this principal rate is permitted (yellow light) and such traffic is considered to *exceed* the rate. However, a clearly-defined upper-limit of tolerance is set (red light), beyond which traffic is considered to *violate* the rate.

Policers complement classification and marking policies. For example, as previously discussed, RFC 2597 defines the AF classes of PHBs. Traffic conforming to the defined rate of a given AF class is marked to the first Drop Preference level of a given AF class (for example, AF21). Traffic exceeding this rate is marked down to the second Drop Preference level (for example, AF22) and violating traffic is either marked down further to the third Drop Preference level (for example, AF23) or simply dropped.

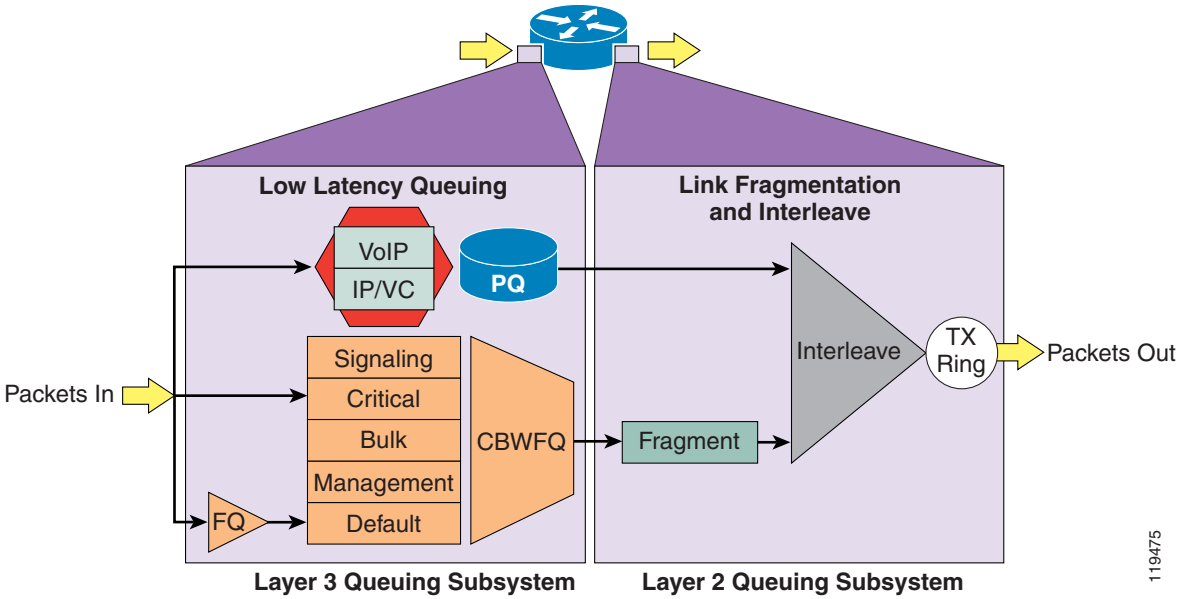
Scheduling Tools

Scheduling tools determine how a frame/packet exits a device. Whenever packets enter a device faster than they can exit it, such as with speed mismatches, then a point of congestion, or bottleneck, can occur. Devices have buffers that allow for scheduling higher-priority packets to exit sooner than lower priority ones, which is commonly called queueing.

Queueing algorithms are activated only when a device is experiencing congestion and are deactivated when the congestion clears. The main Cisco IOS software queuing tools are Low Latency Queueing (LLQ), which provides strict priority servicing and is intended for realtime applications such as VoIP; and Class-Based Weighted Fair Queuing (CBWFQ), which provides bandwidth guarantees to given classes of traffic and fairness to discrete traffic flows within these traffic classes.

Figure 1-3 shows the Layer 3 and Layer 2 queuing subsystems of the Cisco IOS software LLQ/CBWFQ algorithm.

Figure 1-3 LLQ/CBWFQ Operation



Queueing buffers act like a funnel for water being poured into a small opening. If water enters the funnel faster than it exits, eventually the funnel overflows from the top. When queueing buffers begin overflowing from the top, packets may be dropped either as they arrive (tail drop) or selectively before all buffers are filled.

Selective dropping of packets when the queues are filling is referred to as *congestion avoidance*. Congestion avoidance mechanisms work best with TCP-based applications because selective dropping of packets causes the TCP windowing mechanisms to "throttle-back" and adjust the rate of flows to manageable rates.

Congestion avoidance mechanisms are complementary to queueing algorithms. Queueing algorithms manage the front of a queue while congestion avoidance mechanisms manage the tail of the queue. Congestion avoidance mechanisms thus indirectly affect scheduling.

The principle IOS congestion avoidance mechanism is WRED, which randomly drops packets as queues fill to capacity. However, the randomness of this selection can be skewed by traffic weights. The weight can either be IP Precedence values, as is the case with default WRED which drops *lower* IPP values more aggressively (for example, IPP 1 would be dropped more aggressively than IPP 6) or the weights can be AF Drop Preference values, as is the case with DSCP-Based WRED which drops *higher* AF Drop Preference values more aggressively (for example, AF23 is dropped more aggressively than AF22, which in turn is dropped more aggressively than AF21). WRED can also be used to set the IP ECN bits to indicate that congestion was experienced in transit.

Link-Specific Tools

Link-specific tools include the following:

- Shaping tools—A shaper typically delays excess traffic above an administratively-defined rate using a buffer to hold packets and shape the flow when the data rate of the source is higher than expected.
- Link Fragmentation and Interleaving tools—With slow-speed WAN circuits, large data packets take an excessively long time to be placed onto the wire. This delay, called *serialization delay*, can easily cause a VoIP packet to exceed its delay and/or jitter threshold. There are two main tools to mitigate serialization delay on slow (768 kbps) links: Multilink PPP Link Fragmentation and Interleaving (MLP LFI) and Frame Relay Fragmentation (FRF.12).
- Compression tools—Compression techniques, such as compressed Real-Time Protocol (cRTP), minimize bandwidth requirements and are highly useful on slow links. At 40 bytes total, the header portion of a VoIP packet is relatively large and can account for nearly two-thirds or the entire VoIP packet (as in the case of G.729 VoIP). To avoid the unnecessary consumption of available bandwidth, you can use cRTP on a link-by-link basis. cRTP compresses IP/UDP/RTP headers from 40 bytes to between two and five bytes (which results in a bandwidth savings of approximately 66% for G.729 VoIP).
- Transmit ring (Tx-Ring) tuning—The Tx-Ring is a final interface First-In-First-Out (FIFO) queue that holds frames to be immediately transmitted by the physical interface. The Tx-Ring ensures that a frame is always available when the interface is ready to transmit traffic, so that link utilization is driven to 100 % of capacity. The size of the Tx-Ring is dependant on the hardware, software, Layer 2 media, and queueing algorithm configured on the interface. The Tx-Ring may have to be tuned on certain platforms/interfaces to prevent unnecessary delay/jitter introduced by this final FIFO queue.

AutoQoS Tools

The richness of the Cisco QoS toolset inevitably increases its deployment complexity. To address customer demand for simplification of QoS deployment, Cisco has developed the Automatic QoS (AutoQoS) features. AutoQoS is an intelligent macro that allows an administrator to enter one or two simple AutoQoS commands to enable all the appropriate features for the recommended QoS settings for an application on a specific interface.

AutoQoS VoIP, the first release of AutoQoS, provides best-practice QoS designs for VoIP on Cisco Catalyst switches and Cisco IOS routers. By entering one global and/or one interface command, depending on the platform, the AutoQoS VoIP macro expands these commands into the recommended VoIP QoS configurations (complete with all the calculated parameters and settings) for the platform and interface on which the AutoQoS is being applied.

For Campus Catalyst switches, AutoQoS automatically performs the following tasks:

- Enforces a trust boundary at Cisco IP Phones.
- Enforces a trust boundary on Catalyst switch access ports and uplinks/downlinks.
- Enables Catalyst strict priority queuing for voice and weighted round robin queuing for data traffic.
- Modifies queue admission criteria (CoS-to-queue mappings).
- Modifies queue sizes as well as queue weights where required.
- Modifies CoS-to-DSCP and IP Precedence-to-DSCP mappings.

For Cisco IOS routers, AutoQoS is supported on Frame Relay (FR), Asynchronous Transfer Mode (ATM), High-Level Data Link Control (HDLC), Point-to-Point Protocol (PPP), and FR-to-ATM links.

For Cisco IOS routers, AutoQoS automatically performs the following tasks:

- Classifies and marks VoIP bearer traffic (to DSCP EF) and Call-Signaling traffic (to DSCP CS3).
  - Applies scheduling:
  - Low Latency Queuing (LLQ) for voice
  - Class-Based Weighted Fair Queuing (CBWFQ) for Call-Signaling
  - Fair Queuing (FQ) for all other traffic
- Enables Frame Relay Traffic Shaping (FRTS) with optimal parameters, if required.
- Enables Link Fragmentation and Interleaving (LFI), either MLP LFI or FRF.12, on slow (768 kbps) links, if required.
- Enables IP RTP header compression (cRTP), if required.
- Provides Remote Monitoring (RMON) alerts of dropped VoIP packets.

AutoQoS VoIP became available on Cisco IOS router platforms in 12.2(15)T.

In its second release, for Cisco IOS routers only, AutoQoS Enterprise detects and provisions for up to ten classes of traffic, including the following:

- Voice
- Interactive-Video
- Streaming-Video
- Call-Signaling
- Transactional Data
- Bulk Data
- Routing
- Network Management
- Best Effort
- Scavenger

These classes will be explained in more detail later in this document.

The AutoQoS Enterprise feature consists of two configuration phases, completed in the following order:

- Auto Discovery (data collection)—Uses NBAR-based protocol discovery to detect the applications on the network and performs statistical analysis on the network traffic.
- AutoQoS template generation and installation—Generates templates from the data collected during the Auto Discovery phase and installs the templates on the interface. These templates are then used as the basis for creating the class maps and policy maps for your network. After the class maps and policy maps are created, they are then installed on the interface.

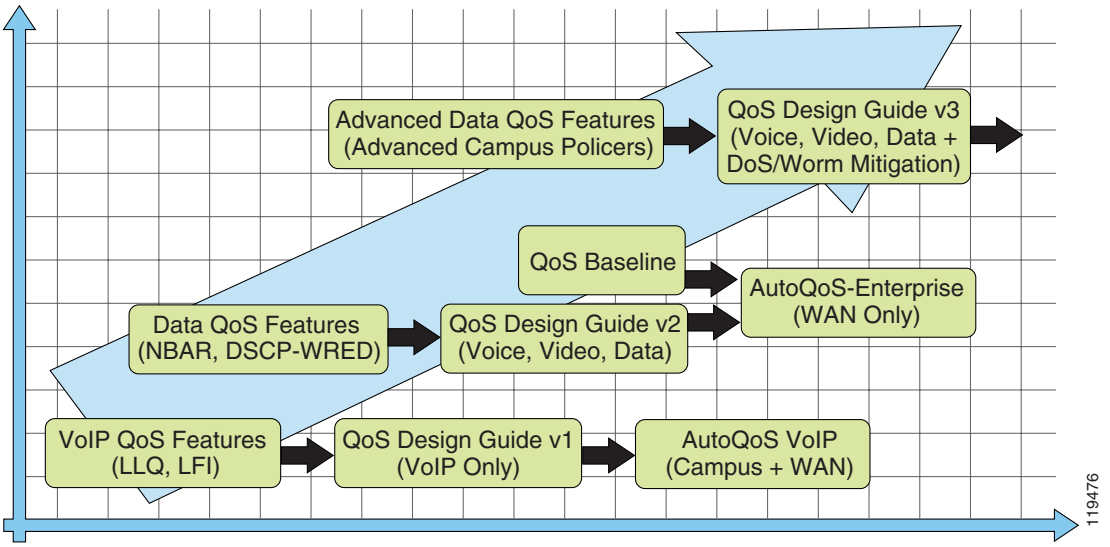
AutoQoS Enterprise became available on Cisco routers in Cisco IOS 12.3(7)T.

Some may naturally then ask: Why should I read the separate QoS design document when I have AutoQoS? While it is true that AutoQoS-VoIP is an excellent tool for customers with the objective of enabling QoS for VoIP (only) on their Campus and WAN infrastructures, and AutoQoS-Enterprise is a fine tool for enabling basic Branch-router WAN-Edge QoS for voice, video and multiple classes of data. For customers that have such basic QoS needs and don't have the time or desire to learn or do more with QoS, AutoQoS is definitely the way to go.

However, it is important to remember where AutoQoS came from. AutoQoS tools are the result of Cisco QoS feature development coupled with Cisco QoS Design Guides based on large-scale lab-testing. AutoQoS VoIP is the product of the first QoS Design Guide, one of the most popular/downloaded technical white papers ever produced within Cisco. AutoQoS Enterprise is the result of the strategic QoS Baseline (discussed later in this document) coupled with the second generation QoS Design Guide. These latest QoS design documents represents the third-generation QoS Design Guide, which is essentially a proposed blueprint for the next version of AutoQoS.

Figure 1-4 shows the relationship between Cisco QoS features, Design Guides, and AutoQoS.

Figure 1-4 Cisco QoS Feature, Design Guide and AutoQoS Evolution



Call Admission Control Tools

After performing the calculations to provision the network with the required bandwidth to support voice, video and data applications, you must ensure that voice or video do not oversubscribe the portion of the bandwidth allocated to them. While most DiffServ QoS features are used to protect voice from data, Call Admission Control (CAC) tools are used to protect voice from voice and video from video.

CAC tools fall into the following three main categories:

- Local—Local CAC mechanisms are a voice gateway router function, typically deployed on the outgoing gateway. The CAC decision is based on nodal information such as the state of the outgoing LAN/WAN link that the voice call traverses if allowed to proceed. Local mechanisms include configuration items to disallow more than a fixed number of calls.

  If the network designer already knows that no more than five VoIP calls will fit across the outgoing WAN link's LLQ configuration because of bandwidth limitations, then it would be recommended to configure the local gateway node to not allow more than five simultaneous calls.

- Measurement-based—Measurement-based CAC techniques look ahead into the packet network to gauge the state of the network to determine whether or not to allow a new call. This usually implies sending probes to the destination IP address, which could be the terminating gateway or endpoint, or another device in between.

The probes return to the outgoing gateway or endpoint information on the conditions found while traversing the network to the destination. Typically, loss and delay characteristics are the interesting elements of information for voice CAC decisions. The outgoing device then uses this information in combination with configured information to decide if the network conditions exceed a given or configured threshold.

- Resource-based—There are two types of resource-based mechanisms: those that calculate resources needed and/or available, and those that reserve resources for the call. Resources of interest include link bandwidth, DSPs and DS0 timeslots on the connecting TDM trunks to a voice gateway, CPU power and memory. Several of these resources could be constrained at one or more nodes that the call traverses to its destination.

Cisco CallManager has additional CAC features to handle management of VoIP network deployments. These features are not mutually exclusive to the features listed above. While CallManager Location-Based CAC is deployed in the overall network to manage VoIP bandwidth availability for both Cisco IP Phones and voice gateways, local measurement-based or resource-based features may be deployed at the same time on the voice gateway to push back calls into the private Branch exchange (PBX) or publicly-switched telephone network (PSTN) if IP network conditions do not allow their entry into the VoIP network.

Note A detailed discussion of CAC configuration is beyond the scope of this document, but CAC configuration is crucial for a successful VoIP deployment. For additional information on CallManager CAC, refer to the IP Telephony Solution Reference Network Design Guide at http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_implementation_design_guides_list.html.

How is QoS Optimally Deployed within the Enterprise?

A successful QoS deployment is comprised of multiple phases, including:

1. Strategically defining the business objectives to be achieved via QoS.
2. Analyzing the service-level requirements of the various traffic classes to be provisioned for.
3. Designing and testing QoS policies prior to production-network rollout.
4. Rolling out the tested QoS designs to the production network.
5. Monitoring service levels to ensure that the QoS objectives are being met.

These phases may need to be repeated as business conditions change and evolve.

Each of these phases will be addressed in more detail in the following sections.

1) Strategically Defining QoS Objectives

QoS technologies are the enablers for business/organizational objectives. Therefore, the way to begin a QoS deployment is not to activate QoS features simply because they exist, but to start by clearly defining the objectives of the organization. For example, among the first questions that arise during a QoS deployment are: How many traffic classes should be provisioned for? And what should they be?

To help answer these fundamental questions, organizational objectives need to be defined, such as:

- Is the objective to enable VoIP only or video also required?
- If so, is video-conferencing required or streaming video? Or both?

- Are there applications that are considered mission-critical, and if so, what are they?
- Does the organization wish to squelch certain types of traffic, and if so, what are they?

To help address these crucial questions and to simplify QoS, Cisco has adopted a new initiative called the "QoS Baseline." The QoS Baseline is a strategic document designed to unify QoS within Cisco, from enterprise to service provider and from engineering to marketing. The QoS Baseline was written by Cisco's most qualified QoS experts, who have developed or contributed to the related IETF RFC standards (as well as other technology standards) and are thus eminently qualified to interpret these standards. The QoS Baseline also provides uniform, standards-based recommendations to help ensure that QoS designs and deployments are unified and consistent.

The QoS Baseline defines up to 11 classes of traffic that may be viewed as critical to a given enterprise. A summary these classes and their respective standards-based marking recommendations are presented in Table 1-1.

Table 1-1 Cisco QoS Baseline/Technical Marketing (Interim) Classification and Marking Recommendations

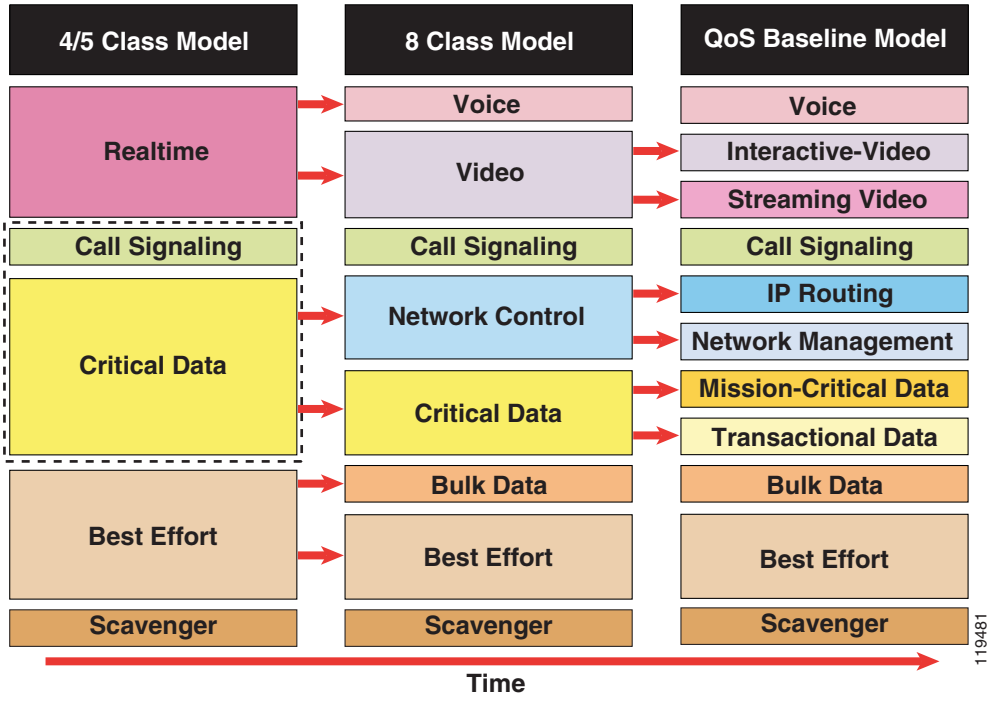
Application	Layer 3 Classification			Layer 2 CoS/MPLS EXP
	IPP	PHB	DSCP	
IP Routing	6	CS6	48	6
Voice	5	EF	46	5
Interactive Video	4	AF41	34	4
Streaming-Video	4	CS4	32	4
Locally-Defined Mission-Critical Data (see note below)	3	—	25	3
Call-Signaling (see note below)	3	AF31/CS3	26/24	3
Transactional Data	2	AF21	18	2
Network Management	2	CS2	16	2
Bulk Data	1	AF11	10	1
Scavenger	1	CS1	8	1
Best Effort	0	0	0	0

Note The QoS Baseline recommends marking Call-Signaling to CS3. However, currently most Cisco IP Telephony products mark Call-Signaling to AF31. A marking migration from AF31 to CS3 is under way within Cisco, but in the interim it is recommended that both AF31 and CS3 be reserved for Call-Signaling and that Locally-Defined Mission-Critical Data applications be marked to a temporary placeholder non-standard DSCP, such as 25. Upon completion of the migration, the QoS Baseline marking recommendations of CS3 for Call-Signaling and AF31 for Locally-Defined Mission-Critical Data applications should be used. These marking recommendations are more in line with RFC 2474 and RFC 2597.

Enterprises do not need to deploy all 11 classes of the QoS Baseline model. This model is intended to be a forward-looking guide that considers as many classes of traffic with unique QoS requirements as possible. Familiarity with this model can assist in the smooth expansion of QoS policies to support additional applications as future requirements arise. However, at the time of QoS deployment, the enterprise needs to clearly define their organizational objectives, which will correspondingly determine how many traffic classes will be required.

This consideration should be tempered with the determination of how many application classes the networking administration team feels comfortable with deploying and supporting. Platform-specific constraints or service-provider constraints may also affect the number of classes of service. At this point you should also consider a migration strategy to allow the number of classes to be smoothly expanded as future needs arise, as shown in Figure 1-5.

Figure 1-5 Example Strategy for Expanding the Number of Classes of Service over Time



Always seek executive endorsement of the QoS objectives prior to design and deployment. QoS is a system of managed unfairness and as such almost always bears political and organizational repercussions when implemented. To minimize the effects of these non-technical obstacles to deployment, address these political and organizational issues as early as possible, garnishing executive endorsement whenever possible.

A strategic standards-based guide like the QoS Baseline coupled with a working knowledge of QoS tools and syntax is a prerequisite for any successful QoS deployment. However, you must also understand the service-level requirements of the various applications requiring preferential or deferential treatment within the network.

2) Analyzing Application Service-Level Requirements

The following sections present an overview of the QoS requirements for voice, video and multiple classes of data, including the following topics:

- QoS Requirements of VoIP
- QoS Requirements of Video
- QoS Requirements of Data Applications
- QoS Requirements of the Control Plane
- QoS Requirements of the Scavenger Class

QoS Requirements of VoIP

This section includes the following topics:

- Voice (Bearer Traffic)
- Call-Signaling Traffic

VoIP deployments require provisioning explicit priority servicing for VoIP (bearer stream) traffic and a guaranteed bandwidth service for Call-Signaling traffic. These related classes will be examined separately.

Voice (Bearer Traffic)

A summary of the key QoS requirements and recommendations for Voice (bearer traffic) are:

- **Voice** traffic should be marked to **DSCP EF** per the QoS Baseline and RFC 3246.
- **Loss** should be no more than **1 %.**
- One-way **Latency** (mouth-to-ear) should be no more than **150 ms**.
- Average one-way **Jitter** should be targeted under **30 ms**.
- **21–320 kbps of guaranteed priority bandwidth is required per call** (depending on the sampling rate, VoIP codec and Layer 2 media overhead).

Voice quality is directly affected by all three QoS quality factors: loss, latency and jitter.

Loss causes voice clipping and skips. The packetization interval determines the size of samples contained within a single packet. Assuming a 20 ms (default) packetization interval, the loss of two or more consecutive packets results in a noticeable degradation of voice quality. VoIP networks are typically designed for very close to zero percent VoIP packet loss, with the only actual packet loss being due to L2 bit errors or network failures.

Excessive latency can cause voice quality degradation. The goal commonly used in designing networks to support VoIP is the target specified by ITU standard G.114, which states that 150 ms of one-way, end-to-end (mouth-to-ear) delay ensures user satisfaction for telephony applications. A design should apportion this budget to the various components of network delay (propagation delay through the backbone, scheduling delay due to congestion, and the access link serialization delay) and service delay (due to VoIP gateway codec and de-jitter buffer).

If the end-to-end voice delay becomes too long, the conversation begins to sound like two parties talking over a satellite link or even a CB radio. While the ITU G.114 states that a 150 ms one-way (mouth-to-ear) delay budget is acceptable for high voice quality, lab testing has shown that there is a negligible difference in voice quality Mean Opinion Scores (MOS) using networks built with 200 ms delay budgets. Cisco thus recommends designing to the ITU standard of 150 ms, but if constraints exist where this delay target cannot be met, then the delay boundary can be extended to 200 ms without significant impact on voice quality.

Note Higher delays may also be viewed as acceptable to certain organizations, but the corresponding reduction in VoIP quality must be taken into account when making such design decisions.

Jitter buffers (also known 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 balance the delay and the probability of interrupted playout due to late packets. Late or out-of-order packets are discarded.

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

An underflow is when the buffer is empty when the codec needs to play out a sample. An overflow is when the jitter buffer is already full and another packet arrives that cannot therefore be enqueued in the jitter buffer. Both jitter buffer underflows and overflows cause packets to be discarded.

Adaptive jitter buffers aim to overcome these issues by dynamically tuning the jitter buffer size to the lowest acceptable value. Well-designed adaptive jitter buffer algorithms should not impose any unnecessary constraints on the network design by:

Instantly increasing the jitter buffer size to the current measured jitter value following a jitter buffer overflow.

Slowly decreasing the jitter buffer size when the measured jitter is less that the current jitter buffer size.

Using a Programmable Logic Controller (PLC) to interpolate for the loss of a packet on a jitter buffer underflow.

Where such adaptive jitter buffers are used, we can in theory engineer out explicit considerations of jitter by accounting for worst-case per hop delays. Advanced formulas can be used to arrive at network-specific design recommendations for jitter based on maximum and minimum per-hop delays. Alternatively, this 30 ms value can be used as a jitter target as extensive lab testing has shown that when jitter consistently exceeds 30 ms voice quality degrades significantly.

Because of its strict service-level requirements, VoIP is well suited to the Expedited Forwarding Per-Hop Behavior, as defined in RFC 3246 (formerly RFC 2598). It should therefore be marked to DSCP EF (46) and assigned strict priority servicing at each node, regardless of whether such servicing is done in hardware (as in Catalyst switches via hardware priority queuing) or in software (as in Cisco IOS routers via LLQ).

The bandwidth consumed by VoIP streams (in bps) is calculated by adding the VoIP sample payload (in bytes) to the 40-byte IP/UDP/RTP headers (assuming that cRTP is not in use), multiplying this value by 8 (to convert it to bits) and then multiplying again by the packetization rate (default of 50 packets per second).

Table 1-2 details the bandwidth per VoIP flow at a default packet rate of 50 packets per second (pps). This does not include Layer 2 overhead and does not take into account any possible compression schemes, such as cRTP.

Table 1-2 Voice Bandwidth (without Layer 2 Overhead)

Bandwidth Consumption	Sampling Rate	Voice Payload in Bytes	Packets per Second	Bandwidth per Conversation
G.711	20 ms	160	50	80 kbps

Table 1-2 Voice Bandwidth (without Layer 2 Overhead)

G.711	30 ms	240	33	74 kbps
G.729A	20 ms	20	50	24 kbps
G.729A	30 ms	30	33	19 kbps

Note The Service Parameters menu in Cisco CallManager Administration can be used to adjust the packet rate. It is possible to configure the sampling rate above 30 ms, but this usually results in poor voice quality.

A more accurate method for provisioning VoIP is to include the Layer 2 overhead, which includes preambles, headers, flags, cyclic redundancy checks (CRCs), and ATM cell-padding. The amount of overhead per VoIP call depends on the Layer 2 technology used:

- 802.1Q Ethernet adds (up to) 32 bytes of Layer 2 overhead.
- Point-to-point protocol (PPP) adds 12 bytes of Layer 2 overhead.
- Multilink PPP (MLP) adds 13 bytes of Layer 2 overhead.
- Frame Relay adds 4 bytes of Layer 2 overhead; Frame Relay with FRF.12 adds 8 bytes.
- ATM adds varying amounts of overhead, depending on the cell padding requirements.

Table 1-3 shows a more accurate bandwidth provisioning example for voice because it includes Layer 2 overhead.

Table 1-3 Voice Bandwidth (Including Layer 2 Overhead)

Bandwidth Consumption	802.1Q Ethernet	PPP	MLP	Frame-Relay w/FRF.12	ATM
G.711 at 50 pps	93 kbps	84 kbps	86 kbps	84 kbps	106 kbps
G.711 at 33 pps	83 kbps	77 kbps	78 kbps	77 kbps	84 kbps
G.729A at 50 pps	37 kbps	28 kbps	30 kbps	28 kbps	43 kbps
G.729A at 33 pps	27 kbps	21 kbps	22 kbps	21 kbps	28 kbps

Note A handy tool for quickly and accurately calculating VoIP bandwidth requirements (factoring in the codec, the use of cRTP and L2 overhead) can be found at: http://tools.cisco.com/Support/VBC/jsp/Codec_Calc1.jsp

Call-Signaling Traffic

The following are key QoS requirements and recommendations for Call-Signaling traffic:

- **Call-Signaling** traffic should be marked as **DSCP CS3** per the QoS Baseline (during migration, it may also be marked the legacy value of DSCP AF31).
- **150 bps** (plus Layer 2 overhead) per phone of **guaranteed bandwidth** is required for voice control traffic; more may be required, depending on the call signaling protocol(s) in use.

Call-Signaling traffic was originally marked by Cisco IP Telephony equipment to DSCP AF31. However, the Assured Forwarding classes, as defined in RFC 2597, were intended for flows that could be subject to markdown and – subsequently – the aggressive dropping of marked-down values. Marking down and aggressively dropping Call-Signaling could result in noticeable delay-to-dial-tone (DDT) and lengthy call setup times, both of which generally translate to poor user experiences.

The QoS Baseline changed the marking recommendation for Call-Signaling traffic to DSCP CS3 because Class Selector code points, as defined in RFC 2474, were not subject to markdown/aggressive dropping. Some Cisco IP Telephony products have already begun transitioning to DSCP CS3 for Call-Signaling marking. In this interim period, both code-points (CS3 and AF31) should be reserved for Call-Signaling marking until the transition is complete.

- Many Cisco IP phones use Skinny Call-Control Protocol (SCCP) for call signaling. SCCP is a relatively lightweight protocol that requires only a minimal amount of bandwidth protection. However, newer versions of CallManager and SCCP have improved functionality requiring new message sets yielding a higher bandwidth consumption. Cisco signaling bandwidth design recommendations have been adjusted to match. The IPT SRND's Network Infrastructure chapter contains the relevant details, available at: http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_implementation_design_guides_list.html.
- Other call signaling protocols include (but are not limited to) H.323, H.225, Session Initiated Protocol (SIP) and Media Gateway Control Protocol (MGCP). Each call signaling protocol has unique TCP/UDP ports and traffic patterns that should be taken into account when provisioning QoS policies for them.

QoS Requirements of Video

This section describes the two main types of video traffic, and includes the following topics:

- Interactive Video
- Streaming Video

Interactive Video

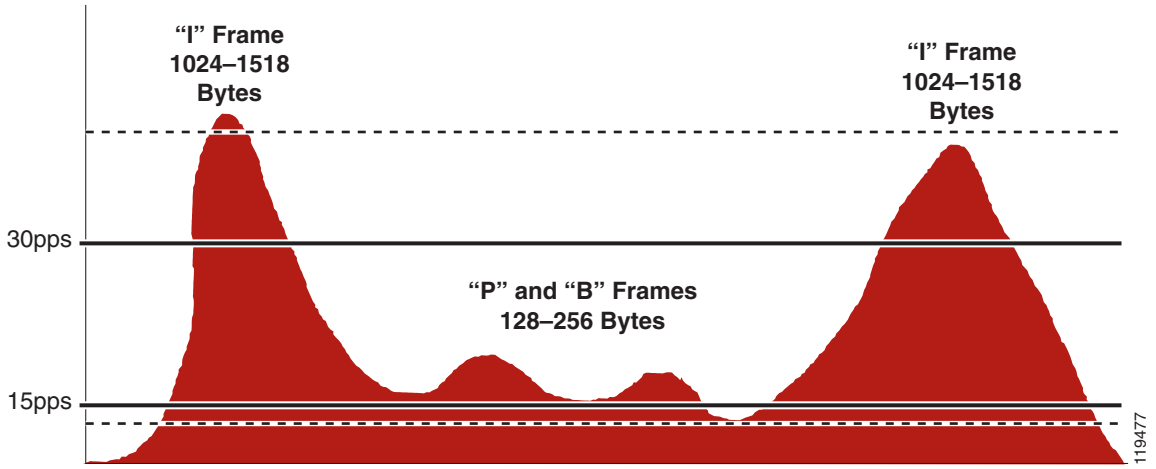
When provisioning for Interactive Video (IP Videoconferencing) traffic, the following guidelines are recommended:

- **Interactive Video** traffic should be marked to **DSCP AF41**; excess Interactive-Video traffic can be marked down by a policer to AF42 or AF43.
- **Loss** should be no more than **1 %.**
- One-way **Latency** should be no more than **150 ms**.
- **Jitter** should be no more than **30 ms**.
- **Overprovision Interactive Video queues by 20%** to accommodate bursts

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 radically different from voice.

For example, videoconferencing traffic has varying packet sizes and extremely variable packet rates, as shown in Figure 1-6.

Figure 1-6 IP Videoconferencing Traffic Rates and Packet Sizes



The videoconferencing rate is the sampling rate of the video stream, not the actual bandwidth the video call requires. In other words, the data payload of videoconferencing packets is filled with 384 kbps worth of video and voice samples.

IP, UDP, and RTP headers (40 bytes per packet, uncompressed) need to be included in IP/VC bandwidth provisioning, as does the Layer 2 overhead of the media in use. Because (unlike VoIP) IP/VC packet sizes and rates vary, the header overhead percentage will vary as well, so an absolute value of overhead cannot be accurately calculated for all streams. Testing, however, has shown a conservative rule of thumb for IP/VC bandwidth provisioning is to overprovision the guaranteed/priority bandwidth by 20 percent. For example, a 384 kbps IP/VC stream would be adequately provisioned with an LLQ/CBWFQ of 460 kbps.

Note The Cisco LLQ algorithm has been implemented to include a default burst parameter equivalent to 200 ms worth of traffic. Testing has shown that this burst parameter does not require additional tuning for a single IP Videoconferencing (IP/VC) stream. For multiple streams, this burst parameter may be increased as required.

Streaming Video

When addressing the QoS needs of Streaming Video traffic, the following guidelines are recommended:

- **Streaming Video** (whether unicast or multicast) should be marked to **DSCP CS4** as designated by the QoS Baseline.
- **Loss** should be no more than **5 %**.
- **Latency** should be no more than **4–5 seconds** (depending on video application buffering 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** and, therefore, Branch routers may not require provisioning for Streaming Video traffic on their WAN/VPN edges (in the direction of Branch-to-Campus).

- Non-organizational Streaming Video applications, such as entertainment videos, may be marked as Scavenger (DSCP CS1) and assigned a minimal bandwidth (CBWFQ) percentage. For more information, see Scavenger-class QoS DoS/Worm Mitigation Strategy.

Streaming Video applications have more lenient QoS requirements because they are delay-insensitive (the video can take several seconds to cue-up) and are largely jitter-insensitive (due to application buffering). However, Streaming Video may contain valuable content, such as e-learning applications or multicast company meetings, and therefore may require service guarantees.

The QoS Baseline recommendation for Streaming Video marking is DSCP CS4.

An interesting consideration with respect to Streaming Video comes into play when designing WAN/VPN edge policies on Branch routers: because Streaming Video is generally unidirectional, a separate class would likely not be needed for this traffic class in the Branch-to-Campus direction of traffic flow.

Non-organizational video content (or video that is strictly entertainment-oriented in nature such as movies, music videos, humorous commercials, and so on) might be considered for a ("less-than-Best-Effort") Scavenger service. This means that these streams play if bandwidth exists, but they are the first to be dropped during periods of congestion.

QoS Requirements of Data Applications

This section includes the following topics:

- Best Effort Data
- Bulk Data
- Transactional/Interactive Data
- Locally-Defined Mission-Critical Data

There are hundreds of thousands of data networking applications. Some are TCP, others are UDP; some are delay sensitive, others are not; some are bursty in nature, others are steady; some are lightweight, others require high bandwidth, and so on. Not only do applications vary one from another, but even the same application can vary significantly in one *version* to another.

Given this, how best to provision QoS for data is a daunting question. The Cisco QoS Baseline identifies four main classes of data traffic, according to their general networking characteristics and requirements. These classes are Best Effort, Bulk Data, Transactional/Interactive Data and Locally-Defined Mission-Critical Data.

Best Effort Data

The Best Effort class is the default class for all data traffic. An application will be removed from the default class only if it has been selected for preferential or deferential treatment.

When addressing the QoS needs of Best Effort data traffic, Cisco recommends the following guidelines:

- **Best Effort** traffic should be marked to **DSCP 0**.
- Adequate bandwidth should be assigned to the Best Effort class as a whole, because the majority of applications will default to this class; reserve **at least 25 percent for Best Effort traffic**.

In 2003, a Wall Street financial company did an extensive study to identify and categorize the number of different applications on their networks. They found over 3000 discrete applications traversing their infrastructure. Further research has shown that this is not uncommon for larger enterprises.

Because enterprises have several hundred, if not thousands, of data applications running over their networks (of which, the majority will default to the Best Effort class), you need to provision adequate bandwidth for the default class as a whole, to handle the sheer volume of applications that will be included in it. Otherwise, applications defaulting to this class will be easily drowned out, which typically results in an increased number of calls to the networking help desk from frustrated users. Cisco therefore recommends that you reserve at least 25 percent of link bandwidth for the default Best Effort class.

Bulk Data

The Bulk Data class is intended for applications that are relatively non-interactive and drop-insensitive and that typically span their operations over a long period of time as background occurrences. Such applications include the following:

- FTP
- E-mail
- Backup operations
- Database synchronizing or replicating operations
- Content distribution
- Any other type of background operation

When addressing the QoS needs of Bulk Data traffic, Cisco recommends the following guidelines:

- **Bulk Data** traffic should be marked to **DSCP AF11**; excess Bulk Data traffic can be marked down by a policer to AF12; violating bulk data traffic may be marked down further to AF13 (or dropped).
- Bulk Data traffic should have a **moderate bandwidth guarantee**, but should be **constrained** from dominating a link.

The advantage of provisioning moderate bandwidth guarantees to Bulk Data applications rather than applying policers to them is that Bulk applications can dynamically take advantage of unused bandwidth and thus speed up their operations during non-peak periods. This in turn reduces the likelihood of their bleeding into busy periods and absorbing inordinate amounts of bandwidth for their time-insensitive operations.

Transactional/Interactive Data

The Transactional/Interactive Data class, also referred to simply as Transactional Data, is a combination to two similar types of applications: Transactional Data client-server applications and Interactive Messaging applications.

The response time requirement separates Transactional Data client-server applications from generic client-server applications. For example, with Transactional Data client-server applications such as SAP, PeopleSoft, and Data Link Switching (DLSw+), the transaction is a foreground operation; the user waits for the operation to complete before proceeding.

E-mail is not considered a Transactional Data client-server application, as most e-mail operations occur in the background and users do not usually notice even several hundred millisecond delays in mailspool operations.

When addressing the QoS needs of Transactional Data traffic, Cisco recommends the following guidelines:

- **Transactional Data** traffic should be marked to **DSCP AF21**; excess Transactional Data traffic can be marked down by a policer to AF22; violating Transactional Data traffic can be marked down further to AF23 (or dropped).

- Transactional Data traffic should have an **adequate bandwidth guarantee** for the interactive, foreground operations they support.

Locally-Defined Mission-Critical Data

The Locally-Defined Mission-Critical Data class is probably the most misunderstood class specified in the QoS Baseline. Under the QoS Baseline model, all traffic classes (with the exclusion of Scavenger and Best Effort) are considered critical to the enterprise. The term "locally-defined" is used to underscore the purpose of this class, which is to provide each enterprise with a premium class of service for a select subset of their Transactional Data applications that have the highest business priority for them.

For example, an enterprise may have properly provisioned Oracle, SAP, BEA, and DLSw+ within their Transactional Data class. However, the majority of their revenue may come from SAP, and therefore they may want to give this Transactional Data application an even higher level of preference by assigning it to a dedicated class such as the Locally-Defined Mission-Critical Data class.

Because the admission criteria for this class is non-technical (being determined by business relevance and organizational objectives), the decision of which applications should be assigned to this special class can easily become an organizationally- and politically-charged debate. Cisco recommends that you assign as few applications to this class from the Transactional Data class as possible. You should also obtain executive endorsement for application assignments to the Locally-Defined Mission-Critical Data class, because the potential for QoS deployment derailment exists without such an endorsement.

For the sake of simplicity, this class will be referred to simply as Mission-Critical Data.

When addressing the QoS needs of Mission-Critical Data traffic, Cisco recommends the following guidelines:

- **Mission-Critical Data** traffic should be marked to **DSCP AF31**; excess mission-critical data traffic can then be marked down by a policer to AF32 or AF33. However, DSCP AF31 is currently being used by Cisco IP Telephony equipment as Call-Signaling, so until all Cisco IPT products mark Call-Signaling to DSCP CS3, a **temporary placeholder code point (DSCP 25)** can be used to identify Mission-Critical Data traffic.
- Mission-Critical Data traffic should have an **adequate bandwidth guarantee** for the interactive, foreground operations they support.

Table 1-4 shows some applications and the generic networking characteristics that determine for which data application class they are best suited.

Table 1-4 Data Applications by Class

Application Class	Example Applications	Application/Traffic Properties	Packet / Message Sizes
Interactive	Telnet, Citrix, Oracle Thin-Clients AOL Instant Messenger Yahoo Instant Messenger PlaceWare (Conference) Netmeeting Whiteboard	Highly-interactive applications with tight user feedback requirements.	Average message size < 100 B Max message size < 1 KB

Table 1-4 Data Applications by Class

Transactional	SAP, PeopleSoft (Vantive) Oracle—financials, Internet procurement, B2B, supply chain management, and application server Oracle 8i Database Ariba Buyer I2, Siebel, E.piphany Broadvision IBM Bus 2 Bus Microsoft SQL BEA Systems DLSw+	Transactional applications typically use a client-server protocol model. User initiated client-based queries followed by server response. Query response may consist of many messages between client and server. Query response may consist of many TCP and FTP sessions running simultaneously (for example, HTTP based applications)	Depends on application—could be anywhere from 1 KB to 50 MB
Bulk	Database syncs Network-based backups Lotus Notes, Microsoft Outlook E-mail download (SMTP, POP3, IMAP, Exchange) Video content distribution, Large ftp file transfers	Long file transfers Always invokes TCP congestion management	Average message size 64 KB or greater
Best-Effort	All non-critical traffic HTTP Web browsing + other miscellaneous traffic		

QoS Requirements of the Control Plane

This section includes the following topics:

- IP Routing
- Network Management

Unless the network is up and running, QoS is irrelevant. Therefore, it is critical to provision QoS for control plane traffic, which includes IP Routing traffic and Network Management.

IP Routing

By default, Cisco IOS software (in accordance with RFC 791 and RFC 2474) marks *Interior* Gateway Protocol (IGP) traffic such as Routing Information Protocol (RIP/RIPv2), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP) to DSCP CS6. However, Cisco IOS software also has an internal mechanism for granting internal priority to important control datagrams as they are processed within the router. This mechanism is called PAK_PRIORITY.

As datagrams are processed though the router and down to the interfaces, they are internally encapsulated with a small packet header, referred to as the PAKTYPE structure. Within the fields of this internal header there is a PAK_PRIORITY flag that indicates the relative importance of control packets to the internal processing systems of the router. PAK_PRIORITY designation is a critical internal Cisco IOS software operation and, as such, is not administratively configurable in any way.

Note that *Exterior* Gateway Protocol (EGP) traffic such as Border Gateway Protocol (BGP) traffic is marked by default to DSCP CS6 but does not receive such PAK_PRIORITY preferential treatment and may need to be explicitly protected in order to maintain peering sessions.

When addressing the QoS needs of IP Routing traffic, Cisco recommends the following guidelines:

- **IP Routing** traffic should be marked to **DSCP CS6**; this is default behavior on Cisco IOS platforms.
- IGPs are usually adequately protected with the Cisco IOS internal PAK_Priority mechanism; Cisco recommends that EGPs such as BGP have an **explicit class for IP routing with a minimal bandwidth guarantee**.
- Cisco IOS automatically marks IP Routing traffic to DSCP CS6.

Additional information on PAK_PRIORITY can be found at:
http://www.cisco.com/warp/public/105/rtgupdates.html

Network Management

When addressing the QoS needs of Network Management traffic, Cisco recommends the following guidelines:

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

Network management traffic is important to perform trend and capacity analysis and troubleshooting. Therefore, you can provision a separate minimal bandwidth queue for Network Management traffic, which could include SNMP, NTP, Syslog, NFS and other management applications.

QoS Requirements of the Scavenger Class

The Scavenger class, based on an Internet-II draft, 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 organizational objectives of the enterprise and are typically entertainment-oriented. These include: Peer-to-Peer (P2P) media-sharing applications (such as KaZaa, Morpheus, Grokster, Napster, iMesh, and so on), gaming applications (Doom, Quake, Unreal Tournament, and so on), and any entertainment video applications.

Assigning a minimal bandwidth queue to Scavenger traffic forces it to be squelched to virtually nothing during periods of congestion, but allows it to be available if bandwidth is not being used for business purposes, such as might occur during off-peak hours. This allows for a flexible, non-stringent policy control of non-business applications.

When provisioning for Scavenger traffic, Cisco recommends the following guidelines:

- **Scavenger** traffic should be marked to **DSCP CS1**.
- Scavenger traffic should be assigned the **lowest configurable queuing service**; for instance, in Cisco IOS this would mean assigning a **CBWFQ of 1 %** to Scavenger.

The Scavenger class is a critical component to the DoS/worm mitigation strategy presented later in this document.

3) Designing the QoS Policies

Once a QoS strategy has been defined and the application requirements are understood, end-to-end QoS policies can be designed for each device and interface, as determined by its role in the network infrastructure. A separate QoS design document delves into the specific details of LAN, WAN, and VPN (both MPLS and IPSec VPN) QoS designs. Because the Cisco QoS toolset provides many QoS design and deployment options, a few succinct design principles can help simplify strategic QoS deployments.

For example, one such design principle is to ***always enable QoS policies in hardware— rather than software—whenever a choice exists.*** Cisco IOS routers perform QoS in software, which places incremental loads on the CPU, depending on the complexity and functionality of the policy. Cisco Catalyst switches, on the other hand, perform QoS in dedicated hardware ASICS and as such do not tax their main CPUs to administer QoS policies. This allows complex policies to be applied at line rates at even Gigabit or Ten-Gigabit speeds.

Other simplifying best-practice QoS design principles include:

- Classification and Marking Principles
- Policing and Markdown Principles
- Queueing and Dropping Principles

Classification and Marking Principles

When classifying and marking traffic, an unofficial Differentiated Services design principle is to ***classify and mark applications as close to their sources as technically and administratively feasible***. This principle promotes end-to-end Differentiated Services and PHBs. Do not trust markings that can be set by users on their PCs or other similar devices, because users can easily abuse provisioned QoS policies if permitted to mark their own traffic. For example, if DSCP EF received priority services throughout the enterprise, a PC can be easily configured to mark all the traffic of the user to DSCP EF, thus hijacking network priority queues to service non-realtime traffic. Such abuse could easily ruin the service quality of realtime applications like VoIP throughout the enterprise.

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**.** Layer 2 markings are lost when media changes (such as a LAN-to-WAN/VPN edge). There is also less marking granularity at Layer 2. For example, 802.1Q/p CoS supports only 3 bits (values 0–7), as does MPLS EXP. Therefore, only up to 8 classes of traffic can be supported at Layer 2, and inter-class relative priority (such as RFC 2597 Assured Forwarding Drop Preference markdown) is not supported. On the other hand, Layer 3 DSCP markings allow for up to 64 classes of traffic, which is more than enough for most enterprise requirements for the foreseeable future.

As the line between enterprises and service providers continues to blur and the need for interoperability and complementary QoS markings is critical, you should ***follow standards-based DSCP PHB markings to ensure interoperability and future expansion***. Because the QoS Baseline marking recommendations are standards-based, enterprises can easily adopt these markings to interface with service provider classes of service. Network mergers—whether the result of acquisitions, mergers or strategic-alliances—are also easier to manage when you use standards-based DSCP markings.

Policing and Markdown Principles

There is little reason to forward unwanted traffic only to police and drop it at a subsequent node, especially when the unwanted traffic is the result of DoS or worm attacks. The overwhelming volume of traffic that such attacks can create can cause network outages by driving network device processors to their maximum levels. Therefore, you should ***police traffic flows as close to their sources as possible***.

This principle applies also to legitimate flows. DoS/worm-generated traffic can masquerade under legitimate, well-known TCP/UDP ports and cause extreme amounts of traffic to be poured onto the network infrastructure. Such excesses should be monitored at the source and marked down appropriately.

Whenever supported, markdown should be done according to standards-based rules, such as RFC 2597 ("Assured Forwarding PHB Group"). For example, excess traffic marked to AF*x*1 should be marked down to AF*x*2 (or AF*x*3, whenever dual-rate policing—such as defined in RFC 2698—is supported). Following such markdowns, congestion management policies, such as DSCP-based WRED, should be configured to drop AFx3 more aggressively than AFx2, which in turn should be dropped more aggressively than AFx1.

However, Cisco Catalyst switches do not currently perform DSCP-Based WRED, and so this standards-based strategy cannot be implemented fully at this time. As an alternative workaround, single-rate policers can be configured to markdown excess traffic to DSCP CS1 (Scavenger); dual-rate policers can be configured to mark down excess traffic to AF*x*2, while marking down violating traffic to DSCP CS1. Traffic marked as Scavenger would then be assigned to a "less-than-Best-Effort" queue. Such workarounds yield an overall effect similar to the standards-based policing model. However, when DSCP-based WRED is supported on all routing and switching platforms, then you should mark down Assured Forwarding classes by RFC 2597 rules to comply more closely with this standard.

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 this may occur. This principle applies not only to Campus-to-WAN/VPN edges, where speed mismatches are most pronounced, but also to Campus Access-to-Distribution or Distribution-to-Core links, where oversubscription ratios create the potential for congestion. There is simply no other way to guarantee service levels than by enabling queuing wherever a speed mismatch exists.

When provisioning queuing, some best practice rules of thumb also apply. For example, as discussed previously, the Best Effort class is the default class for all data traffic. Only if an application has been selected for preferential/deferential treatment is it removed from the default class. Because many enterprises have several hundred, if not thousands, of data applications running over their networks, you must provision adequate bandwidth for this class as a whole to handle the sheer volume of applications that default to it. Therefore, it is recommended that you ***reserve at least 25 percent of link bandwidth for the default Best Effort class.***

Not only does the Best Effort class of traffic require special bandwidth provisioning consideration, so does the highest class of traffic, sometimes referred to as the "Realtime" or "Strict Priority" class (which corresponds to RFC 3246 "An Expedited Forwarding Per-Hop Behavior"). The amount of bandwidth assigned to the Realtime queuing class is variable. However, if you assign too much traffic for strict priority queuing, then the overall effect is a dampening of QoS functionality for non-realtime applications. Remember: the goal of convergence is to enable voice, video, and data to *transparently* co-exist on a single network. When Realtime applications such as Voice or Interactive-Video dominate a link (especially a WAN/VPN link), then data applications will fluctuate significantly in their response times, destroying the transparency of the converged network.

Cisco Technical Marketing testing has shown a significant decrease in data application response times when realtime traffic exceeds one-third of link bandwidth capacity. Extensive testing and customer deployments have shown that a general best queuing practice is to ***limit the amount of strict priority queuing to 33 percent of link capacity***. This strict priority queuing rule is a conservative and safe design ratio for merging realtime applications with data applications.

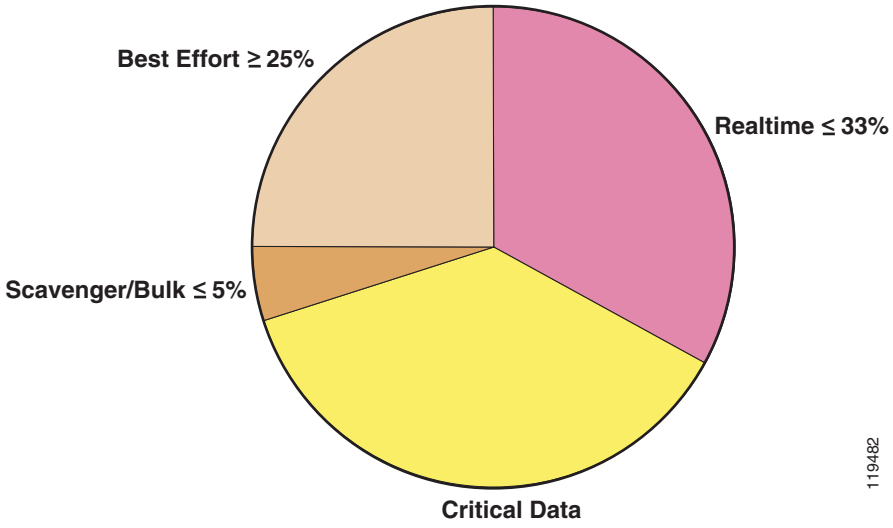
Cisco IOS software allows the abstraction (and thus configuration) of multiple strict priority LLQs. In such a multiple LLQ context, this design principle would *apply to the sum of all LLQs to be within one-third of link capacity.*

Note This strict priority queuing rule (limit to 33 percent) is simply a best practice design recommendation and is not a mandate. There may be cases where specific business objectives cannot be met while holding to this recommendation. In such cases, enterprises must provision according to their detailed requirements and constraints. However, it is important to recognize the tradeoffs involved with over-provisioning strict priority traffic and its negative performance impact on non-realtime-application response times.

Whenever a Scavenger queuing class is enabled, it should be assigned a minimal amount of bandwidth. On some platforms, queuing distinctions between Bulk Data and Scavenger traffic flows cannot be made because queuing assignments are determined by CoS values and these applications share the same CoS value of 1. In such cases you can assign the Scavenger/Bulk queuing class a bandwidth percentage of 5. If you can uniquely assign Scavenger and Bulk Data to different queues, then you should assign the Scavenger queue a bandwidth percentage of 1.

The Realtime, Best Effort and Scavenger queuing best practice principles are shown in Figure 1-7.

Figure 1-7 Realtime, Best Effort and Scavenger Queuing Rules



Because platforms support a variety of queuing structures, configure consistent queuing policies according to platform capabilities to ensure consistent PHBs.

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

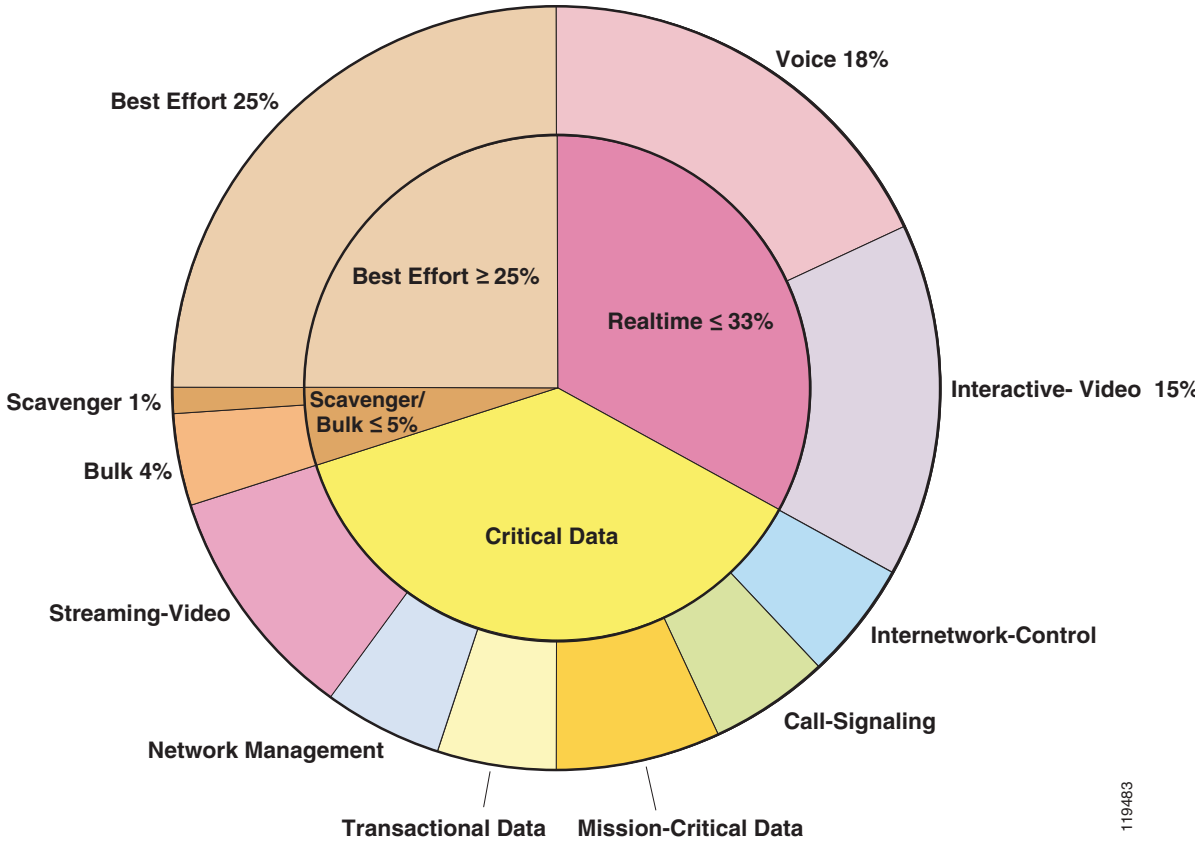
- Realtime (33%)
- Critical Data
- Best Effort Data (25%)
- Scavenger/Bulk (5%)

The queuing policies can be expanded on a platform that supports a full 11-class QoS Baseline queuing model in such a way as to provide consistent servicing to Realtime, Best Effort and Scavenger traffic. For example, on a platform such as a Cisco IOS router that supports 11 queues with DSCP-based admission, an advanced queuing policy could be as follows:

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

The inter-relationship between these compatible queuing models is shown in Figure 1-8.

Figure 1-8 Compatible Four-Class and Eleven-Class Queuing Models following Realtime, Best Effort and Scavenger Queuing Rules



In this way, traffic receives compatible queuing at each node, regardless of platform capabilities, which is the overall objective of DiffServ PHB definitions.

Whenever supported, you should ***enable WRED (preferably DSCP-based WRED)*** on all TCP flows. WRED congestion avoidance prevents TCP global synchronization and increases overall throughput and link efficiency. Enabling WRED on UDP flows is optional.

These and other architecture-specific QoS design best-practices are discussed in more detail in a separate QoS design document, along with the configuration examples.

Furthermore, it is highly-recommended to schedule Proof-of-Concept (PoC) tests to verify that the hardware/software platforms in production support the required QoS features *in combination* with all the other features they are currently running. Remember, ***in theory, theory and practice are the same***. In other words, ***there is no substitute for testing***.

4) Rolling out the QoS Policies

Once the QoS designs have been finalized and PoC tested, it is vital to ensure that the networking team ***thoroughly understand the QoS features and syntax before enabling features on production networks.*** Such knowledge is critical for both rollout and subsequent troubleshooting of QoS-related issues.

Furthermore, it is recommended to ***schedule network downtime in order to rollout QoS*** features. While QoS is required end-to-end, it does not have to be deployed end-to-end at a single instance. A pilot network-segment can be selected for an initial deployment, and pending observation, the ***rollout can be expanded in stages*** to encompass the entire enterprise.

A rollback strategy is always recommended, to address unexpected issues arising from the QoS deployment.

5) Monitoring the Service-Levels

Implementing a QoS solution is not a one-time task that is complete upon policy deployment. A successful QoS policy ***rollout is followed by ongoing monitoring of service levels and periodic adjustments and tuning*** of QoS policies.

Short-term monitoring is useful for verifying that the deployed QoS policies are having the desired end-to-end effect. *Long-term monitoring* (trending) is needed to determine whether the provisioned bandwidth is still adequate for the changing needs of the enterprise. For example, upgrading to a newer version of an application may cause the provisioned bandwidth to be exceeded, as would the addition of new users. Furthermore, business objectives or economic climates themselves may change, and periodically the overall ranking of priority of applications may need revision.

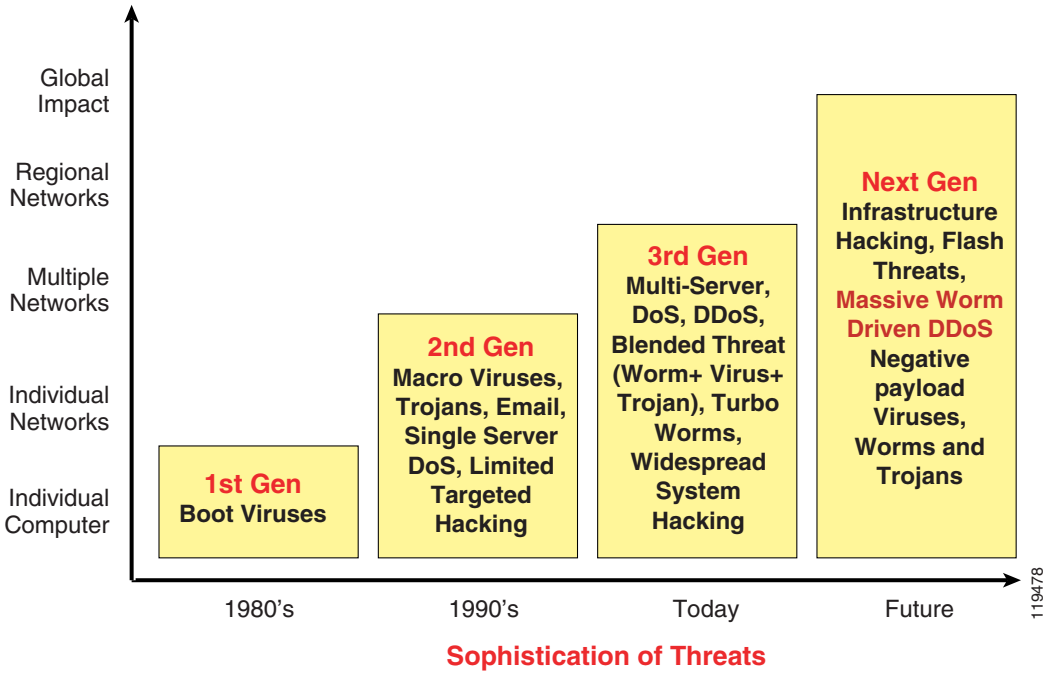
As business conditions change, the enterprise may need to adapt to these changes and ***may be required to begin the QoS deployment cycle anew***, by redefining their objectives, tuning and testing corresponding designs, rolling these new designs out and monitoring them to see if they match the redefined objectives.

How Can I Use QoS Tools to Mitigate DoS/Worm Attacks?

Whenever the business objectives of the enterprise includes mitigating DoS/worm attacks, the Scavenger-class QoS strategy and best practices described in this section apply.

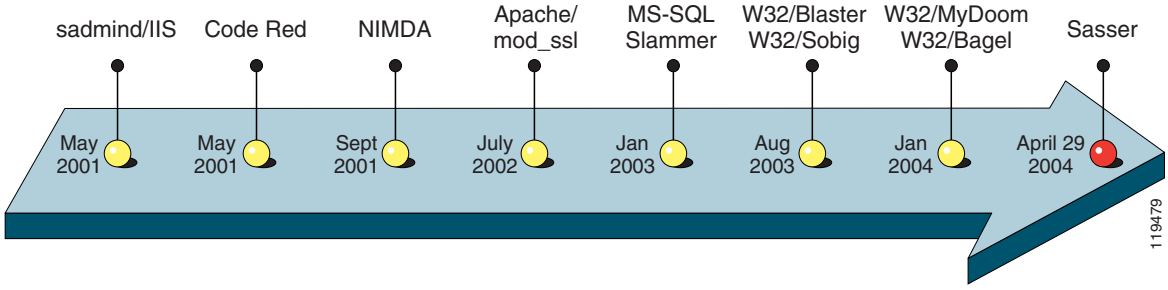
Worms have existed in one form or another since the beginning of the Internet, and have steadily increased in complexity and scope of damage, as shown in Figure 1-9.

Figure 1-9 Business Security Threat Evolution



There has been an exponential increase since 2001 in not only the frequency of DoS/worm attacks, but also in their relative sophistication. For example, more than 994 new Win32 viruses and worms were documented in the first half of 2003, more than double the 445 documented in the first half of 2002. Some of these more recent worms are shown in Figure 1-10.

Figure 1-10 Recent Internet Worms



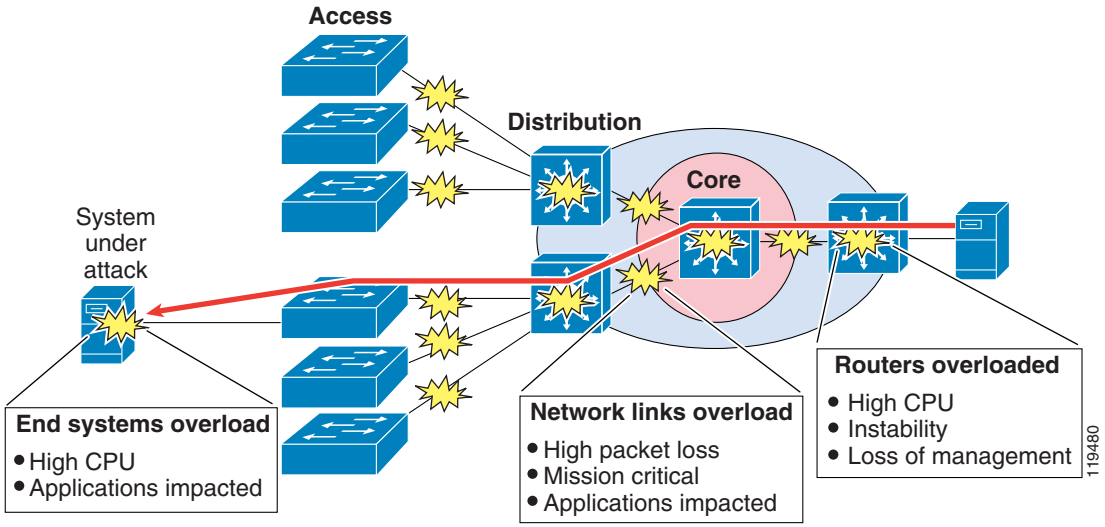
There are two main classes of DoS attacks:

- Spoofing attacks—The attacker pretends to provide a legitimate service, but provides false information to the requester (if any).
- Slamming/flooding attacks—The attacker exponentially generates and propagates traffic until service resources (servers and/or network infrastructure) are overwhelmed.

Spoofing attacks are best addressed by authentication and encryption technologies. Slamming/flooding attacks, on the other hand, can be effectively mitigated through QoS technologies.

Worms, on the other hand, exploit security vulnerabilities in their targets and disguisedly carry harmful payloads that usually include a self-propagating mechanism. Network infrastructure usually isn't the direct target of a worm attack, but can become collateral damage as worms exponentially self-propagate. The rapidly multiplying volume of traffic flows eventually drowns the CPU/hardware resources of routers and switches in their paths, indirectly causing Denial of Service to legitimate traffic flows, as shown in Figure 1-11.

Figure 1-11 Direct and Indirect Collateral Damage from DoS/Worm Attacks



A *reactive approach* to mitigating such attacks is to reverse-engineer the worm and set up intrusion detection mechanisms and/or ACLs and/or NBAR policies to limit its propagation. However, the increased sophistication and complexity of worms make them harder and harder to separate from legitimate traffic flows. This exacerbates the finite time lag between when a worm begins to propagate and when the following can take place:

Sufficient analysis has been performed to understand how the worm operates and what its network characteristics are.

An appropriate patch, plug or ACL is disseminated to network devices that may be in the path of worm; this task may be hampered by the attack itself, as network devices may become unreachable for administration during the attacks.

These time lags may not seem long in absolute terms, such as in minutes, but the relative window of opportunity for damage is huge. For example, in 2003, the number of hosts infected with the Slammer worm (a Sapphire worm variant) doubled every 8.5 seconds on average, infecting over 75,000 hosts in just 11 minutes and performing scans of 55 million more hosts within the same time period.

Note Interestingly, a 2002 CSI/FBI report stated that the majority of network attacks occur from *within* an organization, typically by disgruntled employees. This underscores the need to protect the Access-Edges of enterprise networks as well as their Internet edges.

A *proactive approach* to mitigating DoS/worm flooding attacks within enterprise networks is to immediately respond to out-of-profile network behavior indicative of a DoS or worm attack using Campus Access-Layer policers. Such policers meter traffic rates received from endpoint devices and markdown excess traffic spikes to the Scavenger class (DSCP CS1) when these exceed specified watermarks (at which point they are no longer considered normal flows).

In this respect, the policers are relatively dumb. They do not match specific network characteristics of specific types of attacks, but simply meter traffic volumes and respond to abnormally high volumes as close to the source as possible. The simplicity of this approach negates the need for the policers to be programmed with knowledge of the specific details of how the attack is being generated or propagated.

It is precisely this dumbness of such access layer policers that allow them to maintain relevancy as worms mutate and become more complex. The policers do not care *how* the traffic was generated or *what* it looks like, they care only *how much* traffic is being put onto the wire. Therefore, they continue to police even advanced worms that continually change the tactics of how traffic is being generated.

For example, in most enterprises it is quite abnormal (within a 95 % statistical confidence interval) for PCs to generate sustained traffic in excess of 5 % of link capacity. In the case of a FastEthernet switch port, this means that it would be unusual in most organizations for an end-user PC to generate more than 5 Mbps of *uplink* traffic on a sustained basis.

Note It is important to recognize that this value (5 percent) for normal Access-Edge utilization by endpoints is just an example value. This value would likely vary within the enterprise and from enterprise to enterprise.

It is very important to recognize that what is being proposed is not to police all traffic to 5 Mbps and automatically drop the excess. Should that be the case, there would not be much reason to deploy FastEthernet or GigabitEthernet switch ports to endpoint devices, because even 10-BaseT Ethernet switch ports have more uplink capacity than a 5 Mbps policer-enforced limit. Furthermore, such an approach would supremely penalize legitimate traffic that did happen to exceed 5 Mbps on an FE switch port.

A less draconian approach would be to couple Access Layer policers with hardware/software (Campus/WAN/VPN) queuing polices, with both sets of policies provisioning a “less-than-Best-Effort” Scavenger class. Access Layer policers would markdown out-of-profile traffic to DSCP CS1 (Scavenger) and then have all congestion management policies (whether in Catalyst hardware or in Cisco IOS software) provision a “less-than Best-Effort” queueing service for any traffic marked to DSCP CS1.

Let’s illustrate how this might work for both *legitimate* traffic exceeding the Access Layer’s policer watermark and also the case of *illegitimate* excess traffic resulting from a DoS or worm attack.

In the former case, assume that the PC generates over 5 Mbps of traffic, perhaps because of a large file transfer or backup. Congestion (under normal operating conditions) is rarely if ever experienced within the Campus because there is generally abundant capacity to carry the traffic. Uplinks to the Distribution and Core layers of the Campus network are typically GigabitEthernet and would require 1000 Mbps of traffic from the Access Layer switch to congest. If the traffic is destined to the far side of a WAN/VPN link (which is rarely over 5 Mbps in speed), dropping occurs even without the Access Layer policer, because of the bottleneck caused by the Campus/WAN speed mismatch. The TCP sliding windows mechanism would eventually find an optimal speed (under 5 Mbps) for the file transfer. Access Layer policers that markdown out-of-profile traffic to Scavenger (CS1) would thus not affect legitimate traffic, aside from the obvious remarking. *No reordering or dropping would occur on such flows as a result of these policers that would not have occurred anyway.*

In the latter case, the effect of Access Layer policers on traffic caused by DoS or worm attacks is quite different. As hosts become infected and traffic volumes multiply, congestion may be experienced even within the Campus. If just 11 end-user PCs on a single switch begin spawning worm flows to their maximum FastEthernet link capacities, the GigabitEthernet uplink from the Access Layer switch to the Distribution Layer switch will congest and queuing/reordering/dropping will engage. At this point, VoIP, critical data applications, and even Best Effort applications would gain priority over worm-generated traffic (as Scavenger traffic would be dropped the most aggressively). Furthermore, network devices would remain accessible for administration of the patches/plugs/ACLs/NBAR-policies required to fully

neutralize the specific attack. WAN/VPN links would also be protected: VoIP, critical data and even Best Effort flows would receive priority over any WAN/VPN traffic marked down to Scavenger/CS1. This is a huge advantage, because generally WAN/VPN links are the first to be overwhelmed by DoS/worm attacks. Scavenger-class Access Layer policers thus *significantly mitigate network traffic generated by DoS or worm attacks.*

It is important to recognize the distinction between mitigating an attack and preventing it entirely. The strategy described in this document ***does not guarantee that no DoS or worm attacks will ever happen, but serves only to reduce the risk and impact that such attacks could have on the network infrastructure***.

Scavenger-class QoS DoS/Worm Mitigation Strategy

Let's recap the most important elements of the Scavenger-class QoS DoS/worm mitigation strategy.

First, network administrators need to ***profile applications to determine what constitutes normal as opposed to abnormal flows, within a 95 percent confidence interval***. Thresholds demarking normal/abnormal flows will vary from enterprise to enterprise and from application to application. Beware of over-scrutinizing traffic behavior because this could exhaust time and resources and could easily change daily. Remember, legitimate traffic flows that temporarily exceed thresholds are not penalized by the presented Scavenger- class QoS strategy. Only sustained, abnormal streams generated simultaneously by multiple hosts (highly-indicative of DoS/worm attacks) are subject to aggressive dropping only *after* legitimate traffic has been serviced.

To contain such abnormal flows, ***deploy Campus Access-Edge policers to remark abnormal traffic to Scavenger (DSCP CS1)***. Additionally, whenever Cisco Catalyst 6500s with Supervisor 720s are deployed in the distribution layer, ***deploy a second line of policing-defense at the distribution layer via Per-User Microflow Policing***.

To complement these remarking policies, it is necessary to ***enforce end-to-end "less-than-Best-Effort" Scavenger-class queuing policies*** within the Campus, WAN and VPN.

It is critically important to recognize, that even when Scavenger class QoS has been deployed end-to-end, this strategy only *mitigates* DoS/worm attacks, and *does not prevent them or remove them entirely.* Therefore, it is vital to overlay security, firewall, intrusion detection, identity, Cisco Guard, Cisco Traffic Anomaly Detector and Cisco Security Agent solutions in addition to QoS-enabled infrastructures.

Summary

This document began by reviewing **Why Quality of Service is important to Enterprise networks**, specifically because as it enables the transparent convergence of voice, video and data onto a single network. Furthermore, this proven technology can be used to mitigate the impact of DoS/worm attacks.

To set a context for discussion, the QoS toolset was reviewed, including classification and marking tools, policing tools, scheduling tools, link specific tools and AutoQoS.

The next section examined **How is QoS is optimally deployed within the enterprise?** The answer consisted of a 5 phase approach:

1) Strategically defining the objectives to be achieved via QoS—Successful QoS deployments begin by clearly defining organizational QoS objectives and then selecting an appropriate number of service classes to meet these objectives. This section introduced Cisco's QoS Baseline as a strategic guide for selecting the number and type of traffic classes to meet organizational objectives; also a migration

strategy was presented to illustrate how enterprises could start with simple QoS models and gradually increase their complexity as future needs arose. Executive endorsement is recommended, especially when choosing the select few applications to be serviced by the Mission-Critical data class.

2) Analyzing the service-level requirements of the various traffic classes to be provisioned for—the service level needs of voice, video, data and the control plane were discussed. Some of these highlights include the following:

Voice requires 150 ms one-way, end-to-end (mouth-to-ear) delay, 30 ms of one-way jitter and no more than 1 % packet loss. Voice should receive strict priority servicing, and the amount of priority bandwidth assigned for it should take into account the VoIP codec, the packetization rate, IP/UDP/RTP headers (compressed or not) and Layer 2 overhead. Additionally, provisioning QoS for IP telephony requires that a minimal amount of guaranteed bandwidth be allocated to Call-Signaling traffic.

Video comes in two flavors: Interactive Video and Streaming Video. Interactive Video has the same service level requirements as VoIP because a voice call is embedded within the video stream. Streaming Video has much laxer requirements, because of the high amount of buffering that has been built into the applications.

Control plane requirements, such as provisioning moderate bandwidth guarantees for IP Routing and Network Management protocols, should not be overlooked.

Data comes in a variety of forms, but can generally be classified into four main classes: Best Effort (the default class), Bulk (non-interactive, background flows), Transactional/Interactive (interactive, foreground flows) and Mission-Critical. Mission-Critical Data applications are locally-defined, meaning that each organization must determine the select few Transactional Data applications that contribute the most significantly to their overall business objectives.

3) Designing and testing QoS policies prior to production-network deployment—several best-practice QoS design principles were presented to help simplify and streamline the QoS design phase. These included: always performing QoS policies in hardware – rather than software – whenever a choice exists, classifying and marking (with standards-based DSCP markings) as close to the source as technically and administratively feasible, policing as close to the source as possible, and queuing on every node that has a potential for congestion. Queuing guidelines also included not provisioning more than 33% of a link for realtime traffic and reserving at least 25% of a link for the default Best Effort class.

4) Rolling-out the tested QoS designs to the production-network – Once the QoS designs have been finalized and PoC tested, it is vital ensure that the networking team thoroughly understands the QoS features and syntax before enabling features on production networks. Furthermore, it is recommended to schedule network downtime in order to rollout QoS features. A pilot network-segment can be selected for an initial deployment, and pending observation, the rollout can be expanded in stages to encompass the entire enterprise. A rollback strategy is always recommended, to address unexpected issues arising from the QoS deployment.

5) Monitoring service levels to ensure that the QoS objectives are being met – Implementing a QoS solution is not a one-time task that is complete upon policy deployment. A successful QoS policy rollout is followed by ongoing monitoring of service levels and periodic adjustments and tuning of QoS policies.

As business conditions change, the enterprise may need to adapt to these changes and may be required to begin the QoS deployment cycle anew, by redefining their objectives, tuning and testing corresponding designs, rolling these new designs out and monitoring them to see if they match the redefined objectives.

The document concluded by addressing the highly-relevant question: **How can QoS tools be used to mitigate DoS/Worm Attacks?**

A “less-than-Best-Effort” traffic class, called Scavenger, was introduced, and a strategy for using this class for DoS/worm mitigation was presented. Specifically, flows can be monitored and policed at the Campus Access-Edge (and also at the Distribution Layer if Catalyst 6500s with Supervisor 720s are

used). Out-of-profile flows can be marked down to the Scavenger marking (of DSCP CS1). To complement these policers, queues providing a “less-than-Best-Effort” Scavenger service during periods of congestion can be deployed in the LAN, WAN and VPN. Such a strategy would not penalize legitimate traffic flows that were temporarily out of profile; however sustained abnormal streams, highly-indicative of DoS/worm attacks, would be subject to aggressive dropping only after legitimate traffic was fully serviced.

It is critically important to recognize, that even when Scavenger-class QoS has been deployed end-to-end, this strategy only mitigates DoS/worm attacks, and does not prevent them or remove them entirely. Therefore, it is vital to overlay security, firewall, intrusion detection, identity, Cisco Guard, Cisco Traffic Anomaly Detctor and Cisco Security Agent solutions in addition to QoS-enabled infrastructures.

References

Standards

- RFC 791 “Internet Protocol Protocol Specification”
  http://www.ietf.org/rfc/rfc791
- RFC 2474 "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
  http://www.ietf.org/rfc/rfc2474
- RFC 2597 "Assured Forwarding PHB Group"
  http://www.ietf.org/rfc/rfc2597
- RFC 2697 "A Single Rate Three Color Marker"
  http://www.ietf.org/rfc/rfc2697
- RFC 2698 "A Two Rate Three Color Marker"
  http://www.ietf.org/rfc/rfc2698
- RFC 3168 "The Addition of Explicit Congestion Notification (ECN) to IP"
  http://www.ietf.org/rfc/rfc3168
- RFC 3246 "An Expedited Forwarding PHB (Per-Hop Behavior)"
  http://www.ietf.org/rfc/rfc3246

Books

- Szigeti, Tim and Christina Hattingh. *End-to-End QoS Network Design: Quality of Service in LANs, WANs and VPNs*. Indianapolis: Cisco Press, 2004.

Cisco Documentation

- Cisco IOS QoS Configuration Guide – Cisco IOS version 12.3
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/qos_vcg.htm
- Cisco IOS Configuration Guide – Configuring Data Link Switching Plus - Cisco IOS version 12.3

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fibm_c/bcfpart2/bcfdlsw.htm

- Understanding How Routing Updates and Layer 2 Control Packets Are Queued on an Interface with a QoS Service Policy (PAK_PRIORITY)

  http://www.cisco.com/warp/public/105/rtgupdates.html