SUN BLUEPRINTS APPROVED

Enterprise Quality of Service (QoS) - Part I: Internals

By Deepak Kakadia - Enterprise Engineering

Sun BluePrints™ OnLine - February 2002

http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 816-4163-10
Revision 1.0, 12/18/01
Edition: February 2002

Enterprise Quality of Service (QoS) Part I: Internals

Enterprise customers, are realizing that as a result of deploying new emerging real-time and mission-critical applications, that the traditional “Best Effort” IP network service model is unsuitable. The main concern is that non-well behaved flows, adversely affect other flows that share the same resources. It is difficult to tune resources so that the requirements of all deployed applications are met.

Quality of Service (QoS) can be thought of as a performance and availability delivery specification of a service. QoS can usually be referred to as a measure of the ability of network and computing systems to provide different levels of services to selected applications and associated network flows. Customers that are deploying mission-critical applications and real time applications have an economic incentive to invest in QoS capabilities so that acceptable response times are guaranteed within certain tolerances.

This article, Part I of a two part series, explains QoS functional components and mechanisms and provides the reader with the technical background helpful to better understand the trade-offs between alternative QoS solutions.

Next month, Enterprise Quality of Service (QoS) Part II: Enterprise Solution focuses on Enterprise Networks detailing what corporations can do to prioritize traffic in an optimal manner to ensure that certain applications receive priority over less important applications.

The Need for QoS

In order to understand the need for QoS, let’s look at what has happened to enterprise applications over the past decade. In the late 1980’s and early 1990’s, the client server was the dominant architecture. The main principle involved a thick client and local server, where 80% of the traffic would be from the client to a local server and 20% of the client traffic would need to traverse the corporate backbone. In the late 1990’s with the rapid adoption of Internet-based applications, the architecture changed to a thin client, and servers were located anywhere and everywhere. This had one significant implication, the network became a critically shared resource, where priority traffic was dangerously impacted by nonessential traffic. A common example is the difference between downloading images versus processing sales orders. Different applications have different resource needs. This section describes why different applications have different QoS requirements and why QoS is becoming a critical resource for enterprise data centers and service providers whose customers drive the demand for QoS.

Classes of Applications

There are five classes of applications, having different network and computing requirements. They are:

- Data transfers
- Video/voice streaming
- Interactive video/voice
- Mission-critical
- Web-based

These classes are important in classifying, prioritizing, and implementing QoS. The following sections detail these five classes.

Data Transfers

Data transfers include applications such as FTP, email, and database backup. Data transfers tend to have zero tolerances for packet loss, and high tolerances for delay and jitter. Typical acceptable response times range from a few seconds for FTP transfers to hours for email. Bandwidth requirements in the order of Kbyte/sec are acceptable, depending on the file size, which keeps response times to a few seconds. Depending on the characteristics of the application, (for example, size of a file) disk I/O transfer times can contribute cumulatively to delays along with network bottlenecks.

Video/Voice Streaming

Video/voice streaming includes applications such as Apple’s QuickTime Streaming or Real Networks’ Streaming video and voice products. Video/voice streamings tend to have low tolerances for packet loss, and medium tolerances for delay and jitter. Typical acceptable response times are in the order of a few seconds. This is due to the fact that the server can pre-buffer multimedia data on the client to a certain degree. This buffer then drains at a constant rate on the client side, while simultaneously, receiving bursty streaming data from the server with variations in delay. As long as the buffer can absorb all variations (without draining empty), the client sees a constant stream of video and voice. Typical bandwidth requirements are in the order of Mbyte/sec, depending on frame rate, compression/decompression algorithms, and size of images. Disk I/O and Central Processing Unit (CPU) also contribute to delays. Large Motion Pictures Experts Group (MPEG) files must be read from disks and compression/decompression algorithms.

Interactive Video/Voice

Interactive video/voice tends to have low to medium levels of tolerance for packet loss, and low tolerance for delay and jitter. Typical bandwidth requirements are tremendous (depending on number of simultaneous participants in the conference, growing exponentially). Due to the interactive nature of the data being transferred, tolerances are very low for delay and jitter. As soon as one participant moves or talks, all other participants need to immediately see and hear this change. Response times requirements range from 250 to 500 ms. This response time is compounded by the bandwidth requirements with each stream requiring a few Mbit/sec. In a conference of five participants, each participant is pumping out their voice and video stream while at the same time receiving the other participants' streams.

Mission-Critical Applications

Mission-critical applications vary in bandwidth requirements, but tend to have zero tolerance for packet loss. Depending on the application, bandwidth requirements are in the order of Kbyte/sec. Response times are in the order of 500 ms to a few seconds. Server resource requirements vary, depending on the application (for example, in terms of CPU, disk, and memory).

Web-Based Applications

Web-based applications tend to have low bandwidth requirements, (unless large image files are associated with the request web page) and grow in CPU and disk requirements, due to dynamically generated web pages and web transaction based applications. Response time requirements range from 500 ms to 1 second.

Different classes of applications have different network and computing requirements. The challenge is to align the network and computing services to the application's service requirements from a performance perspective.

Approaches

The two most common approaches used to satisfy the service requirements of applications are:

- Over provisioning
- Managing and controlling

Over provisioning allows over allocation of resources to meet or exceed peak load requirements. Depending on the deployment, over provisioning can be viable if it is a simple matter of just upgrading to faster local area network (LAN) switches and network interface cards (NICs), adding memory, adding CPU, or disk. However, over provisioning may not be viable in certain cases, for example when dealing with relatively expensive long haul wide area network (WAN) links, resources that on average are under utilized, or source busy only during short peak periods.

Managing and controlling allows allocation of network and computing resources. Better management of existing resources attempts to optimize utilization of existing resources such as limited bandwidth, CPU cycles, and network switch buffer memory.

QoS Components

To give you enough background on the fundamentals and an implementation perspective, this section describes the overall network and systems architecture and identifies the sources of delays and a good overall understanding why QoS is essentially about controlling network and system resources in order to achieve more predictable delays for preferred applications. In this section, a generic QoS system overview is presented, describing the following high level QoS internal functional components.

Implementation Functions

The following are necessary implementation functions as well as challenges that may be experienced in practice:

1) **Traffic Rate Limiting and Traffic Shaping** - Token Leaky Bucket Algorithm. Network traffic is always bursty. The level of burstiness is controlled by the time resolution of the measurements. Rate limiting controls the burstiness of the traffic coming into a switch or server. Shaping refers to the smoothing of the egress traffic. Although these two functions are opposite, the same class of algorithms are used to implement these functions.

2) **Packet Classification** - Individual flows must be identified and classified at line rate. Fast packet classification algorithms are crucial, as every packet must be inspected and matched against a set of rules that determine the class of service that the specific packet should receive. The packet classification algorithm has serious scalability issues; as the number of rules increases it takes longer to classify a packet.

3) **Packet Scheduling** - In order to provide differentiated services, the packet scheduler needs to decide quickly which packet should be scheduled and when. The most simplest packet scheduling algorithm is strict priority, however, this often does not work as low priority packets are starved and may never get scheduled.

QoS Metrics

QoS is defined by a multitude of metrics. The simplest is bandwidth, which can be conceptually visioned as a logical (or smaller) pipe of a larger pipe. But since actual network traffic network traffic is bursty, a fixed bandwidth would be wasteful since at one instant in time one flow would perhaps use 1% of this pipe, while another customer may need 110% of his allocated pipe. To reduce waste, certain burst metrics are used to determine how much of a burst and how long a burst can be tolerated. Other important metrics that directly impact the quality of service include packet loss rate, delay and jitter (variation in delay). The network and computing components that control these metrics are described later in this article.

Network and Systems Architecture Overview

In order to fully understand where QoS fits into the overall picture of network resources, it is useful to take a look at the details of the complete network path traversal, starting from the point where a client sends a request, traverses various network devices, and finally arrives at the destination where the server processes the request.

There are different classes of applications, having different characteristics and requirements (see the Section, “The Need for QoS” for additional details). It is due to the fact that there are several federated networks combined with different traffic characteristics that makes end-to-end QoS a complex issue.

FIGURE 1 illustrates a high-level overview of the components involved in an end-to-end packet traversal of an enterprise that relies on a service provider. There are two different paths shown, both *originate* from the client and *end* at a server.

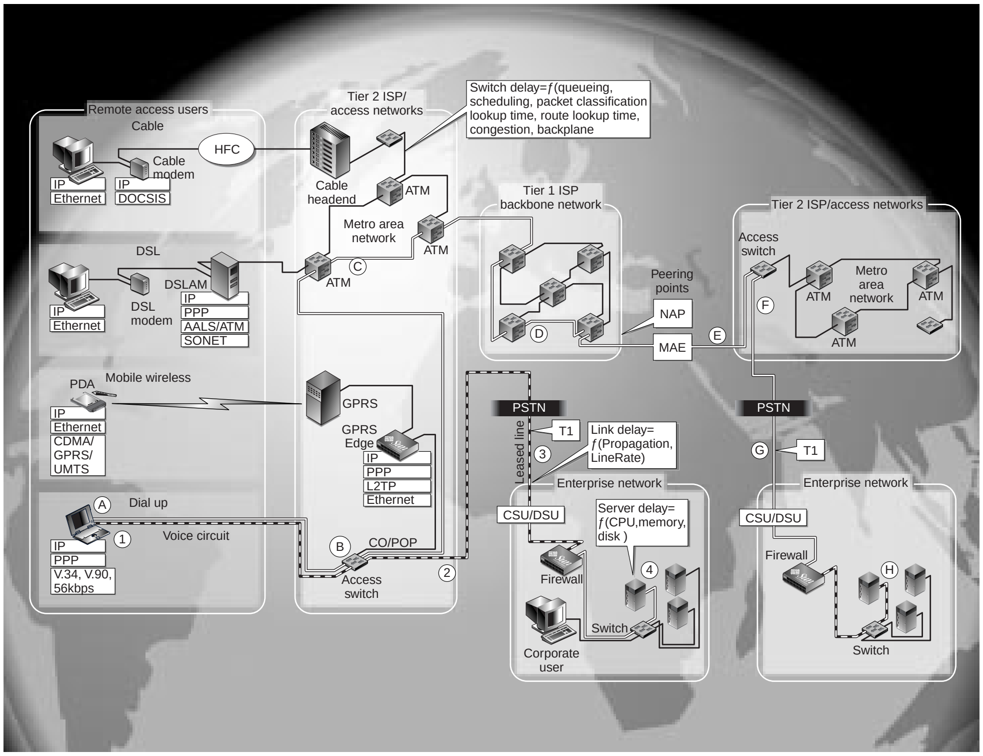

FIGURE 1 Overview of End to End Network and Systems Architecture

Path A-H is a typical scenario, where the client and servers are connected to different local Internet Service Providers (ISPs) and need to traverse different ISP networks. There can be multiple Tier 1 ISPs traversed, connected together by peering points such as Metropolitan Area Exchange (MAE)-East or private peering points such as Sprints Network Access Point (NAP).

Path 1-4 is an example of the client and server connected to the same local Tier 2 ISP, when both client and server are physically located in the same geographical area.

In either case, the majority of the delays are attributed to the switches in the Tier 2 ISP. The links from the end-user customers to the Tier 2 ISP tend to be slow links, but the Tier 2 ISP aggregates many links, hoping that not all subscribers will use the links at the same point in time. If they do, packets get buffered up and eventually get dropped.

Implementing QoS

The previous section explained the positioning of deploying a QoS capable device, which can be a network switch / router or a server. The server can implement QoS on the network interface card or in the protocol stack. In either case, between the application socket end points. This section describes how this device actually implements QoS, with a focus on network traffic.

You can implement QoS in many different ways. Each domain has control over its resources and can implement QoS on its portion of the end-to-end path using different technologies. Two particular domains of implementation are:

1. Enterprise—Enterprises can control their own networks and systems. From a local ethernet / token ring LAN perspective, IEEE 801.p, can be used to mark frames according to priorities. These marks allow the switch to offer preferential treatment to certain flows across Virtual Local Area Networks (VLANS). For computing devices, there are facilities that allow processes to run at higher priorities, thus obtaining differentiated services from a process computing perspective.

2. Network Service Provider (NSP)—The NSP, in general, aggregates traffic and forwards either within their own network or hands-off to another NSP. The NSP can use technologies such as DiffServ or IntServ, to prioritize the handling of traffic within their networks. Service Level Agreements (SLAs) are required between NSP to obtain a certain level of QoS for transit traffic.

Asychronous Transfer Mode (ATM)

This section takes a quick look at ATM from a QoS perspective. After 1995, ATM started taking off in data networks, with one of its advantages being that ATM provided QoS. NSPs implement QoS at both the IP layer and the ATM layer while most ISPs still have ATM networks, that carry IP traffic. ATM itself offers six types of QoS services. These six types are:

1. Constant Bit Rate (CBR)—Provides a constant bandwidth, delay and jitter throughout the life of the ATM connection.

2. Variable Bit Rate-Real Time (VBR-rt)—Provides constant delay and jitter, but variations in bandwidth

3. Variable Bit Rate-Non Real Time (VBR-nrt)—Provides variable bandwidth, delay and jitter, but has a low cell loss rate.

4. Unspecified Bit Rate (UBR)—Provides “Best Effort” service, no guarantees.

5. Available Bit Rate (ABR)—Provides no guarantees, expects the applications to adapt according to network availability.

6. Guaranteed Frame Rate (GFR)—Provides some minimum frame rate, delivers entire frame or none, used for ATM Adaptation Layer 5 (AAL5).

One of the main difficulties in providing an end-to-end QoS solution is that there are so many private networks that must be traversed, and each network has their own QoS implementations and business objectives. The Internet is constructed such that networks interconnect or “Peer” with other networks. One network may need to forward traffic of other networks. Depending on the arrangements, competitors may not forward the traffic in the most optimal manner. This is what is meant by business objectives.

Sources of Unpredictable Delay

From a non-real-time system computing perspective, delays that are unpredictable are often due to limited CPU resources or disk I/O latencies. These degrade during a heavy load. From a network perspective, there are many components that add up to the cumulative end-to-end delay. This section describes some of the important components that contribute to delay. The aim of this section is to explain that the choke points are at the access networks, where the traffic is aggregated and forwarded to a backbone or core. Service providers will over allocate their networks to increase profits and hope that not all subscribers will want network access at the same instant in time.

FIGURE 2 was constructed by taking out path A-G in FIGURE 1 and projecting it onto a Time-Distance plane. This is a typical web client accessing the Internet site of an enterprise. The vertical axis indicates the time that elapsed for a packet to travel a certain link segment. The horizontal axis indicates the link segment that a packet traverses. At the top, we see the network devices and vertical lines that project down to the distance axis, clearly showing the corresponding link segment. In this illustration, an IP packet’s journey starts from the point in time when a user clicks on a web page. The Hyper Text Transfer Protocol (HTTP) request maps first to a TCP three-way handshake to create a socket connection. The first TCP packet is the initial SYN packet, which first traverses segment 1 and is usually quite slow since this link is typically 30 Kbyte/sec using a 56 Kbyte/sec modem, depending on the quality and distance of the last mile wiring.

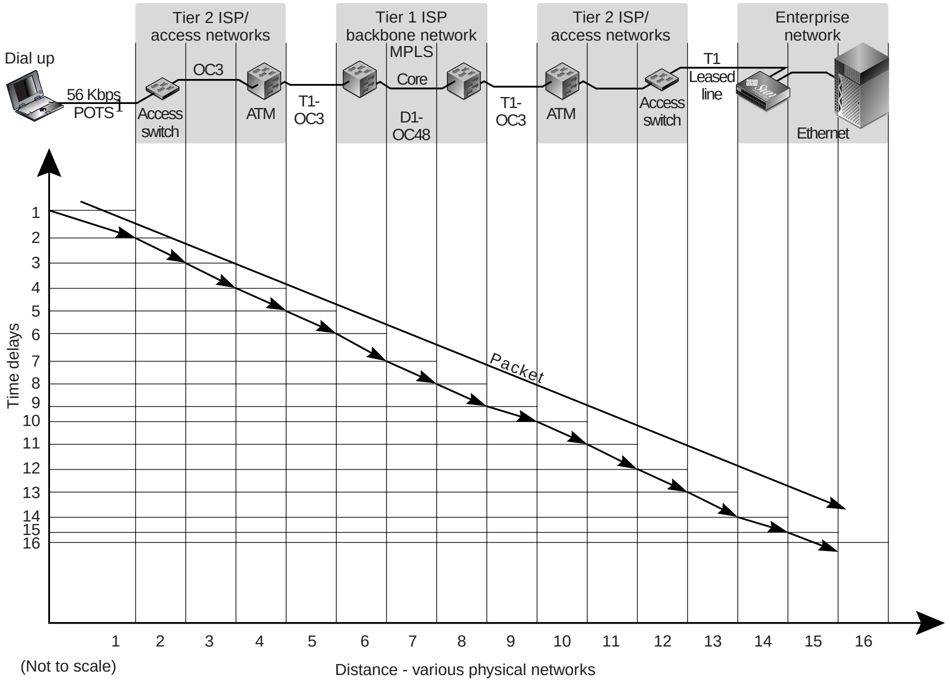

FIGURE 2 One Way End-to-End Packet Data Path Transversal

Network Delay is composed of two components:

1. Propagation delay that depends on the media and distance.
2. Line rate that primarily depends on the link rate and loss rate or Bit Error Rate (BER).

The odd number links of FIGURE 2 represent the link delays. Please note that segment and link are used interchangeably.

- Link 1, in a typical deployment, is the copper wire, or the "last mile" connection from the home or Small Office / Home Office (SOHO) to the Regional Bell Operating Company (RBOC). This is how a large portion of consumer clients connect to the Internet.
- Link 3 is an ATM link inside the Carriers internal network, usually a Metropolitan Area Network Link.
- Link 5 connects the Tier 2 ISP to the Tier 1 ISP.

  This provides a Backbone Network. This link is a larger pipe, which can range from T1 to Operating Carrier 3 (OC-3) while growing.

- Link 7 is the Core Network (POTS, Plain old telephone system) of the backbone Tier 1 provider.

  This core is typically extremely fast consisting of DS3 links (the same ones used by International Discount Telecommunication (IDT)) or more modern links (like the ones used by VBNS of OC-48) and links who are beta testing OC-192 links while running Packet over Synchronous Optical Network (SONET) and eliminating the inefficiencies of ATM altogether.
- Links 9 and 11 are a reflection of links 5 and 3.
- Link 13 is a typical leased line, T1 link to the enterprise. This is how most enterprises connect to the Internet. However, after the 1996 telecommunications act, Competitive Local Exchanges (CLECs) emerged. CLECs provide superior service offerings at lower prices. Providers such as Qwest and Telseon provide Gigabit Ethernet connectivity at prices that are often below OC-3 costs (based on prices at the time writing).
- Link 15 is the enterprise's internal network.

  There should be a channel service (Time Division Multiplexing (TDM) side) and data service device (Data side), that terminates the T1 line and converts it to ethernet.

The even number links of FIGURE 2 represent the delays experienced in switches. These delays are composed of switching delays, route lookups, packet classification, queueing, packet scheduling and internal switch forwarding delays, such as sending a packet from the ingress unit, through the backplane to the egress unit.

As FIGURE 2 illustrated, QoS is needed to control access to shared resources during episodes of congestion. The shared resources are servers and specific links. For example, Link 1 is a dedicated point-to-point link, where a dedicated voice channel is setup at calltime, with a fixed bandwidth and delay. While Link 13 is a permanent circuit as opposed to a switched dedicated circuit, however, this is a digital line. QoS is usually implemented in front of a congestion point. QoS will restrict the traffic that is injected into the congestion point. Enterprises will have QoS functions that restrict the traffic that is being injected to their service provider. The ISP will have QoS functions that restrict the traffic that is injected into their core.

Tier 2 ISPs oversubscribe their bandwidth capacities hoping that not all their customers will need bandwidth at the same time. During episodes of congestion, switches buffer up packets until they can be transmitted. Link 5 and 9 are boundary links that connect two untrusted parties. The Tier 2 ISP must *control* the traffic injected into the network that must be *handled* by the Tier 1 ISP's core network. Tier 1 polices the traffic that customers inject into the network at Links 5 and 9. At the enterprise, many clients need to access the servers.

QoS Capable Devices

This section describes the internals of QoS Capable devices. One of the difficulties of describing QoS implementations are the number of different perspectives that may be used to describe all the features. The scope is limited to the priority based model and related functional components to implement this model. The priority based model is in fact the most common implementation approach due to its scalability advantage.

Implementation Approaches

There are two completely different approaches to implementing a QoS capable IP Switch or Server. These approaches are:

The **Reservation Model**, also known as Integrated Services/Resource Reservation Protocol (RSVP) or ATM, is the original approach, requiring applications to signal their traffic handling requirements. After signalling each switch that is in the path from source to destination reserves resources, such as bandwidth and buffer space, that either guarantee the desired QoS service or assure that the desired service is provided. It is not widely deployed because of scalability limitations. Each switch has to keep track of all this information for each flow. As the number of flows increase, the amount of memory and processing increases, hence limiting scalability.

The **Precedence Priority Model**, also known as Differentiated Services, IP Precedence Type of Service (TOS), or IEEE 802.1pQ, takes aggregated traffic, segregates the traffic flows into classes, and provides preferential treatment of classes. It is only during episodes of congestion that noticeable differentiated services effects are realized. Packets are marked or tagged according to priority. Switches then read these markings and treat the packets according to their priority. The interpretation of the markings must be consistent within the autonomous domain. The Differentiated Services model defines eight classes, from highest precedence to lowest: Expedited Forwarding (EF), Assured Forwarding 1-4(AF), and Best Effort (BE). Within each class, there are eight drop precedences, which indicate to the switch which packets are more important than others within that class. This results in a total of 8x8=64 Differentiated Services Code Points (DSCP).

This article focuses on the Precedence Priority model due to the increased scalability and current market acceptance of this approach. The next section details how QoS is actually implemented.

Functional Components—High Level Overview

In the Section, “Implementation Functions”, the three high level QoS Components, packet classification, packet scheduling and traffic shaping, and limiting are described. This section describes these QoS components in further detail.

A QoS capable device consists of the following functions:

- **Admission Control** accepts or rejects access to a shared resource. This is a key component for Integrated Services and ATM networks. Admission control ensures that resources are not oversubscribed. Due to this admission control is more expensive and less scalable.
- **Congestion Management** prioritizes and queues traffic access to a shared resource during congestion periods.
- **Congestion Avoidance** prevents congestion early using preventive measures. Algorithms such as Weighted Random Early Detect (WRED), exploit TCPs congestion avoidance algorithms to reduce traffic injected into the network, preventing congestion.
- **Traffic Shaping** reduces the burstiness of egress network traffic by smoothing the traffic and then forwarding out to the egress link.
- **Traffic Rate Limiting** controls the ingress traffic by dropping packets that exceed burst thresholds, thereby reducing device resource consumption such as buffer memory.
- **Packet Scheduling** schedules packets out the egress port so that differentiated services are effectively achieved.

In the next section, the modules that implement these high level functions are described in more detail.

QoS Profile

The **QoS Profile** contains information, inputted by the network/systems administrator on the definition of classes of traffic flows and how these flows should be treated in terms of QoS. For example, a QoS profile might have a definition that web traffic from the CEO should be given AF1 DiffServ Marking, Committed Information Rate (CIR) 1Mbs, Peak Information Rate (PIR) 5 Mbs, Excess Burst Size (EBS) 100 Kbytes, Committed Burst Size (CBS) 50 Kbytes. This profile defines the flow and what QoS the web traffic from the CEO should receive. This profile is compared against the actual measured traffic flow. Depending on how the actual traffic flow compares against this profile, the TOS field of the IP header is re-marked or an internal tag is attached to the packet header, which controls how the packet is handled inside this device.

The profile defines the grade of QoS that a flow should receive, such as Platinum, Gold, Silver, or Bronze. But the actual amount of traffic that a flow injects could exceed what was allocated and hence that traffic may be capped if it exceeds certain thresholds. However, if the switch is not busy, and no one else is using the resources, the switch may allow the flow to exceed the thresholds, since it does not make sense to waste unused resources. However, if the switch is busy, and gets congested, it enforces the flows thresholds and limits the amount of traffic according to the profile. Its like an airplane, the first class seats may be unused, so instead of wasting them, it makes sense for the airline to give the seats away to coach customers. However, if the first class seats are booked then the airline is very strict about seating assignments.

Functional Components—Detailed Modules

FIGURE 3 shows the main functional components that are involved in delivering prioritized differentiated services, that apply to a switch or a server. These include: the packet classification engine, the metering, the marker function, policing/shaping, I/P forwarding module, queuing, congestion control management and packet scheduling function.

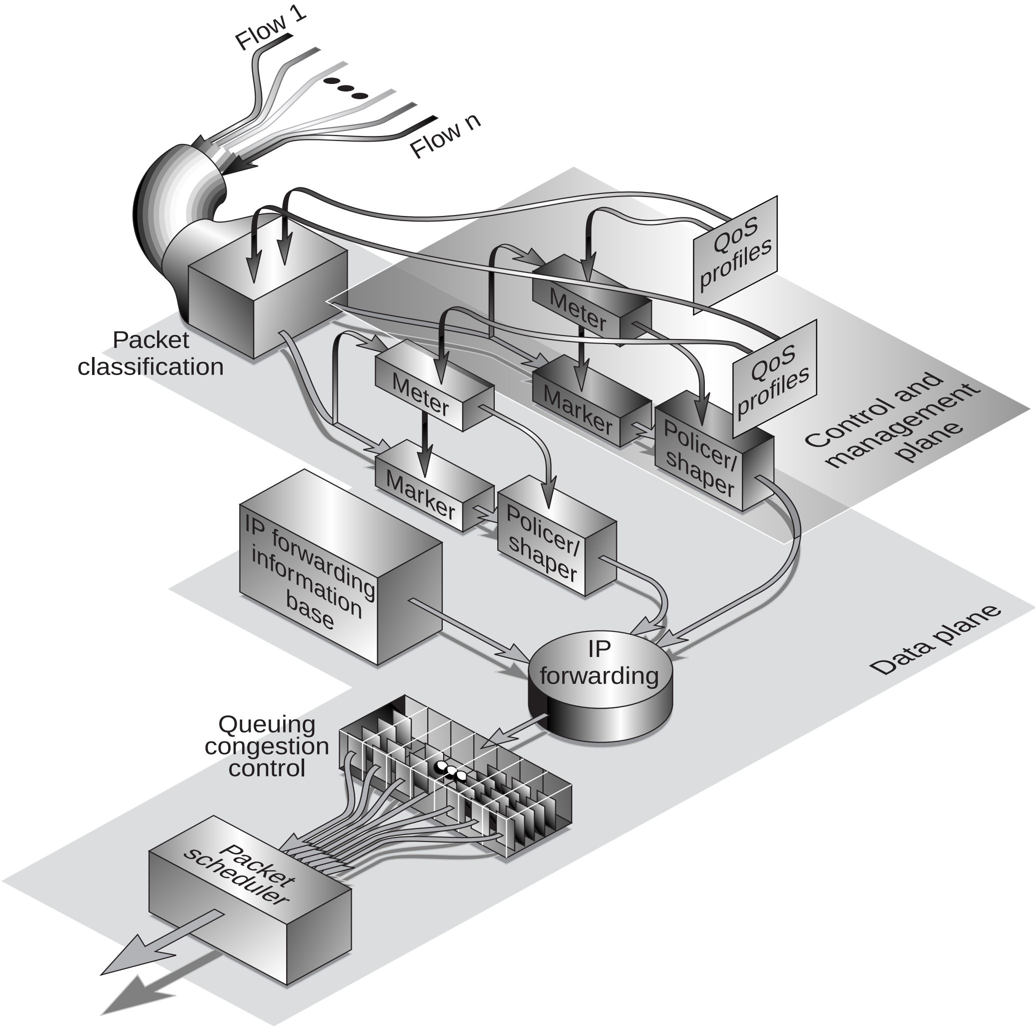

FIGURE 3 QoS Functional Components

Deployment of Data and Control Planes

Typically, if the example in FIGURE 3 was deployed on a network switch, there would be an ingress board and an egress board, connected together via a backplane. It would be deployed on a server and these functions would be implemented in the network protocol stack, either in the IP module, adjacent to the IP module, or possibly on the network interface card, offering superior performance due to the Application Specific Integrated Circuit (ASIC)/Field Programmable Gate Arrays (FPGA) implementation.

There are two planes:

1. Data Plane operates the functional components that actually read/write the IP header.
2. Control Plane operates the functional components that control how the functional units read information from the Network Administrator, directly or indirectly.

Packet Classifier

The Packet Classifier is a functional component that is responsible for identifying a flow and matching it with a filter. The filter is composed of source and destination, IP address, port, protocol, and the TOS field, all in the IP Header. The filter is also associated with information that describes the treatment of this packet. Aggregate ingress traffic flows are compared against these filters. Once a packet header is matched with a filter, the QoS profile is used by the meter, marker, and policing/shaping functions. Packet Classification performance is critical and much research has been published on it. One algorithm to note is the Recursive Flow Classification (RFC) algorithm. The basic idea behind the RFC packet classification is that the fields of the packet header are projected onto a finite natural number plane and divided up into equivalent sets. The rules are then parsed as indices are created. When a packet header is compared, a hierarchy of indexes are also compared in logarithmic base 2 searches. The algorithm achieves a good balance between reasonable memory requirements and lookup speed.

Metering

The metering function compares the actual traffic flow against the QoS profile definition. FIGURE 4 illustrates the different measurement points. The input traffic on average can arrive at 100 Kbytes/sec. However, for a short period of time, the switch or server allows the input flow rate to reach 200 Kbytes/sec for 1 second, which computes to a buffer of 200 Kbytes. For the time period of t=3 to t=5, the buffer is draining at a rate of 50 Kbytes/sec as long as the input packets arrive at 50 Kbytes/sec, keeping the output constant. Another more aggressive burst, arrives at the rate of 400 Kbytes/sec for .5 secs, filling up the 200 Kbytes buffer. From t=5.0 to 5.5, however, 50 Kbytes are drained, leaving 150 Kbytes at t=5.5 secs. This buffer drains for 1.5 secs at a rate of 100 Kbytes/sec. This example is simplified, so that the real figures need to be adjusted to account for the fact that the buffer is not completely filled at t=5.5 secs because of the concurrent draining. Notice that the area under the graph or the integral, represents the number of bytes in the buffer approximately, and bursts represent the high sloped lines above the green dotted line, representing the average rate or the CIR.

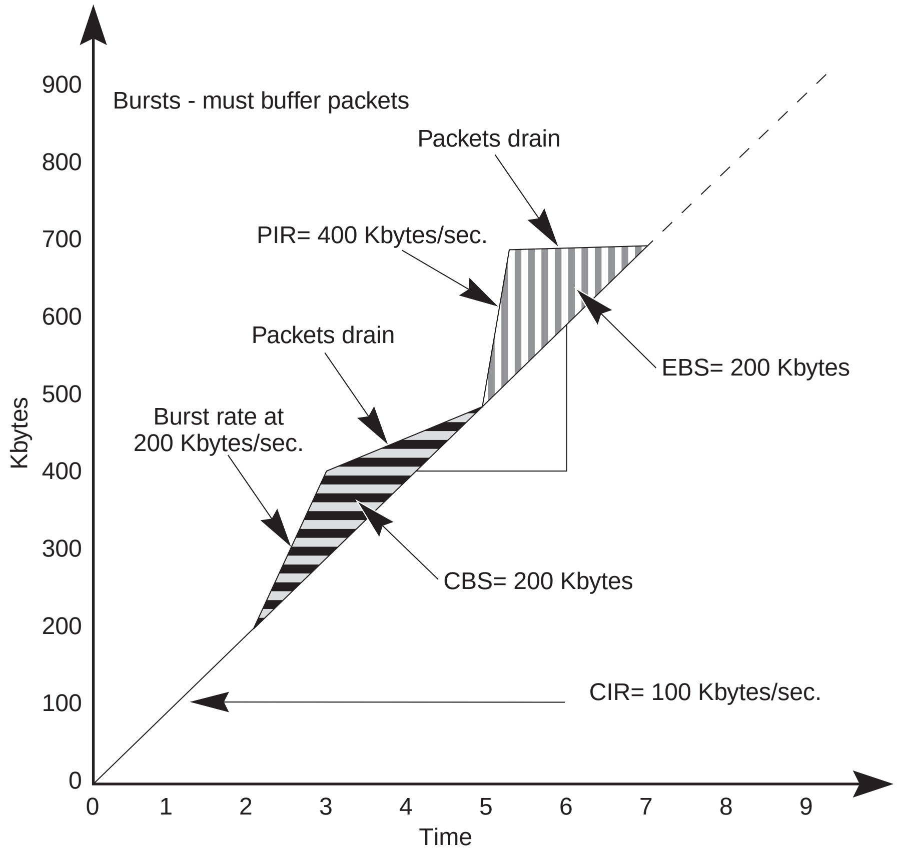

FIGURE 4 Traffic Burst Graphic

Marking

Marking is tied in with metering so that when the metering function compares the actual measured traffic against the agreed QoS profile the traffic is handled appropriately. The measured traffic measures the actual burst rate and amount of packets in the buffer against the CIR, PIR, CBS, and EBS. The Two Rate Three Color (TrTCM) algorithm is a common algorithm that marks the packets green if the actual traffic is within the agreed CIR. If it is above CIR or below PIR, the packets are marked yellow. Finally, if the actual metered traffic is at PIR or above, the packets are marked red. The device then uses these markings on the packet in the policing/shaping functions to determine how the packets are treated, for example, whether the packets should be dropped, shaped, or queued in a lower priority queue.

Policing/Shaping

The policing functional component uses the metering information to determine if the ingress traffic should be buffered or dropped. Shaping pumps out the packets at a constant rate, buffering packets in order to achieve a constant output rate. The common algorithm used here is the Token Bucket algorithm to shape the egress traffic and to police ingress traffic.

IP Forwarding Module

The IP forwarding module inspects the destination IP address and determines the next hop using the Forwarding Information Base. The forwarding information base is a set of tables populated by routing protocols and/or static routes. The packet is then forwarded internally to the egress board, which places the packet in the appropriate queue.

Queuing

Queuing encompasses two dimensions or functions. The first function is congestion control that controls the number of packets queued up in a particular queue (see the next section). The second function is differential services. Differential services' queues are serviced by the packet scheduler in a certain manner (providing preferential treatment to pre-selected flows) by servicing packets in certain queues more often than others.

Congestion Control

There is a finite amount of buffer space or memory, so the number of packets that can be buffered within a queue must be controlled. The switch or server forwards packets at line rate, however when a burst occurs or if the switch is oversubscribed and congestion occurs, packets are buffered. There are several packet discard algorithms. The simplest is Tail Drop, once the queue fills up any new packets are dropped. This works well for UDP packets, however there are severe disadvantages for TCP traffic. Tail drop causes TCP traffic in already established flows to quickly go into congestion avoidance mode and exponentially drops the rate at which packets are sent. This known problem is called global synchronization. It occurs when all TCP traffic is simultaneously increasing and decreasing flow rates at the same periods in time. What is needed is to have some of the flows slow down, so that the other flows can take advantage of the freed up buffer space. Random Early Detection (RED) is an active queue management algorithm that drops packets before buffers fill up, and randomly reduces global synchronization.

FIGURE 5 describes the RED algorithm. Looking at line C on the far right, when the average queue occupancy is from empty up to 75% full, no packets are dropped. However, as the queue grows past 75%, the probability that random packets are discarded quickly increases, up until the queue is full, then the probability reaches certainty. WRED takes RED one step further by giving some of the packets different thresholds at which packet probabilities of discard, start. As illustrated in FIGURE 5, Line A starts to get random packets dropped at only 25% average queue occupancy, making room for higher priority flows B and C.

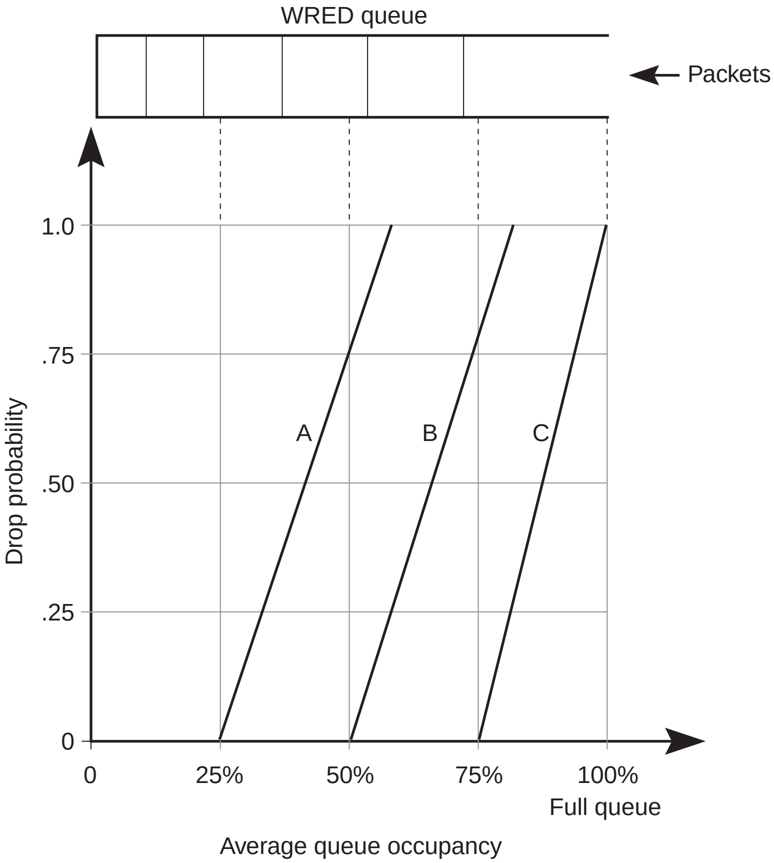

FIGURE 5 Congestion Control: RED, WRED Packet Discard Algorithms

Packet Scheduler

The packet scheduler is one of the most important QoS functional components. The packet scheduler pulls packets from the queues and sends them out the egress port, or forwards them to the adjacent STREAMS module, depending on implementation. There are several packet scheduling algorithms that service the queues in a different manner. Weighted Round Robin (WRR) scans each queue, and depending on the weight assigned a certain queue, allows a certain number of packets to be pulled from the queue and sent out. The weights represent a certain percentage of the bandwidth. In actual practice, unpredictable delays are still experienced, since a large packet at the front of the queue may hold up smaller-sized packets behind this large packet. Weight Fair Queuing (WFQ) is a more sophisticated packet scheduling algorithm that computes the time the packet arrived and the time to actually send out the entire packet. WFQ is then able to handle varying sized packets and optimally select packets for scheduling. WFQ conserves work, meaning that no packets are waiting idle, when the scheduler is free. WFQ is also able to put a bound on the delay, as long as the input flows are policed and the length of the queues are bound. In Class Based Queuing (CBQ) (used in many commercial products) each queue is associated with a class, where higher classes are assigned a higher weight translating to relatively more service time from the scheduler that the lower priority queues.

Summary

Enterprise QoS refers to the ability of enterprises to offer differentiated services to their customers. Businesses require that mission-critical applications have priority over non-essential traffic during episodes of congestion. This article described why certain emerging applications require QoS. In order for QoS to be effective, there needs to be a consistent common framework infrastructure from the services offered by the service providers and the enterprise infrastructure. The sources of delays and where QoS fits in the overall end-to-end solution was described. Finally, the main functional components of a QoS capable device were described in order to provide you with an understanding of the commercial implementations.

Author’s Bio: Deepak Kakadia

Deepak Kakadia is a staff engineer at Sun Microsystems Inc. located in Menlo Park, California. He works in Enterprise Engineering, Network Software. Deepak has been with Sun for seven years. He previously worked for various companies including Corona Networks as a Principal Engineer; Network Management Systems, as a team leader for the QoS Policy Based NMS subsystem; Digital Equipment Corp, where he worked on DEC OSF/1; Nortel Networks (Bell Northern Research) in Ottawa as member of the technical staff. Deepak received his B.Eng in Computer Systems from Carleton University, Ottawa, Canada and an MSc Computer Science from the New Jersey Institute of Technology, New Jersey, where he also completed his Ph.D qualifying exams and coursework. Deepak has also filed 2 patents: 1) Event Correlation and 2) QoS in the area of Network Management.