



White Paper
Intel Information Technology
Networking

Implementing Practical WAN Quality of Service

Intel IT implemented a Quality of Service (QoS) capability to improve application performance over our WAN and to prepare the WAN for Voice over IP (VoIP). We conducted extensive testing over time and gained a thorough understanding of the underlying QoS technologies to determine our best course of action. Through this approach, we successfully minimized the negative impacts of network congestion and provided a more solid WAN core.

Blaine Bauer, Tim Verrall, and Lilin Xie, Intel Corporation

November 2007

IT@Intel

Executive Summary

To provide global users with smooth application performance over the WAN, Intel IT researched and implemented a Quality of Service (QoS) capability that controls flow and prioritizes data handling to improve overall WAN application performance as well as prepares for Voice over IP (VoIP).

QoS technology can improve network performance for WAN applications, particularly with high-bandwidth applications such as VoIP and video.

Intel depends on its WAN circuits to stay efficient and competitive. As these circuits represent a major IT expense, we need to make sure that they are used as efficiently as possible. Efficient circuit utilization means that there isn't a lot of excess bandwidth and, at times, that it is also highly likely that some circuits will be heavily utilized. This results in congestion on the interfaces connecting those circuits.

Applications are adversely impacted when congestion causes packets to be delayed and/or dropped, and congestion from several applications running simultaneously can quickly become problematic if not properly managed. QoS technology can improve network performance for WAN applications, particularly with high-bandwidth applications such as VoIP and video. QoS won't eliminate the congestion; however, careful implementation can limit the impact of congestion to those applications least affected by it.

Through extensive lab testing and pilot production deployment, we developed a thorough understanding of QoS mechanisms and implemented a manageable solution that provides improved application support. Over time, we have continued our process improvements to enhance WAN QoS.

Contents

Executive Summary	2
Business Challenge	4
The Solution	5
Phase I: Understanding Objectives and Technology.....	5
Phase II: QoS Policy Deployment.....	9
Phase III: Policy Improvement.....	10
Results.....	12
Future Challenges	14
Next-Generation TCP.....	14
Extending Classification to the Edge.....	14
Call Admission Control.....	14
Policy-Based Management.....	14
Internet Protocol version 6.....	14
Conclusion	15
Authors	15
Acronyms	15

Business Challenge

WANs are a critical resource vital to our business at Intel. While demand for bandwidth is continually on the rise, investment in WAN bandwidth can be expensive. We find this to be especially true in our global work environment across oceans and in developing countries. For us to provide cost-effective service, managing bandwidth is critical.

Achieving and maintaining high utilization inevitably results in contention for the network. This becomes increasingly problematic with new applications, particularly VoIP and video. Within Intel's widely dispersed groups, multicast video became popular for top-down communication, but drops in the core of the network impacted thousands of viewers downstream.

Requirements also became increasingly divergent between applications that use as much bandwidth as they can get and transaction-based applications that generate little traffic but need quick response times.

Our network engineering community recognized that simply increasing WAN bandwidth would not meet all of the network's requirements. Campus

bandwidth is growing quickly to multi-gigabit speeds, and CPU improvements make the average end system capable of consuming considerable bandwidth. No matter how well provisioned the WAN, at times some links will be undersubscribed.

Additionally, the Intel WAN roadmap includes extensive migration to VoIP. However, unless traffic is differentiated in some way, we can not guarantee performance.

All of these challenges led us to identify a method of providing bandwidth and latency guarantees in the WAN.

The Solution

Our goal was to enhance the WAN with a QoS capability that would provide both bandwidth management and latency prioritization. Since cost-effectiveness is one of the main purposes of implementing QoS, the solution could not result in extensive administrative overhead. We tested and improved QoS policies over time, and the final solution met our basic requirements while providing for future growth.

Our implementation followed an Internet Engineering Task Force (IETF) standard method of marking and handling packets at potential congestion points. The primary benefits were more consistent latency and lower drop rates at router interfaces for applications sensitive to these variations. Since implementation, we have continued to conduct testing to help ensure ongoing process improvement.

Phase I: Understanding Objectives and Technology

Objectives, Design Goals, and Guiding Principles

Our first goal was to define the objectives of the initial rollout. The primary objective was improving performance for both VoIP—standard and proprietary—and multicast video. Our initial data objective was enabling a service level agreement (SLA) environment that could be tailored as needed.

We developed a number of design goals:

- Support convergence of services to IP including voice and video.
- Maintain service levels for bandwidth and jitter.
- Achieve consistent performance within bandwidth and latency limitations of media, for example circuits, regardless of other services in use.
- Provide limited service “priority” in terms of bandwidth guarantees or restrictions.
- Minimize support overhead with policy-based management or reduced complexity.

We were also guided by three principles:

- Use IETF standards wherever possible.
- Avoid penalizing user traffic.
- Prove the technology as stable and manageable before deploying QoS services.

Design

Iterative Design Effort

When we began our design process, information available about QoS tended to focus on specific technologies rather than on how to integrate, deploy, and manage those technologies. We needed to answer a number of questions to determine the initial capability. Our team engaged in an iterative process, analyzing design ideas, framing questions, and investigating through research and lab testing. Questions included:

- Of the dozens of QoS capabilities available in the WAN routers, which would provide tangible improvements? Would any interact negatively?
- How could—or should—various QoS technologies be integrated into a meaningful service?
- What impact could we expect on WAN router performance?
- Which management tools or methods could we use to monitor QoS and/or simplify management?
- How should we classify traffic from the various applications in use within Intel? How many traffic classes would be reasonable?
- Is it reasonable to provide enhanced performance to some customer applications when all customers pay the same for WAN services?
- How does a technical group decide which traffic to prioritize? Would we have to convene various business groups to seek consensus about which group's traffic is more important?

QoS Technologies Used

At the end of our analysis, we decided that the initial rollout should focus on providing a limited set of QoS technologies that wouldn't risk network stability. We based traffic classification and differentiation on application characteristics rather than on business criticality. Applications known to be less sensitive to latency and drops, such as file-transfer applications, we classified and queued separately to keep them from impacting other applications.

We used these QoS technologies:

- **Packet classification and marking.** Traffic is inspected and classified as it enters the WAN for the first time. We classified the WAN routers based on access lists, which match Transmission Control Protocol (TCP)/User Datagram Protocol (UDP) ports in received packet headers. Based on the classification, packets are marked, setting specific bits in the packet header. Since no consistent method exists for end-system applications to mark packets, all pre-existing marking would be over-written.
- **Priority queuing.** The basic output queuing mechanism in most routers is first in, first out (FIFO), where packets land at the end of the queue. While this method is efficient and fast, all packets incur queuing delay and must wait their turn to be transmitted. As packets traverse multiple router hops, some queuing delay can be expected at every hop. For most data packets, this isn't a problem. However, for

voice packets, cumulative queuing delays can adversely affect performance. Applying priority queuing places small, latency-critical traffic such as VoIP at the beginning of the queue rather than the end.

- **Class-based weighted fair queuing.** Class-based weighted fair queuing (CBWFQ) guarantees and/or limits the amount of bandwidth a class of traffic can use. This is typically accomplished by controlling the rate at which packets of different traffic types are added to an output queue. The process helps ensure that traffic classes are guaranteed a certain amount of bandwidth and that lower priority traffic is constricted but not completely starved of bandwidth. Unlike rate limiting, class-based queuing can use all of the available bandwidth. If a higher priority class doesn't need all of its allocated bandwidth, lower priority traffic can use remaining bandwidth. Configuring CBWFQ guarantees specific amounts or percentages of bandwidth; remaining bandwidth is allocated based on the ratio of the guarantees. For example, a queue with a 10 percent guarantee gets five times the bandwidth of a queue with a 2 percent guarantee.

Figure 1 diagrams these QoS capabilities in the Intel network.

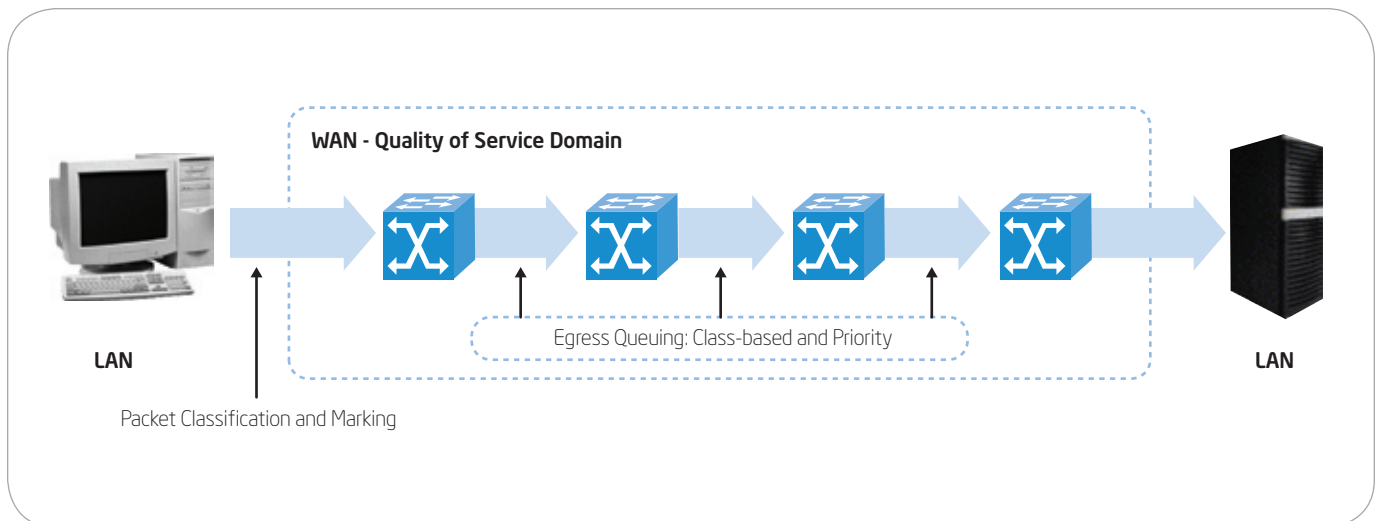


Figure 1. Quality of Service capabilities in the Intel network.

QoS Technologies Not Used

The list of QoS technologies available in the current generation of network equipment is quite extensive. We examined several technologies and determined which ones were not applicable for our initial implementation of QoS in the Intel network:

- **Other queuing technologies.** Fair queuing, weighted fair queuing, and per-flow queuing are generally simpler but didn't fulfill the specific requirements of bandwidth management plus prioritization on the same interface.
- **Rate limiting.** CBWFQ had a significant advantage over rate limiting, as it allowed lower-priority traffic to consume available bandwidth not being used by higher-priority applications. Rate limiting did not provide any advantage in the initial configuration, though it was used in later applications for security purposes. We used one particular type of rate limiting called traffic shaping for VPN and frame relay networks. This was where congestion points were not at the router's egress interface, but CBWFQ was still the mechanism for scheduling packets into the traffic-shaped bandwidth.
- **Congestion avoidance.** Congestion avoidance drops packets, based on the packet marking, before queues become excessively full. Since the network did not have very high drop rates, we determined that randomly dropping packets was more a detriment to the network than an improvement.
- **LAN QoS.** We initially rejected LAN QoS to keep the implementation scope within a reasonable limit. We later deployed it in a very limited fashion due to manageability issues in existing network equipment.

Lab Testing

WANs can use a wide variety of interface technologies, such as asynchronous transfer mode, frame relay, and so on. Since QoS often works quite differently on the various types of interfaces, with lab testing we found some issues that could have resulted in network problems. Some issues simply required changes in the configuration; others required software upgrades.

Throughput testing proved critical to helping ensure that prioritization and congestion management not only worked properly, but also did not adversely impact router platforms. We employed packet generators to create traffic with different QoS markings and packet sizes that corresponded to the expected packets in those traffic classes.

When we tested queuing, we generated traffic at rates that exceeded the intermediate links to induce congestion. Both marking and queuing resulted in processor impacts on most router platforms.

Through lab testing of router platforms, we developed:

- A router platform matrix
- The feature set
- The projected CPU impact

We had to upgrade several routers before QoS deployment to help ensure sufficient router CPU overhead.

One of the key factors for a low-impact rollout was our thorough understanding of the impact of the planned QoS configuration changes.

Phase II: QoS Policy Deployment

Once we attained a better understanding of the queuing methods, we defined an initial set of traffic classes and determined per-hop behaviors (PHBs) for each class. We prioritized voice traffic and bandwidth-managed remaining classes using CBWFQ. We specifically identified some traffic as non-latency-sensitive, allowing it to incur congestion without issue. Typically, these were file transfer applications that inherently generated large amounts of traffic. All other traffic we implicitly defined as latency-sensitive, marking it with a higher Differentiated Services Code Point (DSCP) value. Table 1 shows the bandwidth allocated to various classes.

To help ensure that packets did not remain too long in any router's egress queue, we also reduced queue depths in the latency-sensitive queue.

Pilot Implementation

Before rolling out new technologies in the Intel network, we employ the common practice of implementing the technologies on a network subset to avoid unexpected problems. Typically, we watch affected network components far more closely than usual, with additional monitoring of specific aspects that might be impacted by the change.

We implemented the pilot between sites that represented major administrative domains of the network:

- Bandwidth-rich core sites with minimal congestion
- Mid-tier sites with moderate amounts of bandwidth and occasional congestion
- Smaller sites with less bandwidth and more congestion

We closely monitored queue depth, packet drops per queue, and latency per queue. Resulting data proved that congestion in one queue could occur without causing packet drops in other queues.

Through the pilot, we discovered that monitoring the new queues proved too complex for existing tools. To address this, we developed an internal tool that provided graphing of the per-queue utilization and packet drops, as shown in Figure 2.

Table 1. Initial Quality of Service Policy

Traffic Class	Description	Bandwidth Allocated
Voice	Voice over Internet Protocol (VoIP)—including Voice over Frame Relay encapsulated in Internet Protocol (IP)	Priority queued
Latency-Sensitive 2	Default queue—intended for data traffic that is adversely affected by queuing delays	25%
Latency-Sensitive 1	Unused—available for any traffic needing increased priority	15%
Non-Latency-Sensitive	Data traffic that is not adversely affected by queuing delays—typically bulk data traffic	15%
Streaming Media	Multicast video	10%
Best Effort	Unused—available for troubleshooting purposes	5%

Differentiated Services

IETF's Internet-Draft RFC 2745, *An Architecture for Differentiated Services*, defines many of the technologies used in newer QoS implementations. Three of the more important concepts are:

- **Differentiated Services (DiffServ) domain.** Part of the network that uses a common set of QoS policies. The Intel WAN is a unique DiffServe domain.
- **Differentiated Services Code Point (DSCP).** A set of conventions for placing values in the Type of Service (ToS) byte in the standard IP header. Marking is the action of setting values in a header.
- **Per-hop behavior (PHB).** The administratively determined conventions of how each hop (typically a router) processes packets as they are forwarded through the network. The DSCP marking in the packet headers determines how PHB is performed. In the Intel WAN, the typical PHB is egress—output—queuing of packets as they pass through a router.

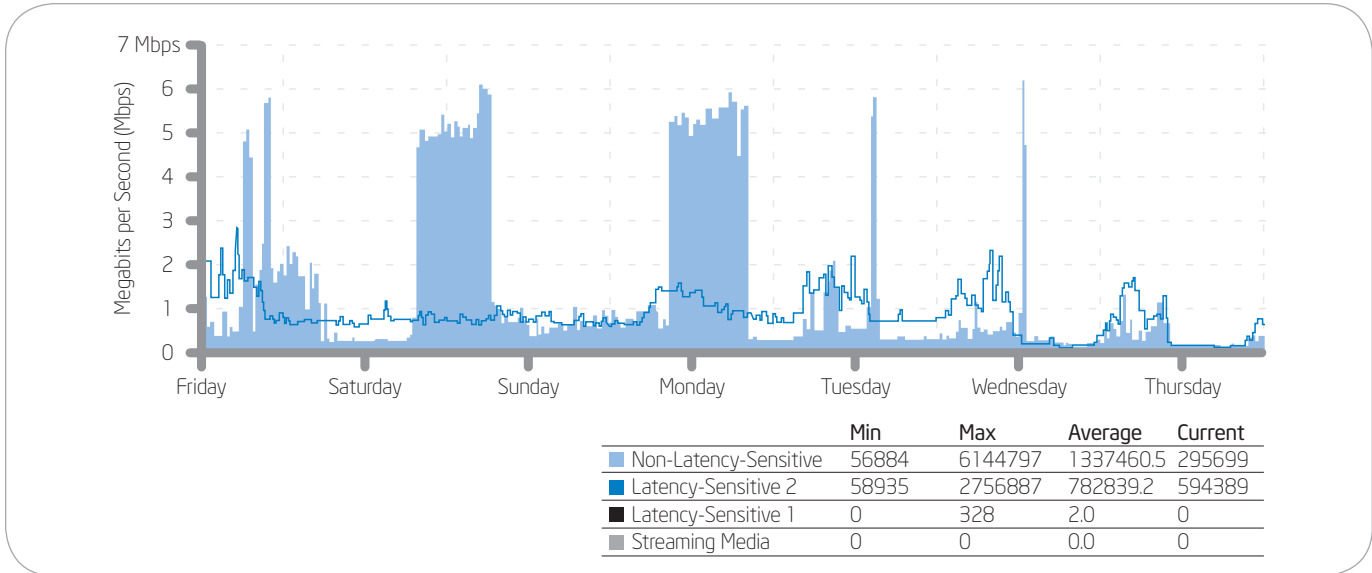


Figure 2. Interface traffic graph showing queues.

Rollout

We conducted a global rollout of QoS capabilities to all WAN routers based on a standard set of policies in the form of router configurations. We employed internally developed automated tools to distribute the policies quickly and efficiently.

Given the scope of this change, we expected that any number of unanticipated problems might occur. However, relatively few trouble calls came in. A small number of systems set the DSCP for their TCP connections to a non-zero value, but eliminated any connections when the network changed the DSCP en-route to a different non-zero value. These were older, proprietary UNIX*-based systems to which we had not applied operating system patches.

Phase III: Policy Improvement

After a period of measurement, we found that some portions of the initial QoS configuration weren't working as anticipated. The latency-sensitive queue experienced frequent congestion, resulting in latency and packet drops that did not provide an improvement over the original, pre-QoS configuration. Analysis revealed that far too much traffic was being classified into this queue.

Two types of protocols proved particularly problematic for classification and marking:

- Protocols using unspecified TCP/UDP ports for a specific application. This method is

commonly used by applications with remote procedure calls (RPCs).

- Protocols using a single destination TCP/UDP port for multiple purposes. Hypertext Transfer Protocol (HTTP) is the most common example: It can be used for Web-based applications that need fast response time or for downloading extremely large files that can cause network congestion.

We learned a critical lesson: If a queue is intended to provide better-than-default performance, the applications using it must be chosen carefully to help ensure that it doesn't become congested.

We retained the latency-sensitive queue, renaming it time-sensitive. We would no longer use it as the default; only specific applications would be classified as time-sensitive. We placed Secure Shell (SSH) and Simple Network Management Protocol (SNMP) traffic from the network management system in this queue to help ensure that routers could be reached even in periods of congestion. This proved beneficial for network maintenance.

During the time QoS was being implemented and improved, security had become a more critical function of the network. Computer worms were an increasing source of concern. Network engineers found ways to identify worm traffic and funnel it over the WAN to a depository to inspect and identify infected hosts. To rate-limit this traffic, we added a QoS queue that limited potential impact on the WAN.

Other tools that became available during Phase III were useful in improving the QoS implementation. Application visibility tools provided engineers with a better understanding of the applications in the network. The most useful tool used data exports from the WAN router, providing detailed profiles of the TCP/UDP ports and the quantity of data being transferred. Using this tool, we determined that computer backups should be placed in a separate queue that would only use bandwidth when it wasn't required for other applications.

Table 2. Improved Quality of Service Policy

Traffic Class	Description	Bandwidth Allocated
Voice	Voice over Internet Protocol (VoIP)—including Voice over Frame Relay encapsulated in Internet Protocol (IP)	Priority queued
Default Data	All other data applications	20%
Streaming Media	Multicast video	10%
Time-Sensitive	Data applications that generate few packets but need quick, reliable throughput	10%
Virus Control	Malicious traffic, typically worms, funneled to a depository for inspection	2%
Best Effort	Applications that may use considerable bandwidth but don't have specific throughput requirements, such as backups and file replication	2%

Other initial QoS policies worked well and we retained those in the new policy. The voice queue provided reliable, low-latency throughput for a limited amount of traffic. The streaming media queue, also used for a strictly limited amount of traffic and controlled at the source, met the goal of supporting of high-quality multicast video without drops in the network.

CBWFQ manages all classes except voice. These classes may exceed the allocated bandwidth percentages if other classes aren't using them. In practical operation, the default data class consumes the majority of bandwidth. The default data class utilizes a DSCP marking of zero, which is the default for most systems. While some older systems had problems with the network changing the DSCP to a non-zero value, we have not had any issues with setting the DSCP to zero.

We also found that the reduced queue depth resulted in unacceptably high packet drop rates. We tested various values, and, ultimately, default queue depths provided the best result for the time-sensitive queue, with a queue depth of 128 packets allowing the default and best-effort queues to minimize drops.

Results

We derived a number of benefits from implementing QoS, including:

- **Wider distribution of multicast video.** Some remote sites did not have sufficient unused bandwidth to sustain high-quality multicast video. With QoS, we can guarantee bandwidth.

- **Voice and data consolidation.** Prior to implementing QoS, we required specialized voice routers for Voice over Frame Relay. With QoS, we can tunnel this traffic over the data network using IP. With prioritized voice queues in place, we are prepared for VoIP.
- **Better control of worm-infected hosts.** We can identify hosts centrally and begin mitigation before an outbreak becomes too widespread.
- **Flexibility in managing application bandwidth.** With multiple data queues, we can keep high-bandwidth applications from adversely impacting business-critical applications.

We also learned many critical lessons through the process of implementing QoS. We validated some of our initial assumptions, while others were proven wrong. Our experience may apply to other large QoS implementations:

- **QoS testing is not just valuable, it is essential.** We avoided a considerable number of network outages by fully understanding the technology, including impact on routers and compatibility with various Layer 2 technologies.
- **Even with extensive testing, some issues are to be expected.** The scope of the issues depends on the scale of the deployment and the variety of systems on the network. In the Intel network, the problem scope was very small and eventually we avoided that by setting the default DSCP to zero. If non-zero marking will be used for specific applications, we now test prior to deployment.

- **QoS queues provide the greatest benefit within two scenarios:**
 - Limiting the impact an application or group of applications can have on the network. We accomplished this by restricting the queue size.
 - Limiting the impact of the network on an application or small set of applications. We accomplished this by placing these in a queue with a minimum amount of guaranteed bandwidth.
- **Limiting the impact of the network on an application is only feasible if the application generates a limited amount of traffic.** If an application generates so much traffic that it is a source of congestion, little can be done to improve its performance. Queues that are implemented to guarantee performance must be strictly controlled so that they rarely, if ever, exceed their allotted bandwidth. QoS is not a substitute for bandwidth.
- **Don't try to do too much with QoS.** QoS configurations can become quite complex. Varying configurations need to comprehend end-to-end differences where an application may be given high priority in one portion of the network and lower priority in another. In many cases, a consistent end-to-end policy is the most manageable approach.
- **Network management is an excellent candidate for a QoS queue.** Administrative access to routers through telnet, SSH, or SNMP generally results in a small amount of

traffic that can be critical to the operation of the network. This traffic can use a queue with other applications, but only if it's understood that the other applications will also generate traffic levels well within the amount guaranteed for that queue.

- **QoS can enhance other network services.** In the Intel network, QoS provides a vital function for network security. The actual benefit for any network is highly dependent on how the services are implemented. For example, not all networks will be able to identify worm traffic at the edge of the network.
- **Application characterization becomes extremely difficult when there is no direct correlation of application to TCP/UDP port.** The two difficult scenarios occur when:
 - An application uses a variety of ports, such as RPC or VoIP.
 - A single port supports a wide variety of network uses, such as HTTP.
- **QoS adds a layer of complexity to network management.** Depending on implementation and application requirements, a QoS-enabled network can realistically have high utilization rates without adversely affecting application SLAs. Because of this, link utilization becomes a less-relevant factor. Network monitoring and capacity management tools must be able to comprehend this shift in bandwidth management. If the network does not have consistent QoS policies end to end, management becomes even more complex.

Future Challenges

Next-Generation TCP

Consumer operating system TCP/IP stacks are currently being revised to incorporate numerous technical changes that increase their ability to adjust TCP window sizes. This results in higher levels of throughput, several times that of most current TCP/IP stacks on big, slow pipes—those with lots of bandwidth and low latency. We do not expect that other operating systems will incorporate similar enhancements.

While end users will appreciate the faster throughput, the network will most likely experience significantly more congestion. This will cause additional QoS requirements for data applications. The problems with classifying RPC and HTTP traffic may also become more significant. In addition, packet classification may require specialized devices that can perform deep-packet inspection at high bit rates.

Extending Classification to the Edge

VoIP is one of the most significant drivers for QoS in current networks. However, VoIP falls into a “difficult-to-classify” category as it can use a large range of UDP ports. Future QoS implementations will likely need to use the classification and marking done by the application. The challenge will be to make sure that marking is based on consistent policies, rather than the actions of end users. For example, a user may change a data application so that it appears to be a VoIP application as a way to achieve better performance, causing disruptions for several VoIP callers.

Call Admission Control

Video over IP and VoIP differ from many other applications in that if adequate bandwidth is not available, the application should typically deny the connection on that circuit and potentially select another call method. Oversubscription results in voice and video drops, which are far more disruptive than delays or drops with data applications.

Call Admission Control (CAC) keeps track of available bandwidth and allows the call or returns the equivalent of a busy signal. Current CAC methods do not support multiple applications such as voice and video. They also have other technical limitations. In some cases, gateway technology can solve the problem and efforts are underway in the IETF “Next Steps in Signaling” workgroup to improve CAC.

Policy-based Management

Currently, QoS management is a fairly manual task that pushes configuration changes to multiple routers. Policy-based management promises to simplify this task by providing a more human-oriented view of the QoS configuration. When this technology can resolve differing QoS policies in various portions of the network, it could become a highly beneficial configuration tool.

Internet Protocol version 6

Internet Protocol version 6 (IPv6) provides for QoS in a manner very similar to the current implementation of IP, version 4. When IPv6 is implemented, the same rigorous testing will be necessary to determine the impacts of IPv6 QoS.

Conclusion

By employing a process of rigorous testing and continuous improvement, Intel IT achieved the goal of effectively managing bandwidth on a combined voice, video, and data network.

With this thorough approach, we successfully minimized the negative impacts of network congestion and ultimately provided a more solid WAN core. We learned critical lessons in the process of implementing QoS that prepare us for future challenges. Our experience can provide insights beneficial to the management of most enterprise networks.

Authors

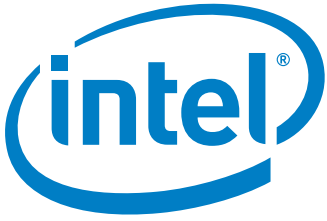
Blaine Bauer is a senior network engineer with Intel Information Technology.

Tim Verrall is a senior network engineer with Intel Information Technology.

Lilin Xie is a network engineer with Intel Information Technology.

Acronyms

ATM	asynchronous transfer mode	PHB	per-hop behavior
CAC	Call Admission Control	QoS	Quality of Service
CBWFQ	class-based weighted fair queuing	RPC	remote procedure call
DiffServe	Differentiated Services	SLA	service level agreement
DSCP	Differentiated Services Code Point	SNMP	Simple Network Management Protocol
FIFO	first in, first out	SSH	Secure Shell
HTTP	Hypertext Transfer Protocol	TCP	Transmission Control Protocol
IETF	Internet Engineering Task Force	ToS	Type of Service
IP	Internet Protocol	UDP	User Datagram Protocol
IPv6	Internet Protocol version 6	VoIP	Voice over Internet Protocol



www.intel.com/IT

This paper is for informational purposes only. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Intel, the Intel logo, Intel. Leap ahead. and Intel. Leap ahead. logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2007, Intel Corporation. All rights reserved.

Printed in USA
1107/SEP/RDA/PDF

 Please Recycle
ITAI Number: 07-3003w