Quality of Service and End-to-End Network Performance

Michael Roecken - 960 537 800
CSC458 Research Project
Professor J. Lee
April 4, 2000

Table of Contents

Introduction

Networks today have put a greater emphasis on bandwidth with the introduction of real-time applications. In order to supply the quality expected from these applications, the networking community has looked towards end-to-end network performance guarantees. This involves looking at end-to-end delay, delay jitter, throughput and packet loss rate. Computer networks have long provided best-effort service, but in these days, such a service does not provide the performance guarantees required. Instead, we must look at traffic models, traffic characteristics, service disciplines and buffering strategies to assist us. Each of these components provides details to how performance can be improved, for example queues and their use in transmitting packets from node to node in a network.

Quality of service (QoS) is an important feature of end-to-end network performance guarantees. It is the generalization of the performance of packet flow through networks. Quality of service software can provide the benefits of: control over resources, more efficient use of network resources, tailored grades of services, coexistence of high priority mission critical applications with other applications not requiring as high a priority and a foundation for a fully integrated network in the future. With such benefits, computer networks will soon be able to provide the end-to-end network performance guarantees that are required for the future evolution of networking.

In this report, many of these aspects of end-to-end network performance guarantees and quality of service will be discussed. First, an introduction into the field with a description of the parameters used to measure performance will be introduced. Then quality of service will be discussed so that later, the reader can have an insight into the material pertaining to this report. Two papers from the field will be summarized and discussed. The first paper by Hui Zhang, focuses on service disciplines for guaranteed performance in packet-switching networks. This paper discusses and compares the different types of service disciplines and relates them to the parameters used to characterize guaranteed service for packets. The second paper is from the network vendor, Nortel Networks, who discusses the Internet Protocol quality of service and how its importance to the business community will help service providers become profitable and competitive. The paper discusses the characteristics a service provider should provide with their networks in order to please their customers. This means having service level agreements that can prioritize the different classes of traffic from customers and being able to policy manage this traffic in order to improve the network implementation and design provided. The paper then concludes with a discussion of the advantages and disadvantages of both papers in reference to the research ideas presented and further possible application ideas that can be derived.

Background Information

Overview - End-to-End Performance Guarantees

In the networks of today, bandwidth is an important aspect. New applications such as RealAudio, RealVideo, Internet Phone software and video conferencing systems need a lot more bandwidth then earlier applications. Traditional network applications such as the World Wide Web (WWW), File Transfer Protocol (FTP) or telnet, cannot tolerate packet loss but are less sensitive to variable delays, whereas real-time applications have opposite characteristics, meaning they can handle a reasonable amount of packet loss but are critical towards high variable delays [2]. The aspect of providing higher bandwidth with reliable performance in order to satisfy these new applications is a main feature in the research area of end-to-end network performance guarantees.

Performance, in these terms, refers to the total effectiveness of a computer system, including throughput, individual response times, and availability. Therefore, achieving end-to-end performance guarantees is bound on:

- **end-to-end delay** - looks at total delay which involves, transmission delay, propagation delay, queuing delay, processing delay at the switches and depacketization delay due to jitter (to be defined later) [1] and the goal is to keep delays at a minimum;

- **end-to-end jitter** - *jitter* is the deviation in or displacement of some aspect of the pulses in a high-frequency digital signal and can be thought of as shaky pulses. The deviation can be in terms of amplitude, phase timing, or the width of the signal pulse. Among the causes of jitter are electromagnetic interference and crosstalk with other signals. Jitter can cause loss of transmitted data between network devices. The amount of allowable jitter depends greatly on the application [4]. Jitter is mainly due to the random queuing delay of packets and one way of eliminating it is as packets arrive at the destination, they are copied into a buffer that, in turn, is read out at a constant rate [1]. This introduces a depacketization delay usually equal to the queuing delay as mentioned above. Research on jitter is based on how effectively it can be handled and eliminated; and

- **packet loss** - due to buffer overflow at a switch, becomes an issue as resources across the network are shared and defined by a network's control strategy. As packets are moved through the network across different switches, buffers are required to temporarily store these packets before the next hop. Fluctuations in data-streams can arise since incoming packets can arrive at a rate faster then the buffer can empty at, and so with delays, a buffer may become full and packet loss occurs. Proper queuing methods and network control strategies can help regulate packet loss.

Performance guarantees can be classified in a manner where it is statistical, meaning performance can be guaranteed for a specific percentage of the packets, or deterministic, meaning performance is guaranteed for all of the packets. The statistical approach usually is more complicated as the network needs to know the arrival and departure bandwidth of the traffic, which are not always readily available [1]. Although, statistical multiplexing of sources is improved, the recording of these numbers also appears rather cumbersome and so it is tempting to look for alternate solutions. A deterministic approach has the advantages that: the source can easily ensure that its traffic meets the specifications; the network can easily verify that the traffic meets the specifications; and the network can guarantee strict bounds on delays and avoid all losses because of buffer overflows [1]. Unfortunately, deterministic approaches are based on worst case behaviours, and so resources can be left unutilized.

The actual components that assist in end-to-end performance guarantees are traffic models and service disciplines. A traffic model describes and characterizes the traffic that is generated at the source, while at a switch a service discipline helps decide which packets are to be forwarded in what order. Combined, these two components determine the bounds on performance guarantees. The challenge of a traffic model is that it must try to capture the nature of traffic and properly allocate the resources needed. This can be done using a statistical or deterministic approach. The challenge of a service discipline is to securely accommodate as many connections as possible with varying performance requirements. To be successful, both traffic models and/or service disciplines must be implemented in a simplified manner, in order to keep complexity at a minimum.

With a brief overview of the areas involved in end-to-end performance guarantees, this paper will further discuss quality of service (QoS), as underlying material for traffic models and service disciplines.

Quality of Service

Overall, quality of service is the ability to reserve resources with the network and terminal devices so as to ensure that certain perceptual or objective performance measures are met [1]. QoS refers to network performance measures such as rate (bandwidth), delay and loss, but also security, reliability and availability of connections. In the best case, QoS guarantees a very small packet loss rate and delay, where the smallest delay is comparable to the propagation delay. The worst quality, is the current so-called best-effort traffic model, where the network promises to deliver the packets only if it finds the resources to do so. The client selects the QoS based on the application. As was mentioned in the outset, it can be seen how real-time applications would want the highest QoS possibly.

The relationship between a network and a client is like a service contract. The contract obligates the network to transfer a client's information with a defined quality of service provided that the client's traffic conforms to its specified limits (bit rate, burstiness, etc.) — this is actually the basis for the guaranteed performance service model. With this, the objective of the network is to fulfill the largest set of contracts. This means that a network must be able to handle many different connections differently, since providing the best QoS for all connections is wasteful [1].

What affects the quality of service is a network's control strategy. A control strategy contains admission control, routing, flow and congestion control, and allocation control. Admission control decides whether there are enough resources within the network to accept a new connection. Routing involves deciding on what path packets should follow from source to destination. Flow and congestion control decides whether bit streams should be forwarded along their paths quickly to reduce delay or should they be slowed down to prevent congestion down the line. Finally, allocation control allows the network to control the bandwidth and buffers allocated to each path and switch. This allocation can be static (fixed at the beginning of the network request) or dynamic (changed during the transfer). This flexibility permits the network to provide connections with different QoS.

Therefore to provide high quality of service, it is important that a proper traffic model and service discipline is implemented along the network. Applications requiring tight control of QoS can be supported by Asynchronous Transfer Mode (ATM), but not easily over Transmission Control Protocol/Internet protocol (TCP/IP). Since the IP stack provides only one QoS, which is the best-effort model, the packets are transmitted without any guarantees for special bandwidth or time delays. Requests are handled on a FIFO basis, which means that all requests have the same priority and are handled one after the other. Therefore new strategies were developed for better QoS.

Integrated Services

Integrated Services brings new enhancements to the IP model to support real-time transmissions and guaranteed bandwidth for specific flows. A flow here, is a distinguishable stream of related packets, from a unique sender to a unique receiver that results from a single user activity and requires the same QoS [2]. This model uses the Resource Reservation Protocol (RSVP) that uses a more sophisticated resource allocation method in the switches (routers). In RSVP, the applications signal to the network their requirements, and the protocol reserves resources in the network switches.

Differentiated Services

Differentiated Services mechanism does not use per-flow signaling as in Integrated Services. Different service levels can be allocated to different groups of users, which means that the whole traffic is split into groups with different QoS parameters. This reduces the maintenance overhead in comparison to Integrated Services [2].

For further information on Integrated Services, RSVP and Differentiated Services please see [2].

Summary of Research Paper

Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks - Hui Zhang (October 1995)

Please see Appendix A for the full paper.

Introduction

Overall, this paper gives a review of the general issues associated with providing performance guarantees in packet-switching networks. It overviews traffic service models, traffic management algorithms and service disciplines. It then discusses two classes of service disciplines, work-conserving and non-work-conserving disciplines. For each, the paper gives a brief description and then illustrates a general framework in order to compare and contrast the two classes. Within each framework, some of the performance parameters such as end-to-end delay and packet loss are discussed, as well as implementation issues.

This section will now summarize some of the major points discussed throughout this paper.

Packet-Switching Network

Recall that in such a network, the data stream at the source is divided into packets of fixed or variable size. These packets follow a route or path in order to reach its destination. In doing so, packets from different links or connections must interact with one another at each switch (router) and without proper control, these interactions may negatively affect the performance experienced by a client of the network. Therefore, it is important that the service disciplines at the switches, which control the order in which packets are serviced, determine how packets from these different links interact with one another.

Service Model

This section introduces and discusses the *guaranteed performance service model* as a model that is based on pre-specified characterization of existing connections. It defines a contractual relationship between the client and the network as discussed earlier. The delay bound is specified by the application and does not change during the lifetime of the connection without the explicit request by the client. Although, this model is used throughout the paper, a new service model was proposed called the *predicted service model*. This model differs from the previous one, in that the current network load is based on measurement, and since the network load may vary, the service commitment is less reliable. The delay bound for a connection, in this model, is provided by the network and may vary due to the network load fluctuation.

For the guaranteed service, there are a few performance parameters that are used to help specify the requirements needed, as touched upon previously. The most important parameter is the end-to-end delay bound, which is essential for real-time applications. Throughput (bandwidth) is obviously also important. Another important parameter is the end-to-end delay jitter bound. For media playback, it is ideal to have zero delay jitter. Having delay jitter bounded makes it possible for a destination to calculate and allocate the buffer space needed in order to eliminate jitter. A small bound typically means that less buffer space is required. Since it is more important to provide proper end-to-end delay and delay jitter bounds, packets that arrive too early may not even be desirable in such an environment. The earlier a packet arrives before its delay bound, the longer it needs to occupy the buffer, one of the main differences between performance requirements of the guaranteed service model and the current best-effort service model. Performance bounds are more important in the guaranteed service model while average performance indices are more important for the best-effort service model. A final parameter is the packet loss probability due to buffer overflow and delay bound violations. A statistical service allows a nonzero loss probability while a deterministic service guarantees a zero loss probability. In this case, all packets will meet performance requirements even in the worst case, while with a statistical service, stochastic and probabilistic bounds are provided instead of worst case bounds. A statistical service does increase the overall network utilization by taking advantage of multiplexing gains.

In terms of traffic models, there is no agreement on which traffic model or which set of traffic parameters should be adopted. Some of the more popular ones are a Poisson model for data, an on-off model for voice and a Markovian model for video. The models listed, are either too simple to characterize the important properties of the source or too complex for easily managed analysis. Newer models have been derived that attempt to bound the traffic rather then characterize the process.

1. (Xmin, Xave, I, Smax) - Looks at a traffic stream where Xmin is the minimum packet interarrival time; Xave is the minimum average packet interarrival time during any interval of length I; and Smax is the maximum packet size.
2. (σ, ρ) - During a traffic stream interval of length *u*, the number of bits in that interval is less than $\sigma + \rho u$. σ can be viewed as the maximum burst size and ρ the long term bounding rate of the source.
3. (*r*, *T*) - A traffic stream where no more than *rT* bits are transmitted on any interval of length *T*.
4. D-BIND - A group of pairs of the form *rT* is specified, where *r* is the bounding rate for the traffic over interval *T*.

Each traffic model above, the exact traffic pattern for each connection is unknown, the only requirement is that the volume of the traffic be bounded in certain ways. This way, it is sufficient for resource memory algorithms to allocate resources by knowing just the bounds on the traffic volume. Actually bounding characterizations can be viewed on page 3 in the paper.

Traffic Management Algorithms

In packet-switching networks, if the arrival rate of traffic is greater than the service rate of packets at the switch, a delay is noticed, and if buffers at these switches become full then packet loss can occur. This problem is called congestion, and although networks are expected to become faster, congestion will most likely never go away. Various control algorithms have been proposed and are classified as either reactive/feedback control schemes or proactive/resource reservation algorithms. "Reactive approaches detect and react dynamically to congestion inside the network by relying on feedback information from the network, while proactive approaches eliminate the possibility of congestion by reserving network resources for each connection." Proactive approaches operate at a packet and connection level. At the connection level, a new connection is accepted only if there are enough resources to satisfy the requirements of the new and existing connections. At the packet level, the service discipline at each switch selects which packet to transmit next depending on a packet's performance requirement. It is important that a service discipline works closely with the admission control conditions. A reactive approach is best geared towards a best-effort service, while a proactive approach is better for the guaranteed performance service model. The two approaches can work together in an Integrated Services network as will be discussed and summarized below.

Service Disciplines

The remainder of this paper focuses on service disciplines and its two types: work-conserving and non-work conserving. As mentioned before, service disciplines and connection admission control algorithms provide two of the most important aspects of a proactive traffic management approach. Service disciplines allocate three types of resources: bandwidth, promptness and buffer space, which affects three performance parameters: throughput, delay and loss rate.

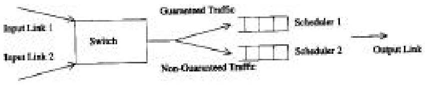


Figure 1: Servicing both guaranteed service and non-guaranteed traffic.

For the remainder of this paper, it is important to keep in mind the architecture shown in Figure 1 above for service disciplines in an Integrated Services network. At a switch, there are separate logical queues and service policies for guaranteed service and other packets. Packets in the non-guaranteed queue represent the best-effort service model and are only serviced when no packets from the guaranteed service model queue are ready for transmission.

A service discipline is designed to be efficient, protective, flexible and simple. "A service discipline is more efficient then another one if it can meet the same end-to-end performance guarantees under a heavier load of guaranteed service traffic." A connection admission control policy needs to limit the number of guaranteed service connections that can be accepted limiting the traffic load in the network, which should result in a higher utilization of the network. Protection from "ill-behaving users", network load fluctuation and best-effort traffic must be provided. A service discipline must be flexible enough to support applications with diverse traffic characteristics and performance requirements. Medical imaging has very different characteristics then video or audio and so the guaranteed performance service needs flexibility in order to support both. In addition, a service discipline should be capable to handle the needs of future applications. Finally, if a service discipline is simple then proper analysis and implementation can be done.

Work-Conserving Service Disciplines

With a work-conserving discipline, a server is never idle when there is a packet to be sent. This type of discipline affects the end-to-end delay analysis, buffer space requirements and delay jitter characteristics. This paper looks at the following disciplines: Delay earliest-due-date (delay EDD), virtual clock, fair queuing, packetized generalized processor sharing (PGPS), self-clocked fair queuing (SCFQ) and worst-case fair weighted fair queuing (WF^{2}Q). Below will be a brief summary of each major discipline before the analysis done by the paper is summarized. For further details about each of the service disciplines, one can further review Appendix A.

Virtual Clock

"This discipline attempts to emulate the time division multiplexing (TDM) system [1]. Each packet is allocated a virtual transmission line, which is the time that the packet would have been transmitted were the server actually doing TDM. Packets are transmitted in the increasing order of virtual transmission times." This algorithm guarantees good performance to a connection that behaves according to it arrival pattern.

PGPS and WF^{2}Q

Both of these disciplines attempt to approximate the fluid fair queuing (FFQ) policy. FFQ divides bandwidth up into *N*-bit cycles where *N* is greater than the number of active connections. FFQ is unrealistic as it assumes that the traffic is infinitely divisible and that a server can serve all connections with nonempty queues concurrently. More realistically, only one connection can be serviced at a time and an entire packet must be served before another one can be served. Therefore, PGPS attempts to approximate FFQ by looking at non-empty queues and sends packets in order of finishing times in FFQ. WF^{2}Q uses both start times and finish times of packets in the FFQ system in order to achieve a more accurate emulation. As proven in the paper, the difference between PGPS and WF^{2}Q does not affect the end-to-end delay bounds, but such a difference may be important if they are used to provide best-effort service.

Delay-Earliest-Due-Date (Delay-EDD)

Delay-EDD is based on the original EDD where an incoming packet is assigned a deadline and the packets are sent in order of increasing deadlines. The deadline is calculated as the sum of a packet's arrival time and the period of the traffic stream. In delay-EDD, the server sets up a service contract with each source, and as long as each source obeys its promised traffic specifications (sending rates) then the server will provide a delay bound. Here, the server sets a packet's deadline to the sum of its expected arrival time and the delay bound at the server.

General Discussion of Aspects Related to Work-Conserving Disciplines

All the methods discussed so far use some sort of a sorted priority queue. Upon arrival of each packet to a server, the packet is updated with a state variable, which is used as a priority index. This state variable is used to monitor and enforce its traffic. Packets are then served in the order of increasing priority index values. Table 1 found in the paper illustrates the formulas used to compute priority indexes by the different disciplines. Although, algorithms are similar, there are two differences 1) whether the calculation is based on just the arrival rate parameter or both the delay and arrival rate parameters and, 2) whether the updating is based on system-load independent or dependent parameters. The paper discusses these formulas in greater detail, and illustrates the two differences listed above. An important point the paper makes is that although delay bounds can be provided for each of these disciplines, having one rate parameter introduces the problem of coupling between the allocation of delay bound and bandwidth. This can result in a waste of resources for different performance levels. Also, under the second difference and through some detailed examples it can be said that the delay of a packet depends on the entire arrival history of the connection.

It is important to be able to characterize traffic in a networking environment, as we are interested in providing end-to-end delay bounds on a per connection basis. We can obtain worst-case local delay bounds at the switches and then use the sum of these local delays as the end-to-end delay bound, or smaller delay bounds can be determined by taking into account the dependencies in the successive switches that a connection traverses. Either way, the traffic needs to be characterized at each switch inside the network on a per connection basis.

Sometimes though, this can be difficult as traffic can become distorted inside the network, as the example in the paper illustrates. Three things can be done to solve this problem:

1) controlling the traffic distortion within the network,
2) accounting for the distortion during scheduling,
3) characterizing the traffic distortion throughout the network.

The first solution requires holding packets even though the server has the extra capacity, which leads into non-working conserving disciplines. The second solution implies that instead of scheduling packets in terms of their arrival times, the server should assign each packet a logical arrival time based on its traffic characterization and previous arrival history, and schedules packets based on this. The third solution involves many challenges that are outlined in great detail in the paper and are left to the reader to examine.

Table 2 in the paper compares the end-to-end delay and its characteristics for each of the disciplines discussed so far. In short, if a connection satisfies a particular traffic constraint for a discipline, and is allocated the right amount of buffer space, it can be guaranteed that an appropriate end-to-end delay bound and delay-jitter bound can be determined, given appropriate admission control conditions are satisfied. This table also illustrates previous points on relationships between end-to-end delay bound and bandwidth, and delay-jitter bounds and queuing delay.

As mentioned, all disciplines use a sorted priority queue, which requires an insertion operation into this sorted list that has a complexity of $O(\log n)$, where n is the number of packets in the queue. A network is designed to support many connections which means a switch usually has buffer space for a large number of packets. In some cases, a queue length can become quite large, and so it may not be feasible to operate such an operation at high speeds. Instead, arranging packets on a per connection basis and sorting the first packet of each queue would be better since packets on the same connection are serviced based on arrival times

Non-Work-Conserving Disciplines

With a non-work-conserving discipline, the server may be idle even when there are packets waiting to be sent. This paper looks at the following disciplines: Jitter-earliest-due-date (jitter-EDD), stop-and-go, hierarchical round robin (HRR) and rate-controlled static priority (RCSO). Below will be a brief summary of each major discipline before the analysis done by the paper is summarized. For further details about each of the service disciplines, one can further review Appendix A.

Jitter-Earliest-Due-Date (Jitter EDD)

Jitter-EDD extends delay-EDD in order to provide delay-jitter bounds. After serving a packet, a field in its header is modified to include the difference between its deadline and the actual finishing time. At the next server, this field is read and the packet is held for this period before it is eligible for scheduling.

Stop-and-Go

Stop-and-go uses a framing strategy and defines departing and arriving frames for each link. At a switch, a mapping occurs between the arriving frame of each incoming link and the departing frame of the outgoing link, by introducing a constant delay. According to the discipline, the transmission of a packet that arrived on any link during a particular frame should be postponed until the beginning of the next frame.

Rate-Controlled Static Priority (RCSP)

Given the other disciplines, RCSP has tried to achieve flexibility in the allocation of delay and bandwidth (as in jitter-EDD), but also simplicity of implementation (as in stop-and-go and HRR). RCSP has a rate-controller, which is a set of regulators that handle packets from arrival to departure calculating and assigning an eligibility time to a packet. Also, a static priority scheduler exists, which takes a packet with an eligibility time and schedules it for transmission. The scheduler always selects the packet at the head of the highest nonempty priority queue (a non-preemptive Static Priority policy). Each priority level corresponds to a delay bound.

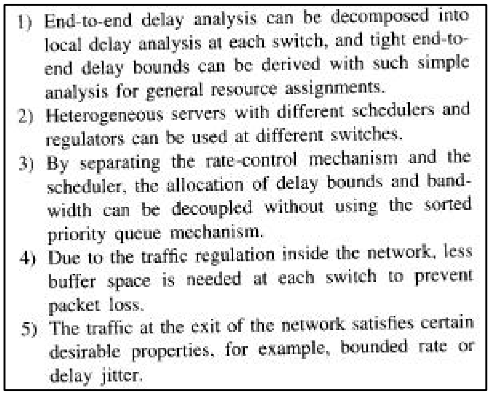
General Discussion of Aspects Related to Non-Work-Conserving Disciplines

The paper discusses that a general class of rate-controlled service disciplines can express all of the non-work-conserving disciplines introduced. A rate-controlled server, as mentioned before, has a rate-controller, which consists of a number of regulators responsible for shaping traffic, and a scheduler, which is responsible for multiplexing eligible packets coming from different regulators. Many different regulators and schedulers can be used, and so we have a general class of disciplines. RCSP and jitter-EDD are rate-controlled servers, while the other two disciplines can be implemented as rate-controlled servers with proper regulators and schedulers chosen. The paper continues into a detailed comparison of what rate-controllers and schedulers work well with each discipline.

Two general classes of regulators called delay-jitter controlling regulators and rate-jitter controlling regulators are defined, as they can be classified as regulators for each of the disciplines discussed. For a delay-jitter controlling regulator, the eligibility time of a packet is defined with reference to the eligibility time of the same packet at the next upstream server. A delay-jitter (DJ) regulator maintains all the traffic characteristics by completely reconstructing the traffic pattern at the output of each regulator.

Looking at Table 3 in the paper, the end-to-end delay characteristics and buffer space requirement for each of the disciplines can be seen. By appropriately setting parameters for regulators and local delay bounds at schedulers, rate-controlled service disciplines can provide end-to-end delay bounds almost as tight as those seen in work-conserving service disciplines. The paper proves this fact. With rate-controlled service disciplines, since the traffic can be characterized throughout the network, end-to-end delay bounds can be derived for general resource assignments. It has been shown that with properly chosen parameters for regulators and schedulers, these disciplines can always outperform FFQ-based disciplines in terms of the number of connections that can be accepted. As well, less buffer space is required to prevent packet loss, which is shown.

The paper has shown that non-work-conserving rate-controlled service disciplines exhibit many interesting features making them desirable for supporting guaranteed performance service. These are re-iterated in Table 1.



1) End-to-end delay analysis can be decomposed into local delay analysis at each switch, and tight end-to-end delay bounds can be derived with such simple analysis for general resource assignments.
2) Heterogeneous servers with different schedulers and regulators can be used at different switches.
3) By separating the rate-control mechanism and the scheduler, the allocation of delay bounds and bandwidth can be decoupled without using the sorted priority queue mechanism.
4) Due to the traffic regulation inside the network, less buffer space is needed at each switch to prevent packet loss.
5) The traffic at the exit of the network satisfies certain desirable properties, for example, bounded rate or delay jitter.

Table 1: Properties that make non-work-conserving service disciplines desirable for supporting guaranteed performance service.

One drawback of these disciplines is that a client is punished with a wasting of resources when it sends more than is specified, such as with live sources (i.e. video conferencing). As well, these disciplines are optimized for guaranteed performance service, and negatively affect the performance of other packets, such as best-effort service packets which may be left waiting in the queue as guaranteed service packets are waiting to become eligible for service. An additional note mentioned is that a non-work-conserving rate-controlled server can be modified to be work-conserving by introducing an extra queue called a standby queue. All packets that are in the rate-controller are also queued in the standby queue. Packets are inserted and deleted simultaneously from both and the scheduler will service the next packet in the standby queue only if there are no non-guaranteed packets and eligible packets in the scheduler. This way non-eligible packets are allowed to standby at the scheduler so that they can be transmitted when there is extra capacity available. This work-conserving rate-controlled server can provide the same end-to-end delay bound as its non-work-conserving complement.

This brings to a conclusion this paper, which summarized a number of packet service disciplines that are available to support guaranteed performance service connections in packet-switching Integrated Services networks.

Summary of Vendor Applications

IP QoS ¾ A Bold New Network - Nortel Networks (September 1998)
An IP Quality of Service backgrounder for service providers

Please see Appendix B for the full paper.

Introduction

During the evolution of the Internet over the past few years into a commercially operated network, Internet Protocol (IP) networks are growing to handle the migration of more then just data traffic, but also traffic from voice, frame relay, asynchronous transfer mode (ATM) and other network architectures. Currently, IP technologies are critical, as they are part of many public and private networks, such as corporate Intranets. The opportunities look endless, as businesses look to public IP networks, such as virtual private networks (VPNs), to handle their network traffic, as it is an opportunity to reduce costs, investment risk and operational complexity.

With this evolution, implementation issues do exist. Great demands on quality of service (QoS) are being placed with the emergence of real-time multimedia traffic over IP networks. All these applications require better performance guarantees than the current *best-effort* service model. Today's Internet, unfortunately, falls short of providing the reliability and performance guarantees that businesses are looking for to provide secure, predictable, measurable and guaranteed service for these applications.

This leaves opportunities for service providers to offer to businesses a public IP network based on IP-services with guaranteed QoS for their applications. Service providers can achieve profitability and competitiveness by providing such services. However, to achieve these goals may not be that simple. IP QoS is still a new concept with vendors offering different proprietary solutions while standards are still being developed.

IP QoS and Service Level Agreements

The paper continues by defining IP QoS as the performance of IP packets flowing through one or more networks. Their characterization of QoS includes, service availability, delay, delay variation (jitter), throughput and packet loss rate. The main goal for service providers, and the Internet, is to provide guaranteed IP QoS to user traffic on IP networks, including data, video, multimedia and voice. A service level agreement (SLA) defines end-to-end service specifications and may consist of the following: availability, services offered, service guarantees, responsibilities, auditing the service and pricing. Table 1 in the paper gives an example of a simple set of IP QoS levels that can be part of a SLA.

IP QoS Architecture

Various QoS architectures have been defined by various organizations, but for IP QoS, researchers are now focusing on two architectures, the Integrated Services architecture (Int-Serv) and the Differentiated Services architecture (Diff-Serv).

Int-Serv

It was proposed that the Resource Reservation Protocol (RSVP) be used as the signaling protocol in this architecture, and it assumes that resources are reserved for every flow requiring QoS at every router hop in the path between receiver and transmitter, using end-to-end signaling. Int-Serv provides three classes of service, as stated in the paper: guaranteed — with bandwidth, bounded delay and no-loss guarantees; controlled load — approximating best-effort service in a lightly loaded network; and best-effort — similar to what the Internet provides now under light to heavy load conditions.

Diff-Serv

Relatively new, Diff-Serv minimizes signaling and concentrates on aggregated flows and per hop behaviour applied to a network-wide set of traffic classes. The goal is to provide differentiated classes of services for Internet traffic, to support various types of applications and specific business needs. Referring to Figure 2 below, from the paper, traffic entering the network at an edge router (ER) is first classified for consistent treatment at each router inside the network.

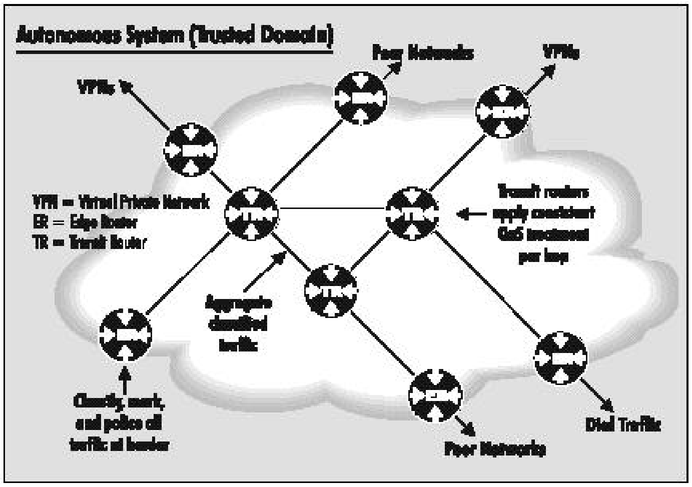

Autonomous System (Trusted Domain)
Peer Networks
VPNs
VPNs
VPN = Virtual Private Network
ER = Edge Router
TR = Transit Router
TR
TR
TR
ER
Transit routers apply consistent QoS treatment per hop
Aggregate classified traffic
Classify, mark, and police all traffic at border
Peer Networks
Dial Traffic

Figure 2: The Diff-Serv framework.

Inside, the traffic is separated accordingly into queues based on the class of traffic. A special field in the packet, called the Differentiated Services (DS) field, is used and marked so that routers downstream know what kind of treatment to use on the packet. This allows equipment providers the opportunity to develop configurable QoS capabilities based on bit patterns.

With Diff-Serv it can also be possible to extend QoS to more then one network domain (a partition of a network). But, there may also be cases where Int-Serv and Diff-Serv co-exist, and inter-networking must take place at the boundaries with a set of governing rules over flow. The paper does not go into more detail about this but references two other papers on the topic.

The paper continues with a discussion about some of the remaining issues with Diff-Serv, as listed on pages 9 - 10 in the paper. They state that a focus point would be to help improve inter-networking between multiple network domains, through better standardization. As well, handling aggregation at transit routers will greatly improve IP QoS, but this is something the industry, as a whole, needs to experiment with. With VPNs being of great importance, a serious challenge arises since Diff-Serv only manages traffic at entry points and does not provide a proper way to ensure exit capacity. This brings up the topic of traffic filtering. Another issue is trying to develop an inter-networking solution for mapping ATM QoS with IP QoS and class of service, which would help with standardizing more multi-protocol switching. Finally, a proper set of management tools must be developed in order to analyze end-to-end service quality.

Implementing IP QoS

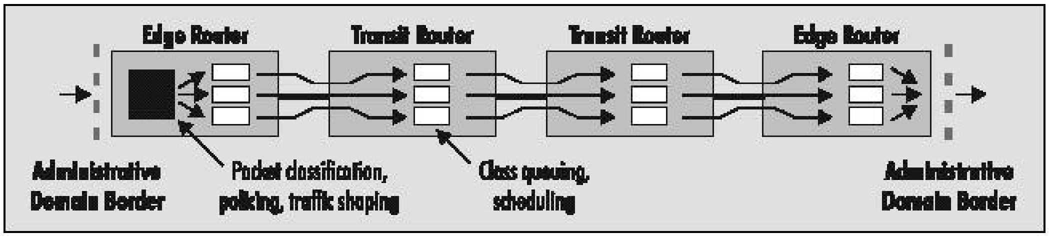

Edge Router
Transit Router
Transit Router
Edge Router
Administrative Domain Border
Packet classification, policing, traffic shaping
Class queuing, scheduling
Administrative Domain Border

Figure 3: Traffic flow across a network domain.

Figure 3 above, from the paper, shows how traffic flows across an IP network. Queues are provided at each node for traffic and where appropriate, dedicated queues are set up for particular traffic classes. Transit routers need not worry about policing of the traffic since it is known that the traffic comes from a reliable source. At a node, the traffic is inserted into a queue based on its DS marking and is transmitted according to traffic management mechanisms. This mechanism looks at the allocation of the output bandwidth and establishes rules for how to drop packets when congestion occurs. Edge routers do similar activities as transit routers, but also use policing methods to classify and mark the traffic incoming to the domain. The packet arrival rate is measured to ensure compliance with the SLA. The paper also includes a description of the different delays identified in IP networks and they can be viewed on page 11 of the paper.

Relating back to the SLA discussed earlier, the paper continues to talk about some of its features and how they can be designed into the network. Challenges faced by the industry is to move towards providing reliable service guarantees. Nodal delay, such as propagation and link speed delay are constant and queuing delay are introduced into the network at each node. Proper planning can control link speed and minimize hop counts, and queuing delay can be controlled with proper scheduling characteristics related to the queues (service disciplines). Delay variation or jitter is introduced by path variation, in part to poor network design. Most jitter is caused when packets get stuck behind other long packets, but class-based queuing can be used to reduce jitter for priority traffic. It is nearly impossible to design link capacity in a way that traffic will not get lost. Utilization and cost-effectiveness are factors and sometimes are involved in a trade-off. Planning capacity for a mixture of high and low priority traffic, so that if low priority traffic is lost, no harm is done can be beneficial. In short, good network design and proper queuing and scheduling mechanisms are key prerequisites for making service guarantees possible. On page 13 of the paper, there are several popular queuing and scheduling mechanisms listed.

IP QoS Traffic Management

During a packet's journey through the network, it comes across many traffic management mechanisms, such as policing, security, filtering, conditioning or classification mechanisms that influence the QoS.

The remainder of this section discusses the following scenario, as illustrated in Figure 4, from the paper.

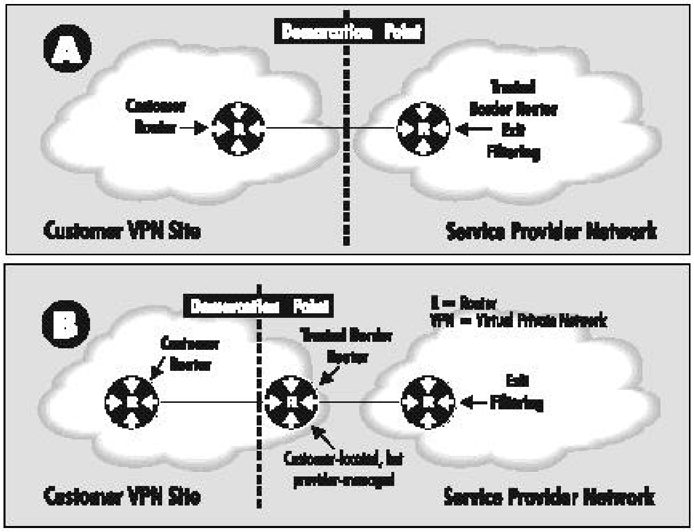


Figure 4: Traffic flow across a domain.

In Figure 4A, a customer owns and administrates its own WAN router, where packets are marked and classified using the DS field according to some agreed policies. In this case, the service provider's router policies the SLA for compliance. In Figure 4B, the service provider owns and administers the router on the customer's site, so the policing point shifts. This allows the service provider the opportunity to shape the traffic and also allow the customer an Ethernet connection, for example, from its router to the on-site service provider's router, affecting the priority scheme mapping done at both ends.

Traffic filtering is mostly done at exiting points from a network and is done for security purposes and to prevent the access link from becoming blocked by low value traffic. A filtering policy could be set up so that mission critical traffic has priority over low priority traffic. Security filtering is also used to keep unauthorized traffic from entering a private domain. Filtering must be done at the service provider's end as illustrated in Figure 4, otherwise malicious users could flood the link, causing denial of service for legitimate users.

Traffic classification is important because it helps determine the differences between different SLAs and how a customer's traffic is handled in the network. The traffic must be marked either by the customer or at the first router on the service provider's end. Multiple criteria are used to classify a customer's traffic.

As mentioned, at each router traffic is conditioned into the appropriate output queues. Each queue will have selectable drop algorithms such as Random Early Detection (RED) and also have programmable schedulers that implement Packetized Generalized Processor Sharing (PGPS), Round Robin (RR) and strict priority. With these, an important feature is configuring queue depth. However, there is a trade-off. Short queues can overflow quickly, but offer low delay. Longer queues are better at handling bursty traffic and provide enhanced throughput, but delay is negatively affected. Therefore, queue length must be configured in conjunction with scheduling and buffering as well as packet prioritizing.

Network Implementation

Network implementation is a difficult process seeing that it combines industry standardization, planning and development by using complex hardware and software configurations, legacy devices and mixed technologies. Router (switches) that forward packets and apply traffic conditioning at high speeds are essential in providing IP QoS. Providing priority carrying at reliable levels is important and will support and improve guarantees to customers. It is suggested in the paper that QoS products offer upwards of four queues per interface with scheduling algorithms that can be selected independently for each queue. A good choice would be RED, PGPS and strict priority so that a rich set of service classes can be used. It is also useful if QoS products can gather proper statistics to help with traffic engineering and service monitoring. This can also help with contract policing so that arrival rates can be verified for each class of service. The paper then also discusses how legacy routers can be dealt with in regards to QoS and is left to the reader to browse. Finally, this section concludes with a discussion of how ATM switches can be used in conjunction with IP routers to better improve QoS. The discussion talks about performance improvements and possible implementation scenarios. It introduces some other protocols, similar to RSVP, and talks about how they can be used in such an environment. Page 18 in the paper is a good reference.

Traffic Engineering and Managing Quality of Service

Under Diff-Serv, a traffic policy is required that allows relatively large amounts of traffic tolerant to packet loss to be dropped to ensure the safety of highly prioritized traffic. From previous sections, it is evident that network design and planning are an essential part of delivering quality to users.

In order to manage QoS, the difficult task of configuring many queues at each interface and translating SLAs into policing contracts at customer interfaces can only be done through proper policy management. Policies are used to define and dynamically control traffic behaviour within a network domain. The paper then discusses policy-based management and states that its five components are: policy editing, policy verification and conflict resolution, policy generation, policy distribution and policy evolution, each of which are described on pages 19 and 20 in the paper.

In order for monitoring and tracking, statistics can be collected at each node about the traffic flowing through each of the queues using the Simple Network Management Protocol (SNMP). These statistics can show average and peak throughput and packet loss levels for each traffic priority. Measuring delay is more difficult since it needs to be calculated between end points across the network for a particular packet. Therefore, it needs to be determined periodically for each of a customer's traffic classes, while jitter can be found over time through minimum and maximum observations.

The Future

IP QoS will be the bedrock for IP networking solutions to carry business critical applications, alongside Internet traffic, securely and reliably. What will advance the industry is the openness of the markets for competitiveness and profitability. Traffic patterns for IP traffic will gain substantially over the next few years, as the industry structure changes to accommodate this and newer trends.

Standards organizations and industry will further develop IP QoS standards. They will need to focus in on the standardizing of traffic conditioning methodology, class of service definition, policy management protocols and policy definition language. With these developments there will be an increased level of end-to-end performance guarantees and quality of service.

In short, this paper has discussed the applications of how a service provider can successfully offer its customers IP quality of service guarantees given the concepts and equipment currently available in order to handle more network demanding applications. In [5], Nortel Networks goes further and attempts to solve some of the issues outlined in this summary.

Suggested Research and Application Ideas

After reading and summarizing the two papers, this section will include an overall discussion about each, including the advantages and disadvantages of the topics discussed and any possible recommendations or additions that can be made.

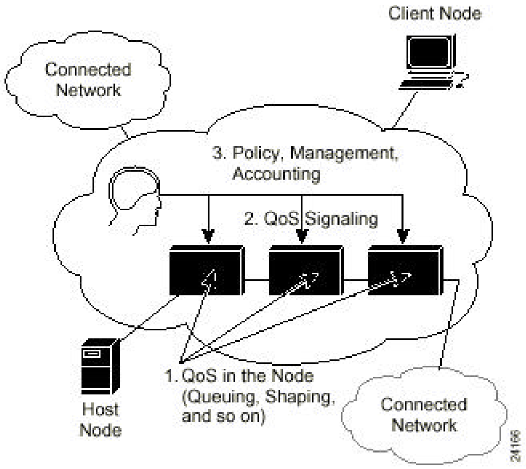

Client Node
Connected Network
3. Policy, Management, Accounting
2. QoS Signaling
1. QoS in the Node (Queuing, Shaping, and so on)
Host Node
Connected Network
24166

Figure 5: A basic quality of service implementation.

Figure 5 from [6] describes three of the components of a quality of service (QoS) implementation. In the first paper by Zhang, his discussion focused mainly on component 1 and he added comments about the other two components. In my opinion, some of his diagrams were either confusing or too simplistic to get a real understanding of the material presented. Otherwise his paper was very thorough touching upon many of the issues related to service disciplines. He was able to compare and contrast many of the disciplines and was able to show how they related to achieving guaranteed performance. One drawback I noticed from his paper was that his assumptions did simplify many of his arguments in that he assumed that proper traffic management and policing was being done. From reading other material on the topic, mainly [1], [2], [3] and [6], it was shown that traffic management and policing are key to how well a queuing and buffering system operate, and so his arguments may be altered if these assumptions were not made. Another point I would have liked to see explained more was that his paper involved talking about work-conserving and non-work-conserving service disciplines, he discussed each separately, but never really did go into detail about the major advantages of each and which ones are being used most frequently in the industry. His discussion was at a theoretical level, but I believe he could have done more by showing the actual practicality. Finally, he suggests that rate-controlled service disciplines will be of important interest for future research, and so when he discusses end-to-end delay under general resource assignments, his bounding of this delay is quite spacious, leaving room for newer techniques to be implemented. From this paper and [6], I realize that further research still needs to be done in congestion management and avoidance, as well as improving efficiencies in queuing and traffic shaping. If congestion can be anticipated or avoided, it makes traffic management easier to handle and thus allowing for better performance guarantees.

Overall, I found this paper very informative on the topic of discussion and ignoring the minor points mentioned above, is a good research source.

The paper from Nortel Networks was aimed at the business audience, especially service providers, and was an informative paper on what service providers should look for when designing their networks to meet quality of service (QoS) requirements. It focused on IP QoS since in their opinion, with the boom of the Internet, more businesses will want to move their data, voice and multimedia across public IP networks with greater bandwidth. Currently best-effort service does not do the job in a reliable and secure fashion, hence the introduction of Integrated and Differentiated Services. Referring to Figure 5, this paper talks about each of the three components of QoS. Mainly, the focus of further development was on providing efficient traffic policing and management. Their discussion of how a network should be designed and what should be planned and expected were done in nice simple detail. They discussed how service level agreements determine the type of service that is provided to the customer, in terms of QoS, but more importantly, traffic prioritizing. This is an important aspect that must be implemented by a service provider, or any network looking to achieve performance guaranteed QoS. In [5], Nortel goes further and describes some of the actual steps of how they would implement efficient and effective network policy management. A few drawbacks I noticed from their proposed implementation was that they relied heavily on current standards and did not mention much about how their implementation would handle the flexibility and scalability of newer protocols and standards. They discussed some new protocols, but focused on the resource reservation protocol (RSVP). From reading [6], it discusses newer protocols that can handle real-time applications better, such as the real-time protocol (RTP), which is really what customers would want implemented because it supposedly helps improve QoS. Finally, with the increased interest in and discussion about asynchronous transfer mode (ATM), it was surprising to see that this paper did not mention any direct implementations for ATM, but focused directly on providing IP QoS. They did provide a section on using the two in a combined environment, but Nortel did not have much more to say about the service.

Overall, this paper was directed for a specific audience and possible clientele. Nortel provided this audience with enough background knowledge so that they could specify in another paper such as [5], direct implementation possibilities relating to policy management. In reference to end-to-end performance guarantees, the paper achieved describing quality of service in a manner that showed its importance and role in helping a service provider gain valuable business; and in that regard, one can say the paper was complete.

In summary, the papers outlined that this area of networking is growing as more and more research is being conducted. The goal of providing the best possibly service for voice and multimedia, given its future practicality in the business world will continue the drive for success.

Conclusion

The research done for this report has shown the importance and significance of end-to-end network performance guarantees and how quality of service affects those guarantees. From the paper written by Zhang, we were able to understand the importance of service disciplines, traffic models and buffer allocation. His comparison of the different types of service disciplines showed the pros and cons of using particular types of queuing methods in order to move packets through a network. In order to achieve high quality of service for the current demands by real-time applications requires a prioritizing of traffic so that both best-effort services and guaranteed services can travel on a network together. The goals of quality of service include dedicated bandwidth, controlled jitter and latency, and improved packet loss characteristics. A proper implementation enables complex networks to control and predictably service a variety of networked applications and traffic types. With time and further research, this will be improved upon so that performance guarantees for voice and multimedia can be done in a secure and reliable fashion.

According to the paper from Nortel Networks, an implementation must also be able to provide proper congestion management and avoidance, as well as proper policy management. Their implementation focuses on the service providers that will provide networking services to business. As many businesses focus on IP networks, it is important for a service provider to be successful and profitable, and that they satisfy their customers by providing IP quality of service that will handle their traffic. In doing so, a proper service level agreement must be set prioritizing traffic between best-effort service and guaranteed service for real-time applications, such as voice and multimedia. But to do this, proper policy management must be available so that traffic is policed to ensure customer traffic suits the service level agreement. The result of well-defined traffic patterns and an improved ability to handle IP traffic is that quality of service will develop into a possibility for everyone to utilize thus causing a demand on service providers to be able to satisfy their customers.

In the end, quality of service can be improved upon with better means of avoiding and managing congestion, as well as constantly trying to improve upon queuing and buffering systems. But with advancements in technology and further research, computer networks will soon be able to achieve high standards in end-to-end network performance guarantees.

Bibliography

[1] Walrand, Jean and Varaiya, Pravin. 1996. *High-Performance Communication Networks*. San Francisco: Morgan Kaufmann Publishers, Inc.

[2] IBM International Technical Support Organization. October 1998. *TCP/IP Tutorial and Technical Overview Sixth Edition*, pages 505 - 534.

[3] Abeysundara, Bandula W. and Kamla, Ahmed E. *High-Speed Local Area Networks and Their Performance: A Survey*. ACM Computing Surveys, Vol 23, No 2, June 1991

[4] Whatis.com. (2000, April 1). Whatis.com Inc. [Online]. Available: HTTP: http://whatis.com

[5] *Preside Quality of Service*. September 1999. Nortel Networks

[6] *Internetworking Technology Overview - Quality of Service (QoS) Networking*. June 1999. Cisco Systems Inc. [Online]. Available: HTTP: http://www.cisco.com

Appendix A: Research Paper - Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks — Hui Zhang

Please find this article on the next page.

Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks

HUI ZHANG

Invited Paper

While today's computer networks support only best-effort service, future packet-switching integrated-services networks will have to support real-time communication services that allow clients to transport information with performance guarantees expressed in terms of delay, delay jitter, throughput, and loss rate. An important issue in providing guaranteed performance service is the choice of the packet service discipline at switching nodes.

In this paper, we survey several service disciplines that are proposed in the literature to provide per-connection end-to-end performance guarantees in packet-switching networks. We describe their mechanisms, their similarities and differences, and the performance guarantees they can provide. Various issues and tradeoffs in designing service disciplines for guaranteed performance service are discussed, and a general framework for studying and comparing these disciplines are presented.

I. Introduction

Communication systems have been revolutionized by technological advances in the last decade. The speed and capacity of various components in a communication system, such as transmission media, switches, memory, processors, have all followed technological curves that have grown either linearly or exponentially over the last ten years [18]. At the periphery of the network, driven by the same underlying technology—microelectronics, the capability of computers has been drastically increased while the cost has been significantly reduced. The advent of high speed networking has introduced opportunities for new applications such as video conferencing, scientific visualization and medical imaging. These applications have stringent performance requirements in terms of throughput, delay, delay jitter, and loss rate. Current packet-switched networks (such as the Internet) offer only a best-effort service, where the performance of each session can degrade significantly when the network is overloaded. There is an urgent need to provide network services with performance guarantees and to develop algorithms supporting these services.

One of the most important issues in providing guaranteed performance services is the choice of the packet service discipline at the switch. In a packet-switching network, packets from different connections will interact with each other at each switch; without proper control, these interactions may adversely affect the network performance experienced by clients. The service disciplines at the switching nodes, which control the order in which packets are serviced, determine how packets from different connections interact with each other.

Although service disciplines and associated performance problems have been widely studied in the contexts of hard real-time systems and queueing systems, results from these studies are not directly applicable in the context of providing guaranteed performance service in packet-switching networks. Analyses of hard real-time systems usually assume a *single server* environment, *periodic* jobs, and the job delay bounded by its *period* [53]. However, the network traffic is *bursty*, and the delay constraint for each individual connection is *independent* of its bandwidth requirement. In addition, bounds on *end-to-end* performance need to be guaranteed in a *networking* environment, where traffic dynamics are far more complex than in a single server environment. Queueing analysis is often intractable for realistic traffic models. Also, classical queueing analyses usually study *average* performance for *aggregate* traffic [32], [57], while for guaranteed performance service *performance bounds* need to be derived on a *per-connection* basis [13], [38]. In addition to the challenge of providing end-to-end per-connection performance guarantees to heterogeneous and bursty traffic, service disciplines must be *simple* so that they can be implemented at very high speeds.

Recently, a number of new service disciplines that are aimed to provide per-connection performance guarantees have been proposed in the context of high-speed packet-switching networks [12], [16], [21], [22], [26], [56], [62], [67]. Also, new analysis techniques have been proposed to address the performance issues of these disciplines [1], [5], [8], [9], [34], [35], [37], [40], [42], [48], [49], [58],

Manuscript received March 7, 1995; revised July 6, 1995.

The author is with The School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891 USA.

IEEE Log Number 9414946.

[60], [63], [64], [66], [68]. In this paper, we give an overview of the proposed service disciplines, and discuss the issues and tradeoffs in designing service disciplines in providing guaranteed performance service in packet-switching networks.

The rest of the paper is organized as follows. In Section II, we review general issues associated with providing performance guarantees in packet-switching networks and demonstrate the important role of service disciplines in the network control architecture. Sections III and IV discuss the two classes of service disciplines, work-conserving and nonwork-conserving disciplines respectively. In each of the two sections, a brief description of each discipline is first given before a general framework is presented to show the similarities and differences among them. The end-to-end delay characteristics, buffer space requirement, and implementation issues of each discipline are then discussed within the framework. In Section V, we summarize the paper by providing a taxonomy for classifying and comparing existing solutions.

II. Background

A. Network Model

We consider a network with arbitrary topology of links and switches.[1] Link are assumed to have bounded delay. Switches are assumed to be "nonblocking," i.e., when packets arrive at an input link, they can be routed directly to the appropriate output links without switching conflicts. Packets destined for different output links do not interfere with each other, and queueing occurs only at the output ports of the switch [30]. With these assumptions, a connection in such a network can be modeled as traversing a number of queueing servers, with each server modeling the output link of a switch. The network supports variable-size packets.

B. Service Model

We consider the following guaranteed performance service model: before the communication starts, the client needs to specify its traffic characteristics and desired performance requirements. When the network accepts the client's request, it guarantees that the specified performance requirements will be met provided that the client obeys its traffic specification.

In this model, the guaranteed performance service defines a contractual relationship between the communication client and the network [13], [15], [55]: the network promises to fulfill its obligation (guaranteeing the performance for the client's traffic) only if the client honors its own part of the contact (not sending more data than declared). In addition, the network may reject the client's request due to lack of resources or administrative constraints. In its basic form, the contract is signed before data transfer during a connection establishment process and is kept effective throughout the life time of the connection [16]. To increase dynamicity and flexibility, the model can also be extended to allow contract to be modified in the middle of a connection [50].

Recently, a new service model called the predicted service was proposed [7]. There are two important differences between the predicted service and the guaranteed performance service discussed in this paper. First, while the admission control, which decides whether there are enough resources within the network to accept a new connection, is used to support both types of service, the criteria are quite different. In order to decide whether there are enough resources, one has to know the current network load. For predicted service, the current network load is based on measurement; for guaranteed service, it is based on prespecified characterization of existing connections. Since the measured network load may vary, the service commitment by predicted service is less reliable. Secondly, in the predicted service, the delay bound or playback point for a connection is provided by the network and may vary due to the network load fluctuation. It is assumed that applications using the predicted service can adapt to the changing of the playback point and tolerate infrequent service disruptions. In the guaranteed performance service model, delay bound is specified by the application and does not change during the life time of the connection without the explicit request by the client.

1) Performance Parameters in Guaranteed Service: The most important clauses in the service contract are the specifications of performance requirements and traffic characteristics. For the performance parameters, the single most important one is the end-to-end delay bound, which is essential for many applications that have stringent real-time requirements. While throughput guarantee is also important, it is provided automatically with the amount specified by the traffic characterization (Section II-B.2). Another important parameter is the end-to-end delay jitter bound. The delay jitter for a packet stream is defined to be the maximum difference between delays experienced by any two packets [13], [56]. For continuous media playback applications, the ideal case would be that the network introduces only *constant* delay, or *zero* delay-jitter. Having a bounded delay-jitter service from the network makes it possible for the destination to calculate the amount of buffer space needed to eliminate the jitter. The smaller the jitter bound, the less amount of buffer space is needed. Since it is more important to provide end-to-end delay and delay-jitter *bounds* than average low delay for guaranteed service class, packets arriving too earlier may not even be desirable in such a environment. In fact, the earlier a packet arrives before its delay bound or playback point, the longer it needs to occupy the buffer. This is one of the most important differences between the performance requirements of the guaranteed-performance service and the best-effort service provided by the traditional computer networks: performance bounds are more important for the guaranteed service while

[1] In the literature, the term "switch" is used in the context of ATM networks, while "gateway" or "router" is used more often in an internetworking environment. In this research, we will call switching elements as "switches."

average performance indices are more important for the best-effort service.

A third important parameter is the loss probability. Packet loss can occur due to buffer overflown or delay bound violation. A statistical service [13], [37], [66] allows a nonzero loss probability while a *deterministic* service guarantees zero loss. With a deterministic service, all packets will meet their performance requirements even in the worst case. With a statistical service, stochastic or probabilistic bounds are provided instead of worst case bounds. Statistical service allows the network to overbook resources beyond the worst-case requirements, thus may increase the overall network utilization by exploiting statistical multiplexing gain.

2) Traffic Models in Guaranteed Service: Although there is a general consensus within the research community on the (super) set of parameters to characterize performance requirements, there is no agreement on which traffic model or which set of traffic parameters should be adopted. In the traditional queueing theory literature, most models are based on stochastic processes. Among the more popular ones are the Poisson model for data [32], on-off model for voice sources [3] and more sophisticated Markovian models for video sources [43]. A good survey for the probabilistic models for voice and video sources is presented in [46]. In general, these models are either too simple to characterize the important properties of the source or too complex for tractable analysis.

Recently, several new models are proposed to *bound* the traffic rather than characterize the process exactly. Among them are: $(X\min, X\mathrm{ave}, I, S\max)$ [16], (σ, ρ) [8] (r, T) [20], [26], and the D-BIND model [35]. A traffic stream satisfies the $(X\min, X\mathrm{ave}, I, S\max)$ model if the inter-arrival time between any two packets in the stream is more than $X\min$, the average packet inter-arrival time during any interval of length I is more than $X\mathrm{ave}$, and the maximum packet size is less than $S\max$. Alternatively, a traffic stream satisfies the (σ, ρ) model if during any interval of length u, the number of bits in that interval is less than $\sigma + \rho u$. In the (σ, ρ) model, σ and ρ can be viewed as the maximum burst size and the long term bounding rate of the source respectively. Similarly, a traffic stream is said to satisfy (r, T) model if no more than $r \cdot T$ bits are transmitted on any interval of length T. Rather than using one bounding rate, the deterministic bounding interval-dependent (D-BIND) model uses a family of rate-interval pairs where the rate is a bounding rate over the corresponding interval length. The model captures the intuitive property that over longer interval lengths, a source may be bounded by a rate lower than its peak rate and closer to its long-term average rate.

In each of the above models, the exact traffic pattern for a connection is unknown, the only requirement is that the volume of the traffic be *bounded* in certain ways. Such bounding characterizations are both general and practical. They can characterize a wide variety of bursty sources. In addition, it is sufficient for resource management algorithms to allocate resources by knowing just the *bounds* on the traffic volume.

A bounding characterization can either be *deterministic* or *stochastic*. A bounding deterministic traffic characterization defines a deterministic traffic constraint function. A monotonic increasing function $b_j(\cdot)$ is called a deterministic traffic constraint function of connection j if during *any* interval of length u, the number of bits arriving on j during the interval is no greater than $b_j(u)$. More formally, let $A_j(t_1, t_2)$ be the total number of bits arrived on connection j in the interval of (t_1, t_2), $b_j(\cdot)$ is a traffic constraint function of connection j if $A_j(t, t+u) \leq b_j(u)$, $\forall t, u > 0$. Notice that $b_j(\cdot)$ is a time invariant deterministic bound since it constrains the traffic stream over every interval of length u. For a given traffic stream, there are an infinite number of valid traffic constraint functions, out of which, a deterministic traffic model defines a parameterized family. All of the above traffic models have corresponding traffic constraint functions. For example, the traffic constraint function of (σ, ρ) model is $\sigma + \rho u$. The traffic constraint can also be stochastic. In [37], a family of stochastic random variables are used to characterize the source. Connection j is said to satisfy a characterization of $\{(\mathbf{R}_{t_1,j}, t_1), (\mathbf{R}_{t_2,j}, t_2), (\mathbf{R}_{t_3,j}, t_3) \cdots\}$, where $\mathbf{R}_{t_i,j}$ are random variables and $t_1 < t_2 < \cdots$ are time intervals, if $\mathbf{R}_{t_i,j}$ is *stochastically larger* than the number of bits generated over any interval of length t_i by source j. This model is extended in [66] by explicitly considering the interval-dependent property of the source: over longer interval lengths, a source may be bounded by a rate lower than its peak-rate and closer to its long-term average. The resulted model is called Stochastic Bounding Interval Dependent or S-BIND model. Another related traffic model is the exponentially bounded burstiness (EBB) process proposed in [59], [60]. A source is said to be EBB with parameters $(\rho, \mathcal{A}, \alpha)$ if $\Pr\{A[s, s+t] \geq \rho t + \sigma\} \leq \mathcal{A}e^{-\alpha\sigma}$ $\forall \sigma \geq 0$ and $s, t > 0$ where random variable $A[t_1, t_2]$ denotes the total number of bits generated by a source in the interval $[t_1, t_2]$.

In this paper, we assume that a communication client uses a deterministic bounding traffic model to specify its traffic if it requests a deterministic service and use a stochastic bounding traffic model to specify its traffic if it requests a statistical service.

C. Traffic Management Algorithms

In packet-switching networks, there is the possibility that the aggregate rate of the input traffic to the network (or a portion of the network) temporarily exceeds the capacity of the network, in which cases packets may experience long delays or get dropped by the network. This is called congestion. Although networks are expected to become even faster, the problem of congestion is not likely to go away [25]. Various congestion control or traffic management algorithms have been proposed in the literature. These solutions can be classified into two classes: reactive, or feedback control schemes [24], [51], and proactive, or resource reservation algorithms [16], [39], [67].

Reactive approaches detect and react dynamically to congestion inside the network by relying on the feedback

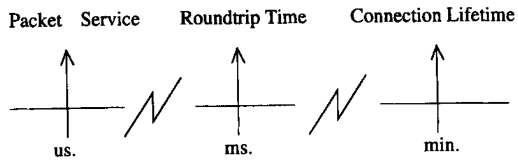


Fig. 1. Control timescales for traffic management algorithms.

information from the network, while proactive approaches eliminate the possibility of congestion by reserving network resources for each connection. From the point of view of control time scale, reactive approaches operate in a time scale of several *round-trip times* since the length of the interval between the time when the congestion is detected and the time when the congestion signal is passed back to the source is on the order of one round-trip time. Proactive approaches, on the other hand, operate on at least two timescales: connection level and packet level. At the connection level, when a new connection request comes in, a set of connection admission control conditions are tested at each switch. The new connection is accepted only if there are enough resources to satisfy the requirements of both the new connection and existing connections. At the packet level, the packet service discipline at each switch selects which packet to transmit next by discriminating packets based on their performance requirements. Usually, different service disciplines need different admission control algorithms. A complete solution needs to specify both the service discipline and the associated connection admission control conditions.

The three timescales used by traffic management algorithms are illustrated in Fig. 1. While a reactive approach is suitable for supporting best-effort service, a proactive traffic management architecture is better for guaranteed performance service. The two approaches can coexist in an integrated services network.

D. Service Disciplines

As can be seen from Fig. 1, packet service disciplines operate at the smallest time scale, or with the highest frequency. Together with connection admission control algorithms, they provide the two most important components in a proactive traffic management architecture. While connection admission control algorithms *reserve* resources during connection establishment time, packet service disciplines *allocate* resources according to the reservation during data transfer. Three types of resources are being allocated by service disciplines [12] bandwidth (*which* packets get *transmitted*), promptness (*when* do those packets get *transmitted*) and buffer space (*which* packets are *discarded*), which, in turn, affects three performance parameters: throughput, delay, and loss rate.

Even in reactive or feedback-based traffic management architecture, appropriate scheduling at packet switches will make end-to-end control more effective [12], [31]. In the rest of the paper, we consider architecture shown in Fig. 2 for service disciplines in integrated services networks. There are separate queues and service policies for guaranteed service and other packets. Best-effort packets are transmitted only when no packets from the guaranteed service queue are available for transmission. It should be noticed that the two queues in Fig. 2 are logical ones. Depending on the service discipline, each logical queue can corresponds to multiple physical queues. For example, if a Static Priority discipline with n priority levels is used for guaranteed traffic and a round robin discipline with m classes is used for nonguaranteed traffic, the number of physical queues is $n + m$ at each output port. In this paper, we will focus on the service disciplines for guaranteed traffic.

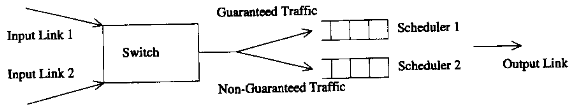


Fig. 2. Servicing both guaranteed service and nonguaranteed traffic.

Although it is possible to build a guaranteed performance service on top of a vast class of service disciplines [14], we would like a service discipline to be efficient, protective, flexible, and simple.

1) Efficiency: To guarantee certain performance requirements, we need a connection admission control policy to limit the guaranteed service traffic load in the network, or limit the number of guaranteed service connections that can be accepted. A service discipline is more efficient than another one if it can meet the same *end-to-end* performance guarantees under a heavier load of guaranteed service traffic. An efficient service discipline will result in a higher utilization of the network.

2) Protection: Guaranteed service requires that the network protects well behaving guaranteed service clients from three sources of variability: ill-behaving users, network load fluctuation, and best-effort traffic. It has been observed in operational networks that ill-behaving users and malfunctioning equipments may send packets to a switch at a higher rate than declared. Also, network load fluctuations may cause a higher instantaneous arrival rate from a connection at some switch, even though the connection satisfies the traffic constraint at the entrance to the network. Another variability is the best-effort traffic. Although the guaranteed service traffic load is limited by connection admission control, best-effort packets are not constrained. It is essential that the service discipline should meet the performance requirements of packets from well behaving guaranteed service clients even in the presence of ill-behaving users, network load fluctuation and unconstrained best-effort traffic.

3) Flexibility: The guaranteed performance service needs to support applications with diverse traffic characteristics and performance requirements. Scientific visualization and medical imaging will have very different characteristics from video. Even for video, conferencing applications, movie applications, and HDTV require different qualities of service. Other factors, such as different coding algorithms and different resolutions, also contribute to the diversity of video requirements. Because of the vast diversity of traffic characteristics and performance requirements of existing applications, as well as the uncertainty about future applications, the service discipline should be flexible to allocate different delay, bandwidth, and loss rate quantities to different guaranteed service connections.

4) Simplicity: The service discipline should be both conceptually simple to allow tractable analysis and mechanically simple to allow high speed implementation.

III. Work-Conserving Service Disciplines

A service discipline can be classified as either work-conserving or nonwork-conserving. With a work-conserving discipline, a server is never idle when there is a packet to send. With a nonwork-conserving discipline, each packet is assigned, either explicitly or implicitly, an *eligibility* time. Even when the server is idle, if no packets are eligible, none will be transmitted. As will be shown later in this paper, whether a service discipline is work-conserving affects the end-to-end delay analysis, buffer space requirements, and delay-jitter characteristics.

In this section, we will study the work-conserving disciplines: Delay earliest-due-date (delay-EDD) [16], [29], [69], virtual clock [67], fair queueing (FQ) [12] and its weighted version (WFQ) also called packetized generalized processor sharing (PGPS) [48], self-clocked fair queueing (SCFQ) [22], and worst-case fair weighted fair queueing (WF^{2}Q) [2]. We first describe each of these disciplines, then present a framework to show the similarities and differences among them. Finally, we examine the end-to-end delay characteristics and buffer space requirements for each of them. Nonwork-conserving disciplines will be discussed in Section IV.

A. Virtual Clock

The virtual clock [67] discipline aims to emulate the time division multiplexing (TDM) system. Each packet is allocated a virtual transmission time, which is the time at which the packet would have been transmitted were the server actually doing TDM. Packets are transmitted in the increasing order of virtual transmission times.

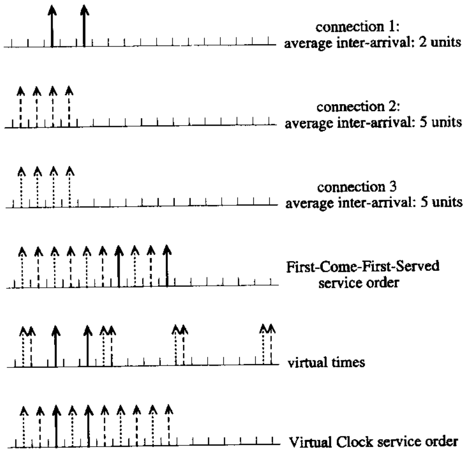


Fig. 3. Comparison of virtual clock and FCFS.

Fig. 3 gives a simple example to illustrate how virtual clock works. In the example, there are three connections sharing the same output link. All three connections specify their traffic characteristics and reserve resources accordingly. Connection 1 has an average packet interarrival time of 2 time units, connection 2 and 3 have an average packet interarrival time of 5 time units. For simplicity, assume packets from all the connections have the same size, and the transmission of one packet takes one time unit. Hence, each of connections 2 and 3 reserve 20% of the link bandwidth, while connection 1 reserves 50% of the link bandwidth. The arrival pattern of the three connections are shown in the first three timelines. As can be seen, connections 2 and 3 send packets at higher rates than reserved, while connection 1 sends packet according to the specified traffic pattern. The fourth timelines show the service order of packets when the service discipline is FCFS. In this case, even though connection 1 reserves more resources, the misbehaviors of connections 2 and 3 affect its performance.

The virtual clock algorithm assigns each packet a virtual transmission time based on the arrival pattern and the reservation of the connection to which the packet belongs. The fifth timeline shows the virtual transmission time assignment. The transmissions of packets are then ordered by the virtual transmission times. The service order of packets under the virtual clock discipline is shown in the sixth timeline. Notice that although connections 2 and 3 are sending packets at higher rates, the virtual clock algorithm ensures that each well behaving connection, in this case connection 1, gets good performance.

B. WFQ and WF^{2}Q

WFQ and WF^{2}Q are two packet policies that try to approximate the same fluid fair queueing (FFQ) or generalized processor sharing (GPS) policy. FFQ is a general form of the head-of-line processor sharing service discipline (HOL-PS) [33]. With HOL-PS, there is a separate FIFO queue for each connection sharing the same link. During any time interval when there are exactly N nonempty queues, the server serves the N packets at the head of the queues simultaneously, each at a rate of one $N\mathrm{th}$ of the link speed. While a HOL-PS server serves all nonempty queues at the same rate, FFQ allows different connections to have different service shares. A FFQ is characterized by N positive real numbers, $\phi_1, \phi_2, \ldots, \phi_N$, each corresponding to one queue. At any time τ, the service rate for a nonempty queue i is exactly $\frac{\phi_i}{\sum_{j \in B(\tau)}} C$ where $B(\tau)$ the set of nonempty queues and C is the link speed. Therefore, FFQ serves the nonempty queues in proportion to their service shares. FFQ is impractical as it assumes that the server can serve all connections with nonempty queues simultaneously and that the traffic is infinitely divisible. In a more realistic packet system, only one connection can receive service at a time and an entire packet must be served before another packet can be served.

There are different ways of approximating FFQ service in a packet system. Among them, the most well known one is the WFQ discipline [12], also known as PGPS [47]. In WFQ, when the server is ready to transmit the next packet at time τ, it picks, among all the packets queued in the system at τ, the first packet that would complete service in the corresponding FFQ system if no additional packets were to arrive after time τ.

While WFQ uses only finish times of packets in the FFQ system, WF^{2}Q uses both start times and finish times of packets in the FFQ system to achieve a more accurate emulation. In WF^{2}Q, when the next packet is chosen for service at time τ, rather than selecting it from among all the packets at the server as in WFQ, the server only considers the set of packets that *have started (and possibly finished) receiving service in the corresponding FFQ system at time* τ, and selects the packet among them that would complete service first in the corresponding FFQ system.

The following example, shown in Fig. 4, illustrates the difference between WFQ and WF^{2}Q. For simplicity, assume all packets have the same size of 1 and the link speed is 1. Also, let the guaranteed rate for connection 1 be 0.5, and the guaranteed rate for each of the other 10 connections be 0.05. In the example, connection 1 sends 11 back-to-back packets starting at time 0 while each of all the other 10 connections sends only one packet at time 0. If the server is FFQ, it will take 2 time units to service each of the first 10 packets on connection 1, one time unit to service the 11th packet, and 20 time units to service the first packet from another connection. Denote the kth packet on connection j to be p_j^k, then in the FFQ system, the starting and finishing times are $2(k-1)$ and $2k$, respectively, for $p_1^k, k = 1 \cdots 10$, 20 and 21, respectively, for p_1^{11}, and 0 and 20, respectively, for $p_j^1, j = 2 \cdots 11$.

For the same arrival pattern, the service orders under the packet WFQ and WF^{2}Q systems are different. Under WFQ, since the first 10 packets on connection 1 ($p_1^k, k = 1 \cdots 10$) all have FFQ finish times smaller than packets on other connections,[2] the server will service 10 packets on connection 1 back to back before service packets from other connections.

Under WF^{2}Q, at time 0, all packets at the head of each connection's queue, p_i^1, $i = 1, \ldots, 11$, have started service in the FFQ system. Among them, p_1^1 has the smallest finish time in FFQ, so it will be served first in WF^{2}Q. At time 1, there are still 11 packets at the head of the queues: p_1^2 and p_i^1, $i = 2, \ldots, 11$. Although p_1^2 has the smallest finish time, it will not start service in FFQ until time 2, therefore, it won't be eligible for transmission at time 1. The other 10 packets have all started service at time 0 at the FFQ system, thus are eligible. Since they all finish at the same time in the FFQ system, the tie-breaking rule of giving highest priority to the connection with the smallest number will yield p_2^1 as the next packet for service. At time 3, p_1^2 becomes eligible and has the smallest finish time among all packets, thus it will start service next. The rest of the sample path for the WF^{2}Q system is shown in Fig. 4.

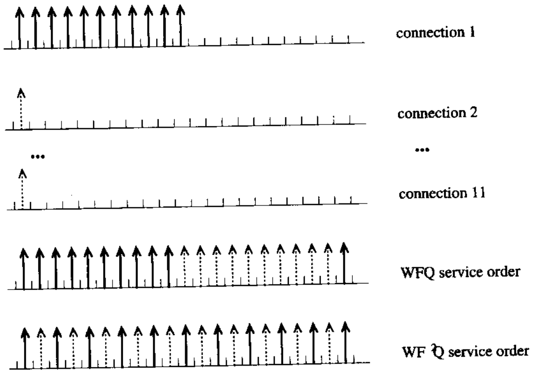


Fig. 4. WFQ and WF^{2}Q.

There are two noteworthy points. First, at any given time τ, the accumulated service provided for each connection (the total amount of bits transmitted) by either packet system *never* falls behind the fluid FFQ system by more than one packet size. Such a relationship with FFQ and the fact that end-to-end delay bounds can be provided in FFQ are the basis for establishing end-to-end delay bounds in WFQ and WF^{2}Q. Also, since in the worst case both WFQ and WF^{2}Q can fall behind FFQ by the same amount of service, they provide the same end-to-end delay bounds. However, as shown in the example, the service order under WFQ and WF^{2}Q can be quite different. Even though WFQ cannot fall much behind FFQ in terms of service, it can be quite far *ahead of* the FFQ system. In the example, by the time 10, 10 packets on connection one have been served in the WFQ system, while only five packets have been served in the FFQ system. The discrepancy between the service provided by WFQ and FFQ can be even larger when there are more connections in the system. In contrast, WF^{2}Q does not have such a problem. In the above example, by the time 10, WF^{2}Q will have served five packets, exactly the same as FFQ. In fact, it can be shown that the difference between the services provided by WF^{2}Q and FFQ is always less than one packet size. Therefore, WF^{2}Q is the most accurate packet discipline that approximates the fluid FFQ discipline.

Even though the difference between WFQ and WF^{2}Q does not affect the end-to-end delay bounds they provide, it is shown in [2] that such a difference may have important implications if they are used to provide best-effort services.

C. Self-Clocked Fair Queueing

Both WFQ and WF^{2}Q need to emulate a reference FFQ server. However, maintaining the reference FFQ server is computationally expensive. One simpler packet approximation algorithm of FFQ is self-clocked fair queueing (SCFQ) [22] also known informally as "Chuck's approximation" [11]. The exact algorithm of SCFQ and the examples illustrating the difference between WFQ and SCFQ will be given in Section III-E.

[2] The FFQ finish time of packet p_1^{10} is 20, the same as that of packets $p_j^1, j = 2 \cdots 11$. If we adopt the following tie-breaking policy in which the packet from the connection with the smallest connection number has a higher priority, packet p_1^{10} will be served before $p_j^1, j = 2 \cdots 11$.

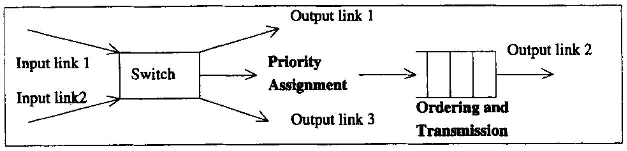


Fig. 5. Sorted priority mechanism.

D. Delay-Earliest-Due-Date

Delay-earliest-due-date algorithm or delay-EDD [16] is an extension to the classic earliest-due-date-first (EDD or EDF) scheduling [41]. In the original EDD, each packet from a periodic traffic stream is assigned a deadline and the packets are sent in order of increasing deadlines. The deadline of a packet is the sum of its arrival time and the period of traffic stream. In delay-EDD, the server negotiates a service contract with each source. The contract states that if a source obeys its promised traffic specification, such as a peak and average sending rate, then the server will provide a delay bound. The key lies in the assignment of deadlines to packets. The server sets a packet's deadline to the time at which it should be sent had it been received according to the contract. This is just the expected arrival time added to the delay bound at the server. For example, if a client assures that it will send packets every 0.2 s, and the delay bound at a server is 1 s, then the kth packet from the client will get a deadline of $0.2k + 1$.

E. Framework for Work-Conserving Disciplines

Virtual clock, delay-EDD, WFQ, WF^{2}Q, and SCFQ all use a similar sorted priority queue mechanism. In such a mechanism, there is a state variable associated with each connection to monitor and enforce its traffic. Upon arrival of each packet from that connection, the variable is updated according to 1) the reservation made for the connection during the connection establishment time and, 2) the traffic arrival history of this connection and/or other connections during the data transfer. The packet is then stamped with the value of the state variable for the connection to which it belongs. The stamped value is used as a priority index. Packets are served in the order of increasing priority index values. This is shown in Fig. 5. WF^{2}Q also needs additional mechanism to mark whether packets are eligible for transmission. As will be discussed in Section IV, this can be implemented with a calendar queue.

In virtual clock, the state variable is called auxiliary virtual clock (auxVC); in WFQ, WF^{2}Q, and SCFQ, it is called the virtual finish time (F); in delay-EDD, it is called Expected Deadline (ExD). In all three cases, auxVC, F and ExD are used as priority indices of packets. The computations of auxVC, F and ExD are based on the formula shown in Table 1. In the table, the subscripts i, j, and k denotes server number, connection number, and packet number, respectively. In delay-EDD, $X\min_j$ is the minimum packet interarrival time for connection j, $d_{i,j}$ is the local delay bound assigned to connection j at server i at connection establishment time. In virtual clock, $V\text{tick}_j$ is the average packet interarrival time for connection j. In WFQ and WF^{2}Q, $V(t)$ is the system *virtual time* at time t, where the virtual time, defined below, is a measure of the system progress. L_j^k is the packet length measured in number of bits, and $a_{i,j}^k$ is arrival time of the kth packet on connection j at switch i.

Table 1 Comparison of Work-Conserving Disciplines

Virtual Clock	$\text{aux}VC_{i,j}^k \leftarrow \max\{a_{i,j}^k, \text{aux}VCk_{i,j}\} + V\text{tick}_{i,j}$
WFQ & WF^{2}Q	$F_{i,j}^k \leftarrow \max\{V_i(a_{i,j}^k), F_{i,j}^{k-1}\} + \frac{L_j^k}{\phi_{i,j}}$
SCFQ	$F_{i,j}^k \leftarrow \max\{\hat{V}_i(a_{i,j}^k), F_{i,j}^{k-1}\} + \frac{L_j^k}{\phi_{i,j}}$
Delay-EDD	$ExD_{i,j}^k \leftarrow$ $\max\{a_{i,j}^k + d_{i,j}, ExDk_{i,j} + X\min_j\}$

As shown in Table 1, the priority index updating algorithms are very similar. However, there are also two important differences. The first is whether the calculation is based on just the rate parameter or both the delay and rate parameters. The second is whether the updating is based on system-load *independent* parameters or system-load *dependent* parameters.

Notice that in delay-EDD, two parameters are used to update the priority index: $X\min_j$, which is the minimum packet inter-arrival time for connection j, and $d_{i,j}$, which is the local delay bound for connection j at switch switch i. In other disciplines, only one rate parameter is used ($V\text{tick}_j$ or $\phi_{i,j}$). Although delay bounds can be provided for all these disciplines, having only one rate parameter introduces the problem of coupling between the allocation of delay bound and bandwidth. For example, in rate-proportional processor sharing (RPPS), which is a special case of WFQ or PGPS and the ϕ's are allocated proportional to the bandwidth required by connections, if the traffic is constrained by (σ, ρ)[3] characterization, the end-to-end delay bound of the connection will be $\frac{\sigma+(n-1)L_{\max}}{\rho} + \sum_{i=1}^n \frac{L_{\max}}{C_i}$, where n is the number of hops traversed by the connection, and C_i is the link speed of the ith server. Notice that the delay bound is inversely proportional to the allocated long term average rate. Thus, in order for a connection to get a low delay bound, a high bandwidth channel needs to be allocated. This will result in a waste of resources when the low delay connection also has a low throughput. Delay-EDD, on the other hand, does not have such a restriction [16], [40].

The second difference between these disciplines is whether the updating of the state variables depends on system load. In virtual clock and delay-EDD, the updating depends only on per connection parameters (Vtick for virtual clock, $X\min$ and d for delay-EDD) but not on system load. In WFQ, WF^{2}Q, and SCFQ, the updating is based on a notion of virtual time. The evolution of virtual time measures the progress of the system and depends on system load. For WFQ and WF^{2}Q, the virtual time function $V(\cdot)$ during any busy period $[t_1, t_2]$ is defined as follows

$$V(t_1) = 0 \tag{1}$$

[3] As mentioned in Section I, σ is the maximum burst size, ρ is the long term average rate.

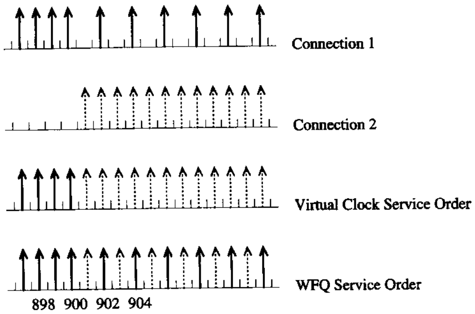


Fig. 6. WFQ and virtual clock.

$$\frac{\partial V(\tau)}{\partial \tau} = \frac{1}{\sum_{j \in B_{\mathrm{FFQ}}(\tau)} \phi_j} \quad \forall t_1 \leq \tau \leq t_2 \qquad (2)$$

where $B_{\mathrm{FFQ}}(\tau)$ is the set of backlogged connections[4] at time τ under the reference FFQ system. In FFQ, if connection j is backlogged at time τ, it receives a service rate of $\frac{\partial V(\tau)}{\partial \tau}\phi_j C$, where C is the link speed. Therefore, V can be interpreted as increasing proportionally to the marginal rate at which backlogged connections receive service in FFQ.

The following example, given in [47] and illustrated in Fig. 6, shows the difference between WFQ and virtual clock. Suppose that there are two connections, both with a specified average rate of one packet every 2 s. All packets are fixed size and require exactly 1 s to service. Starting at time zero, 1000 packets from connection 1 arrive at a rate of 1 packet/s. No packets from connection 1 arrive at the interval (0,900). Starting at time 900, 450 connection 2 packets arrive at a rate of 1 packet/s. Since the first 900 packets from connection 1 are served in the interval (0,900), there are no packets in queue from either connection at time 900^-. If virtual clock algorithm is used, at time 900, the connection 1 auxVC reads 1800 and the connection 2 clock reads 900 (as can be seen in Table 1, the auxVC value is at least the real-time value). When connection 2 packets arrive, they will be stamped 900, 902, ..., 1798, while the connection 1 packets that arrive after time 900 will be stamped 1800, 1804, ..., 1998. Thus *all* of the connection 2 packets will be served before any of the connection 1 packets are served. On the other hand, if WFQ discipline is used, the number of active connections is 1 before time 900 and 2 after time 900. Since both connection 1 and connection reserve half of the link bandwidth, after time 900, the WFQ server will service packets from both connections interleavingly.

The different behaviors of virtual clock and PGPS are due to the fact that virtual clock is defined with reference to the *static* TDM system and the calculation of the virtual transmission time is *independent* of the behaviors of other connections. The delay of a packet depends on the entire arrival history of the connection, which is summarized in the state variable $auxVC$. Once a connection is mishaved (sending more packets than specified), it may be punished by virtual clock, *regardless whether such misbehavior affects the performance of other connections.* WFQ, on the other hand, is defined with reference to another *dynamic* queueing system FFQ. The virtual time of the system *depends* on how many other connections are active in the system.

The dependency on virtual time also introduces extra complexities for WFQ and WF^{2}Q since the system needs to emulate FFQ and keep track of the number of active connections at any moment in FFQ. To reduce the complexity of computing virtual times, SCFQ introduces an approximation algorithm. The algorithm is based on the observation that the system's virtual time at any moment t may be estimated from the virtual service time of the packet currently being serviced. Formally, the approximation virtual time function $\hat{V}(t)$ is defined to be F^p where $s^p < t \leq f^p$, s^p and f^p denote the times packet p starts and finishes service in the SCFQ system.

While the calculation of virtual time is simplier in SCFQ, the inaccuracy incurred can make SCFQ perform much worse than WFQ. Consider the example illustrated in Fig. 7. Again, assume all packets have the same size of 1, the link speed is 1, the guaranteed rate for connection 1 is 0.5, and the guarantee rate for each of the other 10 connections is 0.05. Under FFQ, the finish times will be $2k$ for packets $p_1^k, k = 1 \cdots 10$, 20 for packets $p_j^1, j = 2 \cdots 11$, and 21 for p_1^{11}. Transmitting packets in order of finish times under FFQ, WFQ will produce the service order as shown on the fourth timeline in Fig. 7. If SCFQ is used, at time 0, same as in WFQ, it is p_1^1 that has the smallest virtual finish time, therefore, it receives service first. At time 1, all packets $p_i^1, i = 2, \ldots, 11$, have virtual finish time of $F_i^1 = 20$. With the tie-breaking rule, the first packet on connection 2, p_2^1, is served. Since SCFQ uses the finish time of the packet in service as the the current virtual time value, we have $\hat{V}(1) = \hat{V}(2) = F_2^1 = 20$. As a result when p_1^2 arrives at time 2, its virtual finish time is set to be: $F_1^2 = \max(F_1^1, \hat{V}(2)) + \frac{L}{r_1} = \max(2, 20) + 2 = 22$. Among all the packets ready to be served, p_1^2 has the largest finish number. Therefore, p_1^2 won't start service until all other 10 $p_i^1, i = 2, \ldots, 11$, packets finish services, i.e., it won't depart until time 12. In this example, even though connection 1 sends packet according to the specified average rate, its packets still get significantly delayed.

F. Traffic Characterization Inside the Network

As discussed in Section I, we are interested in providing end-to-end delay bounds on a per connection basis in a networking environment. One solution is to obtain worst-case local delay bounds at each switch independently and use the sum of these local delay bounds as the end-to-end delay bound [16]. Alternatively, smaller end-to-end delay bounds can be obtained by taking into account the dependencies in the sucessive switches that a connection traverses [10], [17], [19], [23], [47], [68]. For the first type of solution, in order to derive local delay bound, traffic

[4] A connection is said to be backlogged at time τ if it has packets queued at time τ.

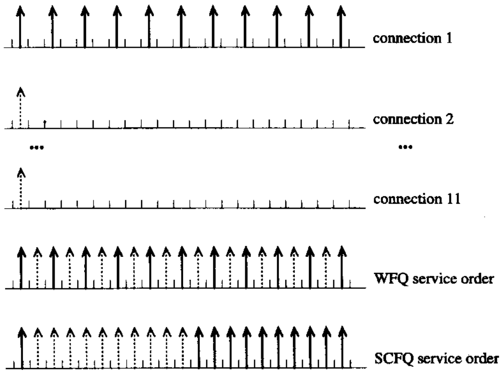


Fig. 7. SCFQ and WFQ.

needs to be characterized on a per connection basis at each switch inside the network. For the second type of solution, while the end-to-end delay bound may be derived for virtual clock, WFQ, SCFQ based only on the source traffic characterization [10], [17], [23], [47], as will be shown in Section III-G, the delay bound couples with bandwidth allocation. In [47], such a resource allocation strategy is called rate-proportional allocation. It has been shown more general resource allocation strategies that decouples throughtput and delay bounds can result in higher utilization of the network. In general, for both types of solutions, the traffic needs to be characterized on a per connection basis at each switch inside the network.

The difficulty arises in a networking environment, where even if a connection's traffic can be characterized at the entrance to the network, traffic pattern may be distorted inside the network, thus make the source characterization not applicable at the servers traversed by the connection. This is shown in the following example illustrated by Fig. 8. In the example, four packets from one connection are sent with a certain interpacket spacing from the source into a network where links have constant delay. At the first server, the first packet is delayed by a certain amount of time (less than the local delay bound) due to instantaneous cross traffic load, but the other three packets pass through instantly. Because the first packet was delayed longer than the second packet, the spacing between first and second packets becomes smaller when they arrive at the second server. At the second server, the first and the second packet are delayed some more time, but packets three and four pass through instantly. At the third server, the first three packets are delayed but packet four passes through with no delay. The figure shows traffic patterns at the entrance to each of the servers. Two things can be observed: 1) the traffic pattern of a connection can be distorted due to network load fluctuations, 2) the distortion may make the traffic burstier and cause instantaneously higher rates. In the worst case, the distortion can be accumulated, and downstream servers potentially face burstier traffic than upstream servers. Therefore, the source traffic characterization may not be applicable inside the network.

There are three solutions to address this problem of traffic pattern distortion:

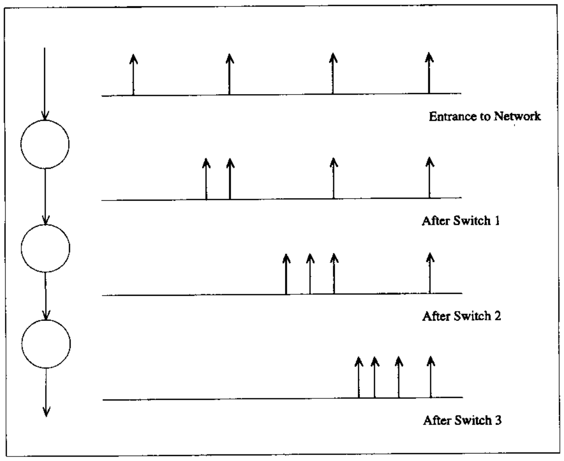


Fig. 8. Traffic pattern distortions due to load fluctuations.

1) controlling the traffic distortion within the network,
2) accounting for the distortion during scheduling,
3) characterizing the traffic distortion throughout the network.

To control traffic distortions within the network, some packets need to be held even when a server has the extra capacity. This requires nonwork-conserving disciplines, which we will discuss in more detail in Section IV.

The second solution accounts for traffic distortions during scheduling. Instead of scheduling packets according to their actual arrival times, the server assigns each packet a logical arrival time based on its traffic characterization and previous arrival history, and schedules packets based on their logical arrival times. Delay-EDD and virtual clock use such an approach. For example, in delay-EDD, the deadline of a packet is the sum of the local delay bound (d) and the expected arrival time of the packet. The service discipline and the admission control policy ensure that the packet is guaranteed to leave before the deadline, or at most d time units after the expected arrival time of the packet. It is possible that a packet is delayed longer in a server than its local delay bound. However, this can only happen if the packet's expected arrival time is larger than its actual arrival time, which means that the packet is ahead of schedule in previous servers. It can be shown that the amount of the time the packet is queued at the server more than its delay bound is always less than the amount of time the packet is ahead of schedule at previous servers.

Accounting for traffic distorting during scheduling works only if the server has a concept of expected arrival time. A more general solution is to characterize the traffic inside the network. The problem can be formulated as the following: given the traffic characterization of all the connections at the entrance to the network and all the service disciplines at the switches, can the traffic be characterized on a per connection basis on all the links inside the network? Several solutions have been proposed to address this problem with different traffic models and service disciplines [1], [8], [37], [47]. They all employ a similar technique that consists of two steps. In the first step, a single node analysis technique is developed to characterize the output traffic of a server given the characterizations of all its input traffic. In the second step, starting from the characterizations of all the source traffic, an iterative process push the traffic characterizations from the links at the edge of the network to those inside the network. There are several limitations associated with such an approach.

First, characterizing the traffic inside the network is difficult and may not always be possible. In [9], [37], [49], it is shown that this is equivalent to solving a set of multivariable equations. In a feedback network, where traffic from different connections forms traffic loops, the resulting set of equations may be unsolvable. To illustrate this, consider the following example discussed in [9], [47].

In the four-nodes network shown in Fig. 9, there are four three-hop connections and the aggregate traffic of the four connections forms a loop. In order to characterize the traffic on link 1, the characterization of the input traffic to server 1 has to be known. Assuming links only introduce fixed delay, the input traffic to server 1 is identical to the output traffic of server 0, or the traffic on link 0. There are three traffic streams on link 0, which are from connections 0, 2, and 3. While connection 0 traffic on link 0 is the same as its source traffic, connection 2 and connection 3 traffic on link 0 needs to be characterized. The characterizations of connection 2 and 3 traffic depend on their characterizations on link 3, which in turn depend on their characterizations on link 2. This dependency finally comes back to traffic characterizations on link 0. Because of this interdependency of traffic, characterizing all the traffic inside the network is equivalent to solving a set of multivariable equations, where each variable corresponds to one parameter in the traffic characterization of one connection on one link. The equations are solvable only under certain conditions. In this particular example, it is shown in [9] that if each server has a policy such that the traffic originating from the server has a lower priority than the through traffic, the condition for solving the equations is that the aggregate throughput of all connections must be less than 75% of the link bandwidth on each of the four links. This condition is not merely a technical restriction, the network *can* actually be unstable, i.e., have unbounded queue lengths, when the condition is violated [47]. How to derive the stability condition in a general networking environment is still a open problem. A distributed algorithm is even more difficult. One notable exception to such a restriction is the case when the service discipline used is a special class of PGPS called rate proportional processor sharing (RPPS) [47]. With RPPS, the stability condition can be derived. We will discuss the exact formula of the delay bound in Section III-G.

The second limitation of characterizing traffic inside the network is that it only applies to networks with *constant* delay links. Constant delay links have the desirable property that the traffic pattern at the receiving end of the link is the same as that at the transmitting end. This property is important for these solutions because central to the analysis is the technique of characterizing the output traffic from a server and using it as the characterization of the input traffic to the next-hop server. However, in an internetworking environment, where the link between two switches may be a subnetwork such as an ATM network or a FDDI network [14], load fluctuations within subnetworks may also introduce traffic pattern distortions. Though it is possible to bound delay over these subnetworks, the delays for different packets will be *variable*. Thus these solutions need to be extended in order to be applicable in an internetworking environment.

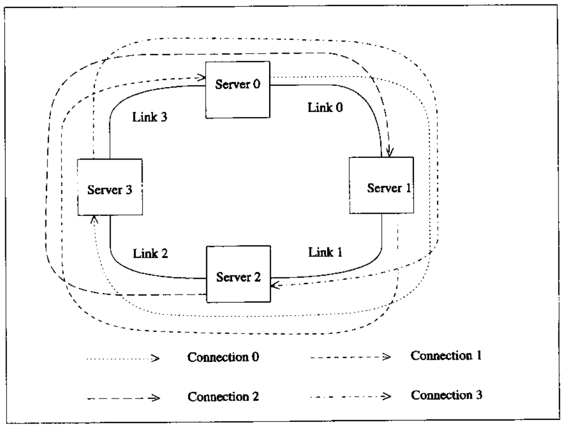


Fig. 9. Example of a feedback network.

Finally, in networks with work-conserving service disciplines, even in situations when traffic inside the network can be characterized, the characterization usually represents a burstier traffic inside the network than that at the entrance. This is independent of the traffic model being used. In [8], it is shown that if the traffic of connection j is characterized by (σ_j, ρ_j) at the entrance to the network, its characterization will be $(\sigma_j + \sum_{h=1}^{i-1} \rho_j d_{h,j}, \rho_j)$ at the entrance to the ith server along the path, where $d_{h,j}$ is the local delay bound for the connection at the hth server. Compared to the characterization of the source traffic, the maximum burst size increases by $\sum_{h=1}^{i-1} \rho_j d_{h,j}$. This increase of burst size grows monotonically along the path.

In [37], a family of stochastic random variables are used to characterize the source. Connection j is said to satisfy a characterization of $\{(\mathbf{R}_{t_1,j}, t_1), (\mathbf{R}_{t_2,j}, t_2), (\mathbf{R}_{t_3,j}, t_3) \cdots\}$, where $\mathbf{R}_{t_i,j}$ are random variables and $t_1 < t_2 < \cdots$ are time intervals, if $\mathbf{R}_{t_i,j}$ is *stochastically larger* than the number of packets generated over any interval of length t_i by source j. If the traffic connection j is characterized by $\{(\mathbf{R}_{t_1,j}, t_1), (\mathbf{R}_{t_2,j}, t_2), \ldots\}$ at the entrance to the network, its characterization will be $\{(\mathbf{R}_{t_1+\sum_{h=1}^{i-1} b_h, j}, t_1), (\mathbf{R}_{t_2+\sum_{h=1}^{i-1} b_h, j}, t_2), \ldots\}$ at the hth switch, where b_h is the length of the maximum busy period at switch h. The same random variable $\mathbf{R}_{t_m+\sum_{h=1}^{i-1} b_h, j}$ that bounds the maximum number of packets over an interval of length $t_m + \sum_{h=1}^{i-1} b_h$ at the entrance to the network, now bounds the maximum number of packets over a much *smaller* interval of length t_m at server i. In other words, the traffic characterization is burstier at server i than at the entrance.

Table 2 End-to-End Delay, Bound Delay, Delay-Jitter, and Buffer Space Requirements

	traffic constraint	end-to-end delay bound	end-to-end delay-jitter bound	buffer space at h^{th} switch
D-EDD	$b_j(\cdot)$	$\sum_{i=1}^{n} d_{i,j}$	$\sum_{i=1}^{n} d_{i,j}$	$b_j(\sum_{i=1}^{h} d_{i,j})$
FFQ	(σ_j, ρ_j)	$\frac{\sigma_j}{r_j}$	$\frac{\sigma_j}{r_j}$	σ_j
VC	(σ_j, ρ_j)	$\frac{\sigma_j + nL_{\max}}{r_j} + \sum_{i=1}^{n} \frac{L_{\max}}{C_i}$	$\frac{\sigma_j + nL_{\max}}{r_j}$	$\sigma_j + hL_{\max}$
WFQ & WF²Q	(σ_j, ρ_j)	$\frac{\sigma_j + nL_{\max}}{r_j} + \sum_{i=1}^{n} \frac{L_{\max}}{C_i}$	$\frac{\sigma_j + nL_{\max}}{r_j}$	$\sigma_j + hL_{\max}$
SCFQ	(σ_j, ρ_j)	$\frac{\sigma_j + nL_{\max}}{r_j} + \sum_{i=1}^{n} K_i \frac{L_{\max}}{C_i}$	$\frac{\sigma_j + nL_{\max}}{r_j} + \sum_{i=1}^{n} (K_i - 1) \frac{L_{\max}}{C_i}$	$\sigma_j + hL_{\max}$

G. End-to-End Delay Characteristics and Buffer Space Requirement

While the problem of deriving end-to-end delay bounds for a network of work-conserving servers has yet to be solved under general resource assignments, results have been obtained for virtual clock, WFQ, WF^{2}Q, SCFQ under the rate-proportional allocation strategy, and for delay-EDD by considering each server in isolation. In both cases, the source traffic specifications are sufficient and traffic characterizations inside the network are not needed. In the former case, the end-to-end delay bound for a connection is a function of the guaranteed rate, which needs to be no less than the connection's average rate. In the latter case, the end-to-end delay bound is calculated as the sum of worst-case local delays at each switch. Since delay-EDD scheduling is based on logical rather actual packet arrival times, local delay bounds at all switches can be calculated using the same source traffic characterization. To prevent packet loss, we assume buffer space is allocated on a per connection basis at each server during connection establishment time.

Table 2 presents the end-to-end characteristics and buffer space requirement of a connection when different work-conserving service disciplines are used. The table can be interpreted as the following. If a connection satisfies the traffic constraint as defined in the second column, and is allocated the amount of buffer space as listed in the fifth column, it can be guaranteed an end-to-end delay bound and delay-jitter bound as listed in the third and fourth column, respectively, provided each server along the path uses the discipline in first column and appropriate admission control conditions are satisfied. In the table, C_i is link speed of the ith switch on the path traversed by the connection, K_i is the number of connections sharing the link with the connection at the ith switch, r_j is the guaranteed rate for the connection, and $L_{\max}$ is the largest packet size. Link delays are omitted from the expressions of end-to-end delays for simplicity.

Notice that the (σ, ρ) traffic model is used to characterize the traffic in all disciplines except delay-EDD where a general traffic constraint function is used. The original delay-EDD uses the $(X\min, X\text{ave}, I, S\max)$ traffic model [16], [69]. However, the algorithm can easily be extended to accommodate connections using arbitrary deterministic traffic models that have associated traffic constraint functions. The corresponding admission control algorithm is described in [40]. A more general traffic model can characterize sources more accurately, thus resulting in a higher network utilization. A more detailed discussion on the relationship between achievable network utilization and accuracy of traffic characterization can be found in [34], [35].

There are several noteworthy points about the table. First, even though virtual clock, WFQ, and WF^{2}Q have a number of differences, they provide identical end-to-end delay bounds for connections that have leaky bucket constrained sources. In fact, if we compare the delay bound provided by them and that provided by the ideal fluid FFQ discipline, we can see that they share the same main term $\frac{\sigma_j}{r_j}$, which can be interpreted as the time to send a burst of size σ_j in a fluid system with the guaranteed rate of r_j. For the three packet policies, there are additional terms to account for the fact that traffic is not infinitely divisible and the server needs to serve one packet at a time. Secondly, with the same guaranteed rate, the delay bound provided by SCFQ is larger than that provided by virtual clock, WFQ, and WF^{2}Q. This is due to the inaccuracy introduced by the approximation algorithm in calculating the virtual time. For all the four disciplines, since the server allocates service shares to connections proportional to their average rates, there is a coupling between the end-to-end delay bound and bandwidth provided to each connection. In particular, the end-to-end delay bound is inversely proportional to the allocated long term average rate. Thus, in order for a connection to get a low delay bound, a high bandwidth channel need to be allocated. This will result in a waste of resources when the low delay connection also has low throughput. WFQ and WF^{2}Q with general resource assignments do not have such a restriction [47]. However, due to the difficulties of characterizing traffic inside the network, the problem of deriving end-to-end delay bound for WFQ and WF^{2}Q under general resource assignments has yet to be solved. Delay-EDD does not have the problem of coupling between the allocations of delay bound and bandwidth either. However, the end-to-end delay bound listed in the table was derived without taking into account the delay dependency among successive switches, and is rather loose. As a final point to be noted, the end-to-end delay-jitter bounds for all disciplines are loose. In fact, the end-to-end delay-jitter bound is equal to the maximum end-to-end queueing delay. This can be easily understood by the following observation. Recall that delay-jitter bound is the maximum difference between delays of any two packets.

In a network with work-conserving disciplines, a packet can experience little queueing delay when the network is lightly loaded while another packet can experience a much longer queueing delay when the network is heavily loaded. Thus the maximum difference between delays experienced by these two packets is the maximum end-to-end queueing delay.

H. Implementation Issues

As described in Section III-E, all the proposed work-conserving disciplines use the mechanism of a sorted priority queue. The insertion operation for a sorted priority queue has an intrinsic complexity of $O(\log N)$ [36], where N is the number of packets in the queue. In a network that is designed to support many connections with bursty traffic, the switch usually has buffer space for a large number of packets. For example, the queue module of each link of the Xunet switch contains memory to store 512 K ATM cells [28]. Potentially, the queue length can be long. It may not be feasible to implement an operation that has an $O(\log N)$ complexity at very high speed. Since all disciplines ensure that packets on the same connection are serviced in the order of their arrivals, a clever implementation can arrange packets on a per-connection basis and sort only the first packet of each connection. Recently, it was reported that a sequencer chip clocked at 33 MHz has been implemented to support sorting of up to 256 packets [6]. Thus up to 256 connections or classes of connections can be supported with such an implementation. It is unclear whether such an implementation can scale to higher speed or more connections.

A sorted priority queue mechanism also requires computation of the priority index on a per packet basis. For service disciplines that use real time to compute the priority index, such as virtual clock and delay-EDD, the computation is simple and straightforward. For service disciplines that use virtual times in another reference queueing system, such as WFQ and WF^{2}Q, the computation is more complex. In particular, both WFQ and WF^{2}Q need to keep track the set of connections that have packets queued at any time instant. This is very difficult to implement at high speed. SCFQ simplifies the computation by using an approximation algorithm that does not need to keep track of the set of active connections.

IV. Non-Work-Conserving Disciplines

In Section III-F, we showed that in order to derive end-to-end delay bounds and buffer space requirements in a networking environment, traffic needs to be characterized inside the network on a per connection basis. With work-conserving disciplines, the traffic pattern is distorted inside the network due to network load fluctuation, and there are a number of difficulties and limitations in deriving the traffic characterization after the distortion.

Another approach to deal with the problem of traffic pattern distortions is to control the distortions at each switch using *nonwork-conserving disciplines.* With a nonwork-conserving discipline, the server may be idle even when there are packets waiting to be sent. Nonwork-conserving disciplines were seldom studied in the past. This is mainly due to two reasons. First, in most of previous performance analyses, the major performance indices are the *average* delay of all packets and the *average* throughput of the server. With a nonwork-conserving discipline, a packet may be held in the server even when the server is idle. This may increase the average delay of packets and decrease the average throughput of the server. Secondly, most previous queueing analyses assumed a single server environment. The potential advantages of nonwork-conserving disciplines in a networking environment were therefore not realized. In guaranteed performance service, the more important performance index is the end-to-end delay *bound* rather than the average delay. In addition, delay needs to be bounded in a *networking* environment rather than just in a single node. Therefore, the above reasons for not using nonwork-conserving disciplines do not hold any more.

Several nonwork-conserving disciplines have been proposed in the context of high speed integrated services networks. Among them are: Jitter earliest-due-date (jitter-EDD) [56], stop-and-go [21], hierarchical round robin (HRR) [26], and rate-controlled static priority (RCSP) [62]. In this section, we first describe each of the algorithms in turn, then present a unified framework called rate-controlled service disciplines and show that all of them can be represented in such a framework. Finally, we discuss the end-to-end delay characteristics and buffer space requirements for these disciplines within the framework of rate-controlled service disciplines.

A. Jitter-Earliest-Due-Date

The jitter-EDD discipline [56] extends delay-EDD to provide delay-jitter bounds (that is, a bound on the maximum delay difference between two packets). After a packet has been served at each server, a field in its header is stamped with the difference between its deadline and the actual finishing time. A regulator at the entrance of the next server holds the packet for this period before it is made eligible to be scheduled.

Jitter-EDD is illustrated in Fig. 10, which shows the progress of a packet through two adjacent servers. In the first server, the packet got served *PreAhead* seconds before its deadline. So, in the next server, it is made eligible to be sent only after *PreAhead* seconds. Since there is a constant delay between the eligibility times of the packet at two adjacent servers, the packet stream can be provided a delay jitter bound. Assuming there is no regulator at the destination host, the end-to-end delay jitter bound is the same as the local delay bound at the last server.

B. Stop-and-Go

As shown in Fig. 11, stop-and-go uses a framing strategy [20]. In such a strategy, the time axis is divided into frames, which are periods of some constant length T. Stop-and-go defines *departing* and *arriving* frames for each link. At each

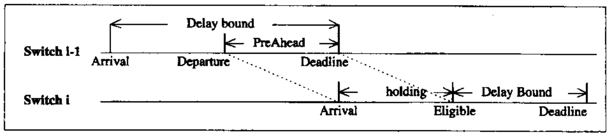


Fig. 10. Packet service in jitter-EDD.

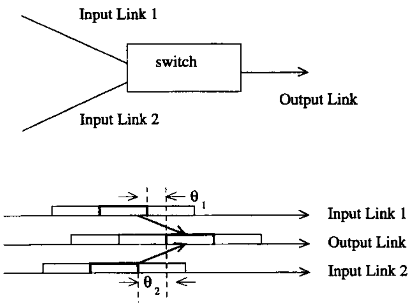


Fig. 11. Synchronization between input and output links in stop-and-go.

switch, the arriving frame of each incoming link is mapped to the departing frame of the output link by introducing a constant delay θ, where $0 \leq \theta < T$. According to the stop-and-go discipline, the transmission of a packet that has arrived on any link l during a frame f should always be postponed until the beginning of the next frame. Since packets arriving during a frame f of the output link are not eligible for transmission until the next frame, the output link may be left idle even when there are packets in the switch to be transmitted, thus stop-and-go is a nonwork-conserving policy.

Stop-and-go ensures that packets on the same frame at the source stay in the same frame throughout the network. If the traffic is characterized at the source by (r, T), i.e., no more than $r \cdot T$ bits are transmitted during any frame of size T, it satisfies the same characterization throughout the network. By maintaining traffic characteristics throughout the network, end-to-end delay bounds can be guaranteed in a network of arbitrary topology as long as each local server can ensure local delay bounds for traffic characterized by (r, T) specification.

The framing strategy introduces the problem of coupling between delay bound and bandwidth allocation granularity. The delay of any packet at a single switch is bounded by two frame times. To reduce the delay, a smaller T is desired. However, since T is also used to specify traffic, it is tied to bandwidth allocation granularity. Assuming a fixed packet size P, the minimum granularity of bandwidth allocation is $\frac{P}{T}$. To have more flexibility in allocating bandwidth, or a smaller bandwidth allocation granularity, a larger T is preferred. It is clear that low delay bound and fine granularity of bandwidth allocation cannot be achieved simultaneously in a framing strategy like stop-and-go.

To get around this coupling problem, a generalized version of stop-and-go with multiple frame sizes is proposed. In the generalized stop-and-go, the time axis is divided into a hierarchical framing structure as shown in Fig. 12. For a n level framing with frame sizes $T_1, \ldots, T_n$, and $T_{m+1} = K_m T_m$ for $m = 1, \ldots, n-1$, packets on a level p connection need to observe the stop-and-go rule with frame size T_p, i.e., level p packets which arrived at an output link during a T_p frame will not become eligible for transmission until the start of next T_p frame. Also, for two packets with different frame sizes, the packet with a smaller frame size has a nonpreemptive priority over the packet with a larger frame size. With multiframe stop-and-go, it is possible to provide low delay bounds to some channels by putting them in frames with a smaller frame time, and to allocate bandwidth with fine granularity to other channels by putting them in levels with a larger frame time. However, the coupling between delay and bandwidth allocation granularity still exists within each frame. In [52], a scheme is proposed to add a separate shaping mechanism at the network entry point for networks with framing based disciplines. With traffic shaping at the entrance to the network, it is possible to multiplex several connections on a single slot of a frame, therefore avoid the problem of coupling between frame size and bandwidth allocation granularity.

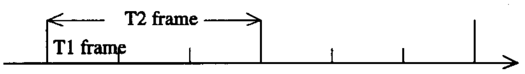


Fig. 12. Two levels of framing with $T_2 = 3T_1$.

C. Hierarchical Round Robin

HRR is similar to stop-and-go in that it also uses a multilevel framing strategy. A slot in one level can either be allocated to a connection or to a lower level frame. The server cycles through the frame and services packets according to the assignment of slots. If the server cycles through a slot assigned to a connection, one packet from that connection is transmitted; if it cycles through a slot assigned to a lower level frame, it will service one slot from the lower level frame in the same fashion. HRR is nonwork-conserving in the sense that if it cycles through a slot with no packets waiting, it will leave the server idle for that slot time rather than sending packets assigned to other slots.

Similar to stop-and-go, HRR also maintains traffic smoothness inside the network due to its nonwork-conserving nature. However, there are also important differences between HRR and stop-and-go. The example shown in Fig. 13 illustrates their difference. In the example, it is assumed that three packet transmission times are allocated to a connection in each frame. In stop-and-go, packets that are transmitted in the same frame at the entrance to the network will be transmitted in the same frame on all the links traversed by the connection. The difference between delays experienced by any two packets from the source to any server is bounded by T, where T is the frame size. In HRR, packets that are transmitted in the same frame at the entrance to the network do not necessarily stay in the same frame inside the network; however, the property that *no more than three packets from*

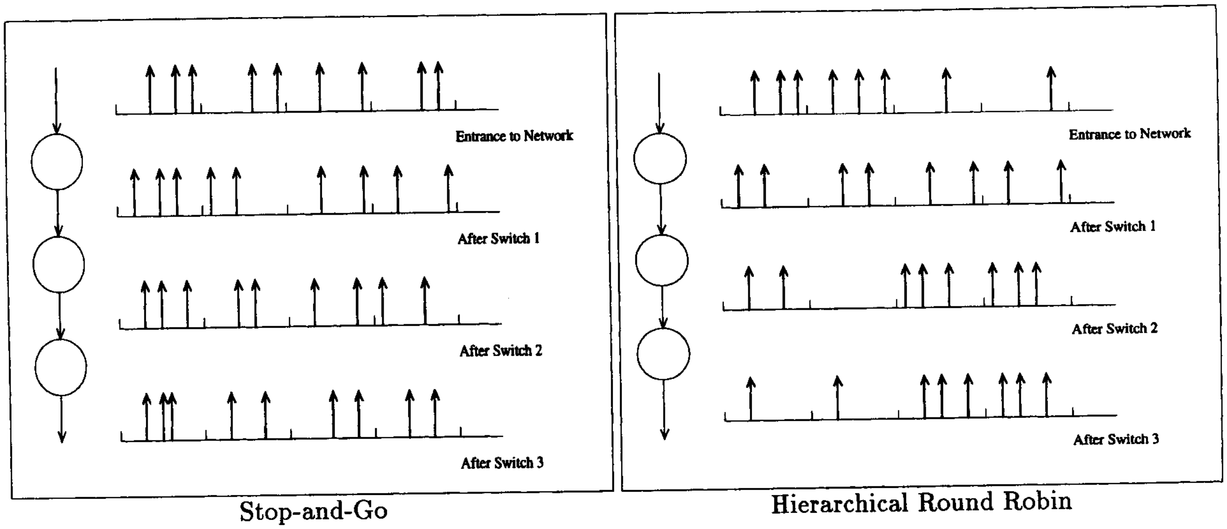


Fig. 13. Difference between stop-and-go and HRR.

the connection are transmitted during one frame time holds throughout the network.

Since HRR uses the framing strategy, it also has the problem of coupling between delay and bandwidth allocation granularity.

D. Rate-Controlled Static Priority

While the Earliest-Due-Date algorithm can provide flexible delay bounds and bandwidth allocation, it is based on a sorted priority mechanism, which is difficult to implement. Stop-and-go and HRR use a framing strategy instead of the sorted priority to achieve simplicity, however, such a strategy introduces coupling between delay bound and bandwidth allocation granularity. The goal of RCSP is to achieve flexibility in the allocation of delay and bandwidth as well as simplicity of implementation.

As shown in Fig. 14, a RCSP server has two components: a rate-controller and a static priority scheduler. Conceptually, a rate controller consists of a set of regulators corresponding to each of the connections traversing the server; each regulator is responsible for shaping the input traffic of the corresponding connection into the desired traffic pattern. Upon arrival of each packet, an eligibility time is calculated and assigned to the packet by the regulator. The packet is held in the regulator till its eligibility time before being handed to the scheduler for transmission. Different ways of calculating the eligibility time of a packet result in different types of regulators. As will be discussed in [61] and Section IV-F, many regulators can be used for RCSP. We will consider two examples in this section. The $(X\min, X\text{ave}, I)$ RJ regulator ensures that the output of the regulator satisfy the $(X\min, X\text{ave}, I)$ traffic model, while the DJ$_r$ regulator ensures that the output traffic of the regulator is exactly the same as the the output traffic of the regulator at the previous server. Thus, if the traffic satisfies the $(X\min, X\text{ave}, I)$ characterization at network entrance, both types of regulators will ensure that the output of the regulator, which is the input to the scheduler, will satisfy the same traffic characterization.

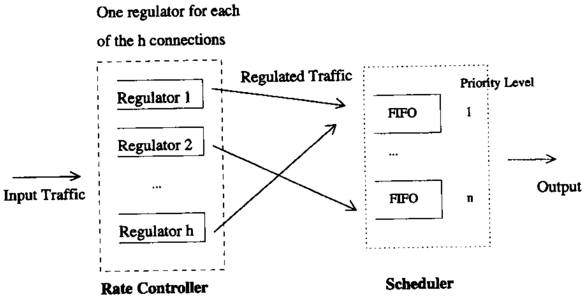


Fig. 14. Rate-controlled static priority.

For a $(X\min, X\text{ave}, I)$ RJ regulator, where $X\min \leq X\text{ave} < I$ holds, the eligibility time of the kth packet on connection j at the ith server along its path, e_i^k, is defined with reference to the eligibility times of packets arriving earlier at the server on the same connection

$$e_{i,j}^k = -I, \quad k < 0 \tag{3}$$

$$e_{i,j}^1 = a_{i,j}^1 \tag{4}$$

$$e_{i,j}^k = \max(e_{i,j}^{k-1} + X\min, \\ e_{i,j}^{k-\lfloor \frac{I}{X\text{ave}} \rfloor+1} + I, \ a_{i,j}^k), \quad k > 1 \tag{5}$$

where $a_{i,j}^k$ is the time the kth packet on connection j arrived at the ith server. (3) is defined for convenience so that (5) holds for any $k > 1$.

From this definition, we can see that $e_{i,j}^k \geq a_{i,j}^k$ always holds, i.e., a packet is never eligible before its arrival. Also, if we consider the sequence of packet eligibility times at ith server , $\{e_{i,j}^k\}_{k=1,2,\ldots}$, it always satisfies the $(X_{\min}, X_{\text{ave}}, I)$ traffic characterization.

The eligibility time of a packet for a DJ$_r$ regulator is defined with reference to the eligibility time of the same packet at the immediately upstream server. The definition assumes that the queueing delays of packets on the connection, and the link delay from the upstream server to the

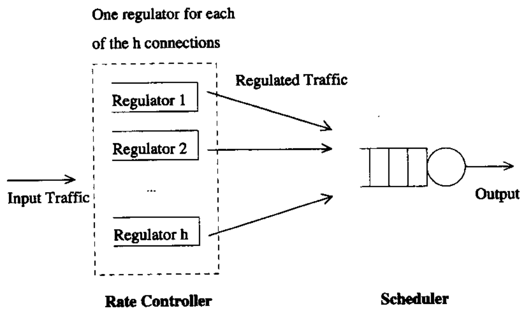


Fig. 15. Rate-controlled service disciplines.

current server, are bounded. Let $d_{i-1,j}$ be the local delay bound for the connection in the scheduler at the $(i-1)$th server, and π_i be the maximum link delay from the $(i-1)$th server to the ith server. The DJ$_r$ regulator is defined as

$$e^k_{0,j} = a^k_{0,j} \tag{6}$$

$$e^k_{i,j} = e^k_{i-1,j} + d_{i-1,j} + \pi_i, \quad i > 0. \tag{7}$$

It is easy to show that the following holds

$$e^{k+1}_{i,j} - e^k_{i,j} = a^{k+1}_{0,j} - a^k_{0,j} \quad \forall k, i \geq 0 \tag{8}$$

i.e., the traffic pattern on a connection at the output of the regulator of every server traversed by the connection is exactly the same as the traffic pattern of the connection at the *entrance* to the network.

The scheduler in a server RCSP uses a nonpreemptive Static Priority policy: it always selects the packet at the head of highest priority queue that is not empty. The SP scheduler has a number of priority levels with each priority level corresponding to a delay bound. Each connection is assigned to a priority level during connection establishment time. Multiple connections can be assigned to the same priority level, and all packets on the connections associated with a priority level are appended to the end of the queue for that priority level.

E. A Framework for Nonwork-Conserving Disciplines

In previous sections, we described four nonwork-conserving disciplines. In this section, we show that all of them can be expressed by a general class of disciplines called rate-controlled service disciplines [64]. As shown in Fig. 15, a rate-controlled server can be considered as a generalization of RCSP: it also has two components, a rate-controller and a scheduler. The rate controller, which consists of a number of regulators, is responsible for shaping traffic. The scheduler is responsible for multiplexing eligible packets coming from different regulators. While RCSP uses two types of regulators and the Static Priority scheduler, many other regulators and schedulers can be used. By having different combinations of regulators and schedulers, we have a general class of disciplines. Among the four disciplines discussed in this section, RCSP and jitter-EDD are rate-controlled servers, stop-and-go and HRR can be implemented with rate-controlled servers by selecting appropriate regulators and schedulers.

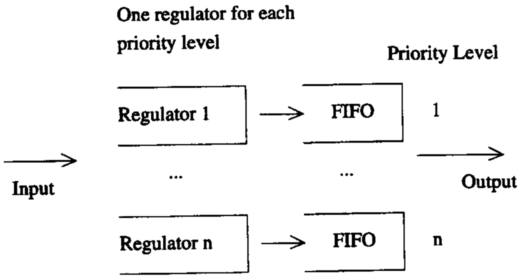


Fig. 16. Implement stop-and-go using a rate-controlled server.

Jitter-EDD can be viewed as a combination of a earliest-due-date scheduler and DJ$_e$ regulators, which are defined as follows

$$e^k_{i,j} = a^k_{i,j} + \text{Ahead}^k_{i-1,j} \tag{9}$$

where $\text{Ahead}^k_{i-1,j}$ is the amount of time the packet is ahead of schedule at the $(i-1)$th server along the path.

A stop-and-go server with n frame sizes ($T_1 < T_2 < \cdots < T_n$) can be implemented by a rate-controlled server with an n-level static priority scheduler and DJ$_s$ regulators

$$e^k_{i,j} = a^k_{i,j} + \text{Ahead}^k_{i-1,j} + \theta_{i,j} \tag{10}$$

where $\text{Ahead}^k_{i-1,j}$ is the amount of time the packet is ahead of schedule in switch $i-1$, and $\theta_{i,j}$ is the synchronization time between the framing structures on the input and output links. Each pair of input and output links in a switch may have a different value of θ. Fig. 11 illustrates this synchronization time. In the static priority scheduler, the delay bound associated with level m is T_m, $1 \leq m \leq n$.

Although the above implementation of stop-and-go is very similar to RCSP, there are also important differences, as can be seen by comparing Fig. 14 and Fig. 16. In an RCSP server, there is a regulator for each connection, and the regulated traffic on each connection can be assigned to *any* priority level in the scheduler. In a stop-and-go server, regulators are associated with priority levels in the scheduler. In fact, there is a one-to-one correspondence between the regulator and the priority level. The traffic on a connection has to be specified with respect to the frame size, which is the same as the connection's local delay bound. This not only introduces the coupling between the allocations of bandwidth and delay bounds, but also implies that admission control algorithm has to be based on a busy period argument, which tends to produce looser bounds when compared to more elaborate analysis [8], [63].

Because of the framing, there are dependencies among the local delay bounds at each priority level in a stop-and-go server. In particular, $T_{m+1} = K_m T_m$ must hold, with $1 \leq m < n$, and K_m being an integer. Furthermore, the delay bound allocations for each connection in different servers are coupled with one another. In [21], a connection has to have the same frame size in all the servers. In [65], a looser requirement is presented: the frame times of a connection along the path should be nondecreasing. None of these restrictions apply to RCSP. The impact of flexibility

Table 3 Nonwork Conserving Disciplines

Discipline	$e_{i,j}^k$ defined in regulator	Scheduler
RCSP/DJ_r	$a_0^k + \text{ahead}_{i-1,j}^k + (\pi_i - \pi_i^k)$	SP
	$e_{i-1}^k + d_{i-1,j} + \pi_i^k$	
Jitter-EDD	$a_{i,j}^k + \text{ahead}_{i-1,j}^k$	EDD
	$e_{i-1}^k + d_{i-1,j} + \pi_i^k$	
Stop-and-Go	$a_{i,j}^k + \text{ahead}_{i-1,j}^k + \theta$	SP
	$e_{i-1}^k + T^m + \pi_i^k$	
RCSP/RJ_r	$\max(e_{i,j}^{k-1} + X\min_j,$ $e_{i,j}^{k-\lfloor \frac{I}{Xave_j} \rfloor+1} + Ia_{i,j}^k)$	SP
HRR	$\max(a_{i,j}^k, e_{i,j}^{k-a_{i,j}} + T_j^m)$	SP

of allocating delay bounds inside the network on network utilization was studied in [45].

A Hierarchical Round Robin server with n frame sizes ($T_1 < T_2 < \cdots < T_n$) can be implemented by a rate-controlled server with an n-level static priority scheduler and RJ$_h$ regulators defined by

$$e_{i,j}^k = \max(a_{i,j}^k + \tau, e_{i,j}^{k-q_{i,j}^m} + T_m) \qquad (11)$$

where $a_{i,j}^k + \tau$ is the beginning time of the next frame and $q_{i,j}^m$ is the maximum number of packets that can be served on the connection within each frame of size T_m. In the static priority scheduler, the delay bound associated with level m is T_m, $1 \leq m \leq n$. If a connection traverses a level-m RJ$_h$ regulator, it has to be assigned to the priority level m in the scheduler. This introduces the coupling between delay and bandwidth allocation. In contrast, in an RCSP server, a connection can be assigned to any priority level regardless of its rate parameters.

Table 3 summarizes the regulators and schedulers for the four disciplines. Notice that there are two equivalent definitions of eligibility times for each of the DJ$_r$, DJ$_e$ and DJ$_s$ regulators.

F. Delay-Jitter-Control and Rate-Jitter-Control Regulators

As shown in Table 3, the regulators for RCSP/DJ$_r$, jitter-EDD, and stop-and-go are very similar. For each of the three regulators, the eligibility time of a packet at a switch is defined with respect to the eligibility time of the *same* packet at the *previous* switch. Also, the regulators for RCSP/RJ$_r$ and HRR are similar in that the eligibility time of a packet at a switch is defined with respect to *earlier arriving* packets at the *same* switch. In [61], two general classes of regulators called delay-jitter controlling regulators and rate-jitter controlling regulators are defined. Regulators for RCSP/DJ$_r$, jitter-EDD, and stop-and-go fall into the former class, whereas regulators for RCSP/RJ$_r$ and HRR are in the later class.

For a delay-jitter controlling regulator, the eligibility time of a packet is defined with reference to the eligibility time of the same packet at the immediately upstream server. The following definition assumes that the queueing delays of packets on the connection at the immediately upstream server and the link delay from the upstream server to the current server are bounded.

$$e_{1,j}^k = a_{1,j}^k \qquad (12)$$

$$e_{i,j}^k = e_{i-1,j}^k + d_{i-1,j} + \pi_{i,j} + \theta_{i,j}, \quad i > 1 \qquad (13)$$

where $a_{1,j}^k$ is the arrival time of the kth packet at the entrance to the network, and $\theta_{i,j}$ is a constant delay.

While delay-jitter (DJ) regulators maintain all the traffic characteristics by completely reconstructing traffic pattern at output of each regulator, rate-jitter (RJ) regulators only maintain certain characteristics of the traffic. Depending on which traffic models are used by the resource allocation algorithm, different RJ regulators can be defined. As discussed in Section II-B.2 and in [61], each deterministic traffic model, such as $(X\min, X\text{ave}, I, S\max)$ [16], (r, T) [21] (σ, ρ) [8], and D-BIND [35], defines a deterministic traffic constraint function $b(\cdot)$. A monotonic increasing function $b_j(\cdot)$ is called a deterministic traffic constraint function of connection j if during *any* interval of length u, the number of bits arriving on j during the interval is no greater than $b_j(u)$. For each traffic model with a corresponding deterministic traffic constraint function $b(\cdot)$, we can construct a rate-jitter controlling regulator with the following definition of $e_{i,j}^k$

$$e_{i,j}^k = \min\{v : v \geq \max(e_{i,j}^{k-1}, a_{i,j}^k), \\ E_{i,j}(u, v) \leq b_j(v - u) \forall u \leq v\} \qquad (14)$$

where $E_{i,j}(.,.)$, defined below, is the number of bits on connection j that become eligible in interval (u, v) at the ith server

$$E_{i,j}(u, v) = \sum_k (L_j^k | u \leq e_{i,j}^k < v) \qquad (15)$$

and L_j^k is the length of the kth packet on connection j.

Equation (14) is very general and defines a class of rate-jitter controlling policies. Any deterministic traffic model that can be defined with a traffic constraint function has a corresponding rate-jitter controlling regulator. The regulator for HRR is a rate-jitter controlling regulator using the (r, T) traffic model, and the regulator for RCSP/RJ$_r$ is the one using the $(X\min, X\text{ave}, I)$ model. In addition, the implementation of rate-jitter controlling regulators can be very simple. For example, the regulator for the (σ, ρ) traffic model can be implemented by the popular leaky bucket mechanism [54].

G. End-to-End Delay Characteristics and Buffer Space Requirements

The end-to-end delay characteristics and buffer space requirement for nonwork-conserving disciplines are shown in Table 3. In the table, $D(b_j, b^*)$ is the worst-case delay

Table 4 End-to-End Delay, Delay Jitter, and Buffer Space Requirement for Nonwork-Conserving Disciplines

	traffic constraint	end-to-end delay bound	end-to-end delay-jitter bound	buffer space at $h^t h$ switch
Stop-and-Go	(r_j, T_j)	$nT_j + \sum_{i=1}^n \theta_i$	T_j	$r_j(2T_j + \theta_i)$
HRR	(r_j, T_j)	$2nT_j$	$2nT_j$	$2r_jT_j$
Rate-Controlled Servers with $b^*(\cdot)$ RJ regulators	$b_j(\cdot)$	$D(b_j, b^*) + \sum_{i=1}^n d_{i,j}$	$D(b_j, b^*) + \sum_{i=1}^n d_{i,j}$	$\sigma_j + b^*(d_{1,j})$ for 1st switch
				$b^*(d_{i-1,j} + d_{i,j})$ for j^{th} switch $j > 1$
Rate-Controlled Servers with $b^*(\cdot)$ RJ regulator for 1st switch and DJ regulators for other switches	$b_j(\cdot)$	$D(b_j, b^*) + \sum_{i=1}^n d_{i,j}$	$D(b_j, b^*) + d_{n,j}$	$\sigma_j + b^*(d_{1,j})$ for 1st switch
				$b^*(d_{i-1,j} + d_{i,j})$ for j^{th} switch $j > 1$

introduced by a RJ regulator with the constraint function $b^*(\cdot)$ for a traffic stream characterized by the constraint function $b_j(\cdot)$.

As shown in the table, the two frame-based disciplines stop-and-go and HRR have similar end-to-end delay bounds and buffer space requirements. The only major difference between them is that stop-and-go provides a tighter jitter bound than HRR. This is because stop-and-go uses delay-jitter control while HRR uses rate-jitter control.

While the end-to-end delay bounds for stop-and-go and HRR are derived by considering each server in isolation, tighter end-to-end delay bounds can be derived for rate-controlled service disciplines by taking into consideration the delay dependencies in successive switches traversed by a connection [19]. The key observation is that, $b^*(\cdot)$, the traffic constraint function used in the regulators, does not have to be the same as $b_j(\cdot)$, the traffic constraint function used to specify the source. By appropriately setting parameters for regulators and local delay bounds at schedulers, rate-controlled service disciplines can provide end-to-end delay bounds at least as tight at those provided by FFQ-based work-conserving service disciplines. To compare with FFQ-based disciplines, assume that the traffic on connection j is characterized by the (σ_j, ρ_j) model. That is

$$b_j(u) = \sigma_j + \rho_j u. \tag{16}$$

We consider two cases. In the first case, only RJ regulators are used. The traffic constraint function for the regulators and the local delay bound for each scheduler are defined as follows

$$b^*(u) = L_{\max} + \rho_j u \tag{17}$$

$$d_{i,j} = \frac{L_{\max}}{\rho_j} + \frac{L_{\max}}{C_i}. \tag{18}$$

In the second case, the first switch still uses the RJ regulator defined above, but all subsequent switches use DJ regulators with $\theta_{i,j} = 0$. Same local delay bounds are assigned to each switch.

It can be shown that the following holds

$$D(b_j, b^*) = \frac{\sigma_j}{\rho_j}. \tag{19}$$

According to Table 4, an end-to-end delay bound of $\frac{\sigma_j + nL_{\max}}{\rho_j} + \sum_{i=1}^n \frac{L_{\max}}{C_i}$ can be provided to the connection in both cases. Compared to Table 2, the above delay bound is identical to that provided by WFQ, WF^{2}Q, and virtual clock servers. The about assignments are just examples to illustrate the flexibility of rate-controlled service disciplines. More elaborate assignments of regulators and local delay bounds can achieve higher network utilization [19]. With rate-controlled service disciplines, since the traffic can be characterized throughout the network, end-to-end delay bounds can be derived for general resource assignments. WFQ, WF^{2}Q, and virtual clock do not have such a property. In fact, it has been shown in [19] that by properly setting parameters for regulators and local delay bounds for schedulers, rate-controlled service disciplines can always outperform FFQ-based disciplines in terms of the number of connections that can be accepted.

Compared to FFQ-based disciplines, rate-controlled service disciplines have the additional advantage of requiring less buffer space inside the network to prevent packet loss. Based on (17)–(19), and Table 4, it can be easily shown that the total amount of buffer space required for connection j in a network of rate-controlled servers is

$$\sigma_j + (2n-1)L_{\max} + \left(2\sum_{i=1}^{n-1} \frac{L_{\max}}{C_i} + \frac{L_{\max}}{C_n}\right)\rho_j \tag{20}$$

which is less than

$$\sigma_j + (4n-2)L_{\max}. \tag{21}$$

Alternatively, based on Table 3, the total amount of buffer space required for connection j in a network of WFQ servers is

$$n\sigma_j + \frac{n(n-1)}{2}L_{\max}. \tag{22}$$

Since σ_j, which is the maximum burst size, is usually much larger than a packet size, the terms with σ_j dominate (21) and (22). While the amount of the buffer space required for a connection increases linearly with the number of hops when WFQ is used, the amount of buffer space is almost independent of the number of hops when rate-controlled service disciplines are used.

Table 5 Delay Bound Tests for FCFS, SP, and EDF Packet Schedulers

Delay Bound Test	Condition
FCFS	$d \geq \sum_{j \in \mathcal{N}} b_j(t) + \max_{k \in \mathcal{N}} s_k$ for all $t \geq 0$.
SP	$(\exists \tau \leq d_p) t + r \geq \sum_{j \in \mathcal{C}_p} b_j(t) + \sum_{q=1}^{p-1} \sum_{j \in \mathcal{C}_q} + B_j(t + \tau) + \max_{r > p} s_r$ for all $p, t \geq 0$.
EDF	$\begin{cases} t \geq \sum_{j \in \mathcal{N}} b_j(t - d_j) & \text{for all } t \geq 0 \\ t \geq \sum_{j \in \mathcal{N}} b_j(t - d_j) + \max_{d_k > t} & \text{for all } d_1 \leq t < d_{\lvert\mathcal{N}\rvert} \end{cases}$

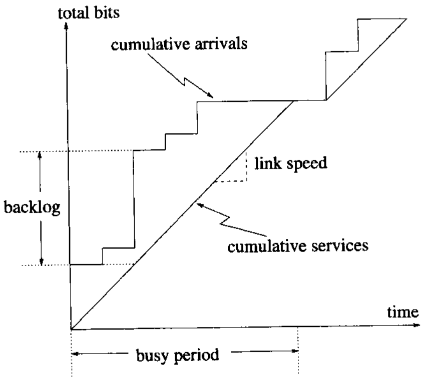


Fig. 17. Concepts: delay, backlog, and busy period.

H. Bounding Delay in a Single Scheduler

In the previous section, we showed that end-to-end delay bounds can be provided in a network of nonwork-conserving servers only when the local delay bound can be provided at the scheduler in each server. Many schedulers such as FCFS, SP, and EDD can be used. Various analysis techniques have been developed to bound the delay in a single scheduler when the input traffic to the scheduler is constrained. In a rate-controlled server, the input traffic to the scheduler is always constrained due to the use of regulators. Therefore, these analysis techniques can be directly applied.

Fig. 17 illustrates the basic concept used in the analysis developed by Cruz [8]. The horizontal axis is time and the vertical axis is bits. The upper curve represents the total number of bits that have arrived in the scheduler by time t and the lower curve represents the total number of bits transmitted by time t. The difference between the two curves is the number of bits currently in the queue, or the *backlog* function. When the backlog function returns to zero (the two curves meet) there are no bits in the queue and thus a busy period has ended. The key to this analysis is that if the upper curve is a deterministic bounding curve, then the maximum delay can be expressed as a function of the two curves. For example, the following two observations hold: the maximum busy period provides an upper bound on delay for any work-conserving server; the maximum backlog divided by the link speed provides an upper bound on delay for a FCFS server. Delay bounds for other policies can also be expressed [1], [8], [40], [48].

Table 5 shows delay bound tests for FCFS, SP, and EDD schedulers as derived in [40]. Notice that while a FCFS scheduler only provides one delay bound and an SP scheduler provides a fixed number of delay bounds, an EDD scheduler can provide a continuous spectrum of delay bounds. In an integrated services networks where applications have diverse traffic characteristics and performance requirements, the flexibility of allocating delay bounds affects the utilization that can be achieved by guaranteed service traffic. In [34], it is shown that SP and EDD schedulers can outperform FCFS scheduler significantly in terms of link utilization when connections have different delay bounds. However, there is little difference in achievable link utilization between SP and EDD schedulers. Since an SP scheduler has only a fixed number of FCFS queues, it is much easier to implement than an EDD scheduler which requires a sorted queue mechanism. Thus, an SP scheduler strikes a good balance between simplicity of implementation and flexibility in allocating delay bounds [62].

I. Implementation Issues

Among the four nonwork-conserving disciplines discussed in this paper, HRR, stop-and-go, and RCSP all use a nonpreemptive Static Priority scheduler. Only delay-EDD use an EDD scheduler which requires a sorted priority queue mechanism. The complexity of implementing sorted priority queue has been discussed in Section III-H. Among HRR, stop-and-go, and RCSP, the former two disciplines implement the rate-controller and the scheduler using one framing mechanism while RCSP needs to implement both using separate mechanisms.

To implement stop-and-go, mechanisms are needed at both the link level and at the queue management level. At the link level, a framing structure is needed, and there is a synchronization requirement such that the framing structure is the same at both the sending and the receiving ends of

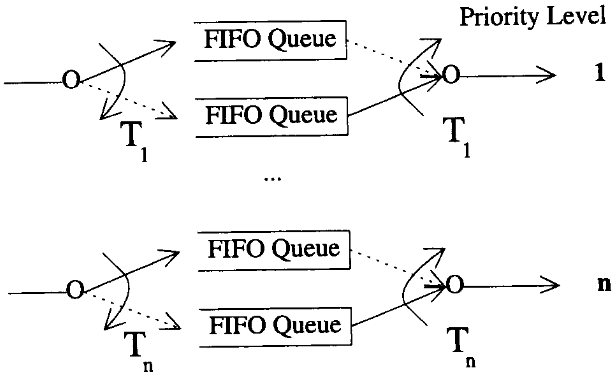


Fig. 18. Implementation of stop-and-go.

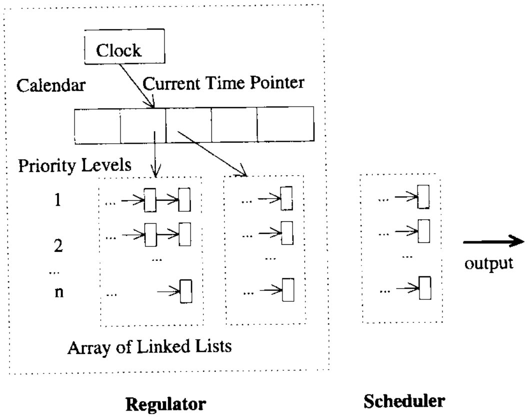


Fig. 19. Implementation of RCSP.

the link. At the queue management level, two FIFO queues are needed for each priority level, one storing the eligible packets ready to be transmitted, the other storing the packets that won't be eligible until the end of the current frame time. Mechanisms are needed to swap the two FIFO queues at the start of each frame time. Also, the set of FIFO queues with eligible packets need to be serviced according to a nonpreemptive static priority policy. This is shown Fig. 18.

HRR does not need the framing structure at the link layer. However, it requires buffering on a per connection basis and a set of timers to perform rate-control. An implementation of a prototype HRR server with 16 priority levels has been reported [27].

RCSP seems to be more complex than stop-and-go and HRR since it requires traffic regulation on a per connection basis. However, the conceptual decomposition of the rate controller into a set of regulators in RCSP does not imply that there must be multiple physical regulators in an implementation; a common mechanism can be shared by all logical regulators. Fig. 19 shows an example implementation of RCSP based on a modified version of a calendar queue [4]. A calendar queue consists of a clock and a calendar, which is a pointer array indexed by time. Each entry in the calendar points to an array of linked lists indexed by priority levels. The clock ticks at fixed time intervals. Upon every tick of the clock, the linked lists in the array indexed by the current time are appended at the end of the scheduler's linked lists. Packets from the linked list of one priority level in the rate-controller are appended to the linked list of the same priority level in the scheduler. The scheduler just selects the first packet at the highest priority queue that is nonempty. As can be seen, the data structures used in the proposed implementation are simple: arrays and linked lists. The operations are all constant-time ones: array indexing, insertion at the tail of a linked list, deletion from the head of a linked list. Another implementation of RCSP that is based on a two-dimensional shifters is proposed in [44].

We would like to point out that a calendar queue is a simpler mechanism that a sorted priority queue. In a calendar queue, only packets pointed by the current time pointer are dequeued at every clock tick. In a sorted priority queue, the next packet needs to be dequeued each time the server finishes service of the current packet. If the sorted queue is implemented by a calendar queue, the dequeueing operation potentially needs to go through *all* the entries in the calendar.

J. Work-Conserving Rate-Controlled Service Disciplines

In previous sections, we showed that nonwork-conserving rate-controlled service disciplines exhibit several interesting properties that make them desirable for supporting guaranteed performance service. These properties include:

1) End-to-end delay analysis can be decomposed into local delay analysis at each switch, and tight end-to-end delay bounds can be derived with such simple analysis for general resource assignments.
2) Heterogeneous servers with different schedulers and regulators can be used at different switches.
3) By separating the rate-control mechanism and the scheduler, the allocation of delay bounds and bandwidth can be decoupled without using the sorted priority queue mechanism.
4) Due to the traffic regulation inside the network, less buffer space is needed at each switch to prevent packet loss.
5) The traffic at the exit of the network satisfies certain desirable properties, for example, bounded rate or delay jitter.

However, nonwork-conserving disciplines also have several disadvantages. First, with nonwork-conserving disciplines, a client is *always* punished when it sends more than specified. Even though this is acceptable under the guaranteed service model, it puts an extra burden on the client to always characterize its traffic correctly. For applications that use live sources such as video conferencing, it is difficult to come up with an accurate traffic characterization *before* the data transmission. If connections are always punished whenever it sends more than specified regardless whether there are spare resources available at that time, they may have to specify the characterization based on an over-estimation of the traffic, which results in a waste of resources. Secondly, while nonwork-conserving disciplines optimize for guaranteed performance service, they may

<table>
<tr><td></td><td colspan="6"><b><i>Delay/Bandwidth Allocation</i></b></td></tr>
<tr><td rowspan="6"><b><i>Interaction of Multiple Servers</i></b></td><td></td><td></td><td colspan="2"><b>Server has one mechanism</b></td><td colspan="2"><b>Server has two mechanisms: Rate-controller and Scheduler</b></td></tr>
<tr><td></td><td></td><td><b>Sorted Priority Queue</b></td><td><b>Multi-level Framing</b></td><td><b>Scheduler using Sorted Queue</b></td><td><b>Scheduler using Static Priority</b></td></tr>
<tr><td rowspan="2"><b>Control Distortion (non work conserving)</b></td><td><b>Delay-jitter Control</b></td><td></td><td>Stop-and-Go</td><td>Jitter-EDD</td><td rowspan="2">RCSP</td></tr>
<tr><td><b>Rate-jitter Control</b></td><td></td><td>HRR</td><td></td></tr>
<tr><td rowspan="2"><b>Accommodate Distortion (work conserving)</b></td><td><b>Index Update Based-on Per Connection Parameter</b></td><td>Delay-EDD
Virtual Clock</td><td rowspan="2"></td><td colspan="2">Rate-controlled servers
With Stand-by Queues</td></tr>
<tr><td><b>Index Update Based-on Reference Queueing Model</b></td><td>WFQ
SCFQ</td><td>WF^{2}Q</td><td></td></tr>
</table>

Fig. 20. Taxonomy of service disciplines.

negatively affect the performance of other packets. For example, with a nonwork-conserving discipline, the server will be idle if there are only guaranteed service packets queued at the server and none of them are eligible for transmission. If some best-effort service packets arrive at the server right after these guaranteed service packets become eligible, the best-effort packets will have to wait before the guaranteed service packets finish service. However, if a work-conserving policy were used, the guaranteed service packets would have been served before the arrival of the best-effort service packets, therefore, the best-effort service packets would not have to wait after they arrive.

A nonwork-conserving rate-controlled server can be easily modified to be work-conserving [10], [19], [62]. In a work-conserving rate-controlled server, there is one more queue in the scheduler, called the *standby queue* [62]. It works as follows:

- All the packets in the rate-controller are also queued in the standby queue. Packets are inserted or deleted from the rate controller and the standby queue simultaneously.
- The scheduler will service the next packet in the standby queue only if there are no nonguaranteed packets and eligible guaranteed packets in the scheduler.

The standby queue allows the noneligible packets to *standby* at the scheduler, so that they can be transmitted when there is spare capacity at the output link.

In [19], it has been shown that the resulted work-conserving rate-controlled server can provide the same end-to-end delay bound as its nonwork-conserving counterpart. Among the five properties listed at the beginning of the section, the first three, and perhaps the more important ones among all, still hold for rate-controlled servers with standby queues.

As a last note in the section, we would like to point out that even without the standby queue, a rate-controlled discipline does not necessarily have to be nonwork-conserving. In [2], it has been shown that the worst-case fair weighted fair queueing (WF^{2}Q) is equivalent to a rate-controlled server with a WFQ scheduler and regulators defined by

$$e_i^k = b_{i,\mathrm{FFQ}}^k \tag{23}$$

where $b_{i,\mathrm{FFQ}}^k$ is the time the packet starts service in the corresponding FFQ system.

In addition, it has been shown that WF^{2}Q is work-conserving. Notice that the regulator defined above is neither a rate-jitter controlling regulator, which is defined by a traffic constraint function, nor a delay-jitter controlling regulator, which is defined by the local delay bound at the previous server. Instead, it is defined with reference to a FFQ system, therefore, the eligibility times of packets are *dependent* on the system load.

V. Summary

In this paper, we have examined a number of packet service disciplines that have been proposed to support guaranteed performance service connections in packet-switching integrated services networks. As shown in Fig. 20, these disciplines can be classified along two dimensions: 1) how the service discipline allocates, explicitly or implicitly, different delay bounds and bandwidths to different connections in a single server; 2) how the service discipline handles traffic distortions in a networking environment.

The first issue relates to the design of a single server. The objective of the allocation of delay bound and bandwidth is that, with a certain discipline, a connection can be guaranteed to receive a certain throughput, and each packet on that connection can be guaranteed to have a bounded delay. In addition to the scheduler, which is responsible for multiplexing packets from different connections and choosing the next packet to transmit, a server can also have a rate-controller. To provide different quality of services to different connections, a server needs to discriminate packets based on their performance requirements. Either a dynamic sorted priority queue or a static priority queue can be used for this purpose. In the case when the server consists of a static priority scheduler and no rate-controller, additional mechanisms are needed to ensure that packets at higher priority levels do not starve packets at lower priority levels. Toward this end, stop-and-go and HRR adopt nonwork-conserving multilevel framing strategies. When compared to the more general rate-controlled service disciplines, multlevel framing suffers from a number of disadvantages.

The second issue concerns the interaction between different servers along the path traversed by the connection. Since the traffic pattern of each connection can be distorted inside the network due to load fluctuations, the server either needs to accommodate the distortion by buffering or control the distortion by regulating the traffic inside the network. Controlling traffic pattern distortion requires nonwork-conserving disciplines, which can be implemented by either using a multilevel framing strategy or decoupling the server into a rate-controller and a scheduler. There are two classes of algorithms to control traffic pattern distortion: delay-jitter control, which maintains the same traffic characteristics at each switch as that at the previous switch, and rate-jitter control, which shapes the traffic according to a prespecified traffic constraint function. All work-conserving disciplines use the sorted priority queue mechanism. This is not coincidental. Only a sorted priority queue has the flexibility to perform both functions of delay bound/bandwidth allocation and adjusting for traffic pattern distortions.

To provide guaranteed performance service, end-to-end delay bounds need to be provided in a networking environment on a per connection basis. Various analysis techniques have been developed. One solution is to analyze the worst-case local delay at each switch independently and bound the end-to-end delay of a connection by using the sum of the local delay bounds at all switches traversed by the connection. Alternatively, it has been observed that smaller end-to-end delay bounds can be obtained by taking into account the delay dependencies among successive switches traversed by the connection. In general, for both types of solutions, the traffic needs to be characterized on a per connection basis at each switch inside the network. For most of the proposed work-conserving disciplines, due to the difficulty of characterizing traffic inside the network, tight end-to-end delay bounds can be derived only for a restricted class of resource assignment strategies called rate-proportional assignments. With rate-proportional assignment, the allocation of delay bounds and bandwidth are coupled. For rate-controlled disciplines, since traffic is regulated inside the network, tight end-to-end delay bounds can be derived for general resource assignments. It has been shown in [19] that by properly setting parameters for regulators and local delay bounds for schedulers, rate-controlled disciplines can always outperform WFQ type of disciplines in terms of the number of connections that can be accepted.

Among the proposed algorithms, rate-controlled service disciplines [19], [64], which separate the server into a rate controller and a scheduler, exhibit the following distinct advantages: 1) simplified stability analysis, which allows tight end-to-end delay bounds to be derived for general resource assignments; 2) decoupling delay bound and bandwidth allocation without using the sorted priority queue; and 3) allowing heterogeneous servers with different schedulers and regulators to be used at different switches. While rate-controlled service disciplines are in general nonwork-conserving, which has the additional advantage of requiring less buffer space within the network to prevent packet loss, they can be easily modified to be work-conserving by introducing a standby queue.

Although we have provided important insights into the issues and tradeoffs of designing service disciplines for integrated services networks, there are several important problems that remain unresolved and need to be addressed in future research. For example, it has been shown that tight end-to-end delay bounds can be derived under general resource assignments for rate-controlled service disciplines but can only be derived under rate-proportional resource assignments for most work-conserving disciplines other than those modified from rate-controlled servers. Future work should develop more advanced techniques to bound end-to-end delay under general resource assignments for FFQ-based work-conserving disciplines Also, how important is it to have general resource assignments? How much higher network utilization can be achieved with general resource assignments compared with rate-proportional resource assignments, and under what traffic mix conditions and network environments? We leave these questions for future research.

As a final note, we would like to point out that the focus the paper is on service disciplines for *guaranteed performance service*. Other services such as the predicted service and various types of best-effort services have different requirements, and there will be different tradeoffs in designing service disciplines for these services. For example, for the same resource assignment, WFQ and WF^{2}Q always provide identical end-to-end delay bounds for all connections. However, as discussed in [2] and Section III-B, the services that they provide or best-effort traffic can be quite different. Issues in designing service disciplines for network services other than the guaranteed performance service are beyond the scope of the paper.

Acknowledgment

The author would like to thank Ed Knightly, Jon C. R. Bennett, and anonymous referees for providing insightful comments that greatly improved the presentation of the paper.

References

[1] A. Banerjea and S. Keshav, "Queueing delays in rate controlled networks," in *Proc. IEEE INFOCOM '93*, pp. 547–556, San Francisco, CA, Apr. 1993.

[2] J. C. R. Bennett and H. Zhang, "WF^{2}Q: Worst-case fair weighted fair queueing, July 95," Submitted to *INFOCOM '96*.

[3] P. Brady, "A techniques for investigating on-off patterns in speech," *Bell Syst. Techn. J.*, vol. 44, pp. 1–22, Jan. 1965.

[4] R. Brown, "Calendar queues: A fast $O(1)$ priority queue implementation for the simulation event set problem," *Commun. ACM*, vol. 31, no. 10, pp. 1220–1227, Oct. 1988.

[5] C. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Trans. Automatic Contr.*, vol. 39, pp. 913–931, May 1994.

[6] H. Chao, "Architecture design for regulating and scheduling user's traffic in ATM networks," in *Proc. ACM SIGCOMM '92*, Baltimore, MD, Aug. 1992, pp. 77–87.

[7] D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: Architecture and mechanism," in *Proc. ACM SIGCOMM '92*, Baltimore, MD, Aug. 1992, pp. 14–26.

[8] R. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, pp. 114–121, Jan. 1991.

[9] R. Cruz, "A calculus for network delay, Part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, pp. 121–141, Jan. 1991.

[10] R. Cruz, "Service burstiness and dynamic burstiness measures: A framework," *J. High Speed Networks*, vol. 1, no. 2, pp. 105–127, 1992.

[11] J. Davin and A. Heybey, "A simulation study of fair queueing and policy enforcement," *Computer Commun. Rev.*, vol. 20, no. 5, pp. 23–29, Oct. 1990.

[12] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *J. Internetworking Res. and Experience*, pp. 3–26, Oct. 1990. Also in *Proc. ACM SIGCOMM '89*, pp. 3–12.

[13] D. Ferrari, "Client requirements for real-time communication services," *IEEE Commun. Magazine*, vol. 28, no. 11, pp. 65–72, Nov. 1990.

[14] ——, "Real-time communication in an internetwork," *J. High Speed Networks*, vol. 1, no. 1, pp. 79–103, 1992.

[15] D. Ferrari, A. Banerjea, and H. Zhang, "Network support for multimedia: A discussion of the Tenet approach," *Computer Networks and ISDN Systems*, vol. 26, no. 10, pp. 1167–1180, July 1994.

[16] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Selected Areas in Commun.*, vol. 8, pp. 368–379, Apr. 1990.

[17] N. Figueira and J. Pasquale, "An upper bound on delay for the virtualclock service discipline,"*IEEE/ACM Trans. Networking*, Dec. 1994.

[18] A. Fraser, "Designing a public data network," *IEEE Commun. Magazine*, vol. 30, pp. 31–35, Oct. 1991.

[19] L. Georgiadis, R. Guérin, and V. Peris, "Efficient network QoS provisioning based on per node traffic shaping," Tech. Rep. RC 20064, IBM T. J. Watson Res. Center, May 1995.

[20] S. Golestani, "Congestion-free transmission of real-time traffic in packet networks," in *Proc. IEEE INFOCOM '90*, San Francisco, CA, June 1990, pp. 527–542, IEEE Computer and Commun. Societies.

[21] ——, "A stop-and-go queueing framework for congestion management," in *Proc. ACM SIGCOMM '90*, Philadelphia, PA, Sept. 1990, pp. 8–18.

[22] ——, "A self-clocked fair queueing scheme for broadband applications," in *Proc. IEEE INFOCOM '94*, Toronto, CA, June 1994, pp. 636–646.

[23] G. Goyal, S. Lam, and H. Vin, "Determining end-to-end delay bounds in heterogeneous networks," in *Proc. 5th Int. Workshop on Network and Operating Syst. Support For Digital Audio and Video*, Durham, NH, Apr. 1995, pp. 287–298.

[24] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM '88*, pp. 314–329, Aug. 1988.

[25] R. Jain, "Congestion control in computer networks: Issues and trends," *IEEE Network Mag.*, pp. 24–30, May 1990.

[26] C. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks," in *IEEE Global Telecommun. Conf.*, San Diego, CA, Dec. 1990, pp. 300.3.1–300.3.9.

[27] C. Kalmanek, S. Morgan, and R. C. Restrick, "A high performance queueing engine for ATM networks," in *Proc. 14th Int. Switching Symp.*, Yokahama, Japan, Oct. 1992.

[28] C. Kalmannek and R. Restrick, "Xunet 2 queue module," *AT&T Bell Labs. Internal Tech. Rep.*, Oct. 1989.

[29] D. Kandlur, K. Shin, and D. Ferrari, "Real-time communication in multi-hop networks," in *Proc. 11th Int. Conf. Distributed Computer Syst.*, May 1991.

[30] M. Karol, M. Hluchyj, and S. Mogan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. 35, pp. 1347–1356, Dec. 1987.

[31] S. Keshav, "A control-theoretic approach to flow control," in *Proc. ACM SIGCOMM '91*, Zurich, Switzerland, Sept. 1991, pp. 3–15.

[32] L. Kleinrock, *Queueing Systems* New York: Wiley, 1975.

[33] ——, *Queueing Systems, Vol. 2: Computer Applications.* New York: Wiley, 1976.

[34] E. Knightly, D. Wrege, J. Liebeherr, and H. Zhang, "Fundamental limits and tradeoffs for providing deterministic guarantees to VBR video traffic," in *Proc. ACM Sigmetrics '95*, Ottawa, CA, May 1995, pp. 275–286.

[35] E. Knightly and H. Zhang, "Traffic characterization and switch utilization using deterministic bounding interval dependent traffic models," in *Proc. IEEE INFOCOM '95*, Boston, MA, Apr. 1995.

[36] D. Knuth, *The Art of Computer Programming. Vol. 3: Sorting and Searching* Reading, MA: Addison-Wesley, 1975.

[37] J. Kurose, "On computing per-session performance bounds in high-speed multi-hop computer networks," In *ACM Sigmetrics '92*, 1992.

[38] ——, "Open issues and challenges in providing quality of service guarantees in high-speed networks," *ACM Computer Commun. Rev.*, vol. 23, pp. 6–15, Jan. 1993.

[39] A. Lazar and C. Pacifici, "Control of resources in broadband networks with quality of service guarantees," *IEEE Commun. Mag.*, pp. 66–73, Oct. 1991.

[40] J. Liebeherr, D. Wrege, and D. Ferrari, "Exact admission control for networks with bounded delay services," Tech. Rep. CS-94-29, Univ. Virginia, Dept. Computer Science, July 1994.

[41] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *J. ACM*, vol. 20, pp. 46–61, Jan. 1973.

[42] S. Low, "Traffic control in ATM networks," Ph.D. dissertation, Univ. Calif. Berkeley, May 1992.

[43] B. Maglaris *et al.*, "Performance models of statistical multiplexing in packet video communications," *IEEE Trans. Commun.*, vol. 36, pp. 834–844, July 1988.

[44] M. Maresca, personal communication, June 1993.

[45] R. Nagarajan, J. Kurose, and D. Towsley, "Local allocation of end-to-end quality-of-service in high-speed networks," in *IFIP TC6 Task Group/WG6.4 Int. Workshop on Performance of Commun. Syst.*, Martinique, Jan. 1993, pp. 99–118.

[46] I. Nikolaidis and I. Akyildiz, "Source characterization and statistical multiplexing in atm networks," Tech. Rep. GIT-CC-92/24, College of Computing, Georgia Inst. Technol., Atlanta, GA, 1992.

[47] A. Parekh, "A generalized processor sharing approach to flow control in integrated services networks," Ph.D. dissertation, MIT, Feb. 1992.

[48] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control—The single node case," in *Proc. INFOCOM '92*, 1992.

[49] ——, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," in *Proc. INFOCOM '93*, San Francisco, CA, Mar. 1993, pp. 521–530.

[50] C. Parris, H. Zhang, and D. Ferrari, "Dynamic management of guaranteed performance multimedia connections," *Multimedia Syst. J.*, vol. 1, pp. 267–283, 1994.

[51] K. Ramakrishnan, D. Chiu, and R. Jain, "Congestion avoidance in computer networks with a connectionless network layer," in *Proc. ACM SIGCOMM '88*, Stanford, CA, Aug. 1988, pp. 303–313.

[52] D. Saha, S. Mukherjee, and S. Tripathi, "Multi-rate traffic shaping and end-to-end performance guarantees in ATM networks," in *Proc. 1994 Int. Conf. on Network Protocols* (*ICNP '94*), Boston, MA, Oct. 1994.

[53] J. Stankovic and K. Ramamritham, *Hard Real-Time Systems.* New York: IEEE Computer Society, 1988.

[54] J. Turner, "New directions in communications(or which way to theinformation age?)," *IEEE Commun. Mag.*, vol. 24, no. 10, Oct. 1986.

[55] D. Verma, "Guaranteed performance communication in high speed network," Ph.D. dissertation, Univ. Calif. Berkeley, Nov. 1991.

[56] D. Verma, H. Zhang, and D. Ferrari, "Guaranteeing delay jitter bounds in packet switching networks," in *Proc. Tricomm '91*, Chapel Hill, NC, Apr. 1991, pp. 35–46.

[57] R. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[58] G. Xie and S. Lam, "Delay guarantee of virtual clock server," Tech. Rep. TR-94-24, Dept. Computer Sci., Univ. Texas at Austin, Oct. 1994. Also in *9th IEEE Workshop on Computer Commun.*.

[59] O. Yaron and M. Sidi, "Calculating performance bounds in communication networks," in *Proc. IEEE INFOCOM '93*, San Francisco, CA, Apr. 1993, pp. 539–546.

[60] ——, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Trans. Network.*, vol. 1, pp. 372–385, June 1993.

[61] H. Zhang, "Providing end-to-end performance guarantees using nonwork-conserving disciplines," *Computer Communications: Special Issue on System Support for Multimedia Computing*.

[62] H. Zhang and D. Ferrari, "Rate-controlled static priority queueing," in *Proc. IEEE INFOCOM '93*, San Francisco, CA, Apr. 1993, pp. 227–236.

[63] ——, "Improving utilization for deterministic service in multimedia communication," In *1994 Int. Conf. on Multimedia Computing and Syst.*, Boston, MA, May 1994, pp. 295–304.

[64] ——, "Rate-controlled service disciplines," *J. High Speed Networks*, vol. 3, no. 4, pp. 389–412, 1994.

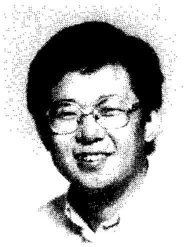
[65] H. Zhang and S. Keshav, "Comparison of rate-based service disciplines," in *Proc. ACM SIGCOMM '91*, Zurich, Switzerland, Sept. 1991, pp. 113–122.

[66] H. Zhang and E. Knightly, "Providing end-to-end statistical performance guarantees with interval dependent stochastic models," in *ACM Sigmetrics '94*, Nashville, TN, May 1994, pp. 211–220.

[67] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switching networks," in *Proc. ACM SIGCOMM '90*, Philadelphia, PA, Sept. 1990, pp. 19–29.

[68] Z. Zhang, D. Towsley, and J. Kurose, "Statistical analysis of generalized processor sharing scheduling discipline," in *Proc. ACM SIGCOMM '94*, London, UK, Aug. 1994.

[69] Q. Zheng and K. Shin, "On the ability of establishing real-time channels in point-to-point packet-switching networks," *IEEE Trans. Commun.*, pp. 1096–1105, Mar. 1994.



Hui Zhang received the B.S. degree in computer science from Beijing University in 1988, the M.S. degree in computer engineering from Rensselaer Polytechnic Institute in 1989, and the Ph.D. degree in computer science from the University of California at Berkeley in 1993.

He is an Assistant Professor of Computer Science at Carnegie Mellon University. His current research interests are in high-speed networks and multimedia systems.

Appendix B: Vendor Paper - IP QoS—A Bold New Network - Nortel Networks

Please find this article on the next page.

NORTEL
NORTHERN TELECOM

Bay Networks

Carrier Packet Networks

IP QoS—A Bold New Network

An IP Quality of Service backgrounder for service providers

White Paper

Abstract

Businesses use the Internet for remote access, information searches, e-mail, and other applications, but do not yet rely on it for all networking needs. Service providers see potential revenue growth in corporate networking services—*if* the security and performance issues of the current Internet can be resolved.

IP quality of service (IP QoS) refers to the performance of IP packet flow through networks. Its purpose is to deliver end-to-end QoS to user traffic. It is is characterized by a small set of metrics, including service availability, delay, delay variation, throughput, and packet loss rate. IP QoS is predicted to lead the way to high-margin business customers, higher-priced service levels, more efficient bandwidth use, and more. It will be a critical enabling technology for the growth of IP networks.

Corporate services are the primary focus of IP QoS, with Service Level Agreements (SLAs) defining the guarantees and responsibilities between subscribers and providers. To forge an agreement that customers can trust, a service provider needs a network with QoS capabilities and a policy management system to configure, control, and maintain performance levels.

Although some work has been done to research, define, and develop IP QoS systems, it is generally agreed that a mature architectural framework, the required supporting hardware, and the appropriate operational techniques are not yet in place.

The evolution of the IP network toward guaranteed QoS promises to be rapid, exciting, and rewarding.

Table of Contents

Executive Summary

From the user side, the Internet has become a powerful consumer and business tool despite its well-publicized shortcomings. Businesses are using the Internet for remote access, information searches, e-mail, and other applications, but do not yet rely on the Internet for all their networking needs.

From the service provider side, corporate networking services constitute a large and profitable revenue opportunity for providers who can solve the security and performance drawbacks of the current Internet.

A NEW LEVEL OF QUALITY

The cornerstone of future IP network growth will be *IP quality of service* (IP QoS). With IP QoS, service providers can achieve greater profitability through high-margin business customers, higher-priced service levels, more efficient bandwidth use, and more.

They can also be more competitive through enhanced service differentiation, better-than-best-effort service, and customized solutions.

IP QOS DEFINED

IP QoS refers to the performance of IP packet flow through one or more networks. The aim is to deliver end-to-end QoS to user traffic. IP QoS is characterized by a small set of metrics, including service availability, delay, delay variation (jitter), throughput, and packet loss rate.

Corporate services are the primary focus of IP QoS, with Service Level Agreements (SLAs) defining the guarantees and responsibilities between subscribers and providers.

ARCHITECTURE

To make a contractual agreement that customers can trust, a service provider needs a network with QoS capabilities and a policy management system to configure, control, and maintain performance levels.

Two IP QoS architectures—Integrated Services Architecture (Int-Serv) and Differentiated Services Framework (Diff-Serv) are currently defined by the Internet Engineering Task Force (IETF). Each has a role and they must be able to interwork.

Int-Serv is implemented at the edge of enterprise networks where user flows can be managed at the desktop user level. More scalable than Int-Serv, Diff-Serv is used in enterprise WANs and plays a key role in the service provider network, based on its ability to prioritize by application or traffic path.

NETWORK SOLUTIONS

In addition to the architectural framework, other elements are required to build real-world IP networks that meet QoS goals.

Routers and switches must meet carrier reliability goals. The network must recover quickly from nodal or link failures. And the QoS mechanisms at each node must be configured to act in concert to deliver end-to-end QoS across the network—a goal that cannot be realistically achieved in any sizable network without a policy manager.

Despite the early stage of development of IP QoS, many components of tomorrow's high-performance, reliable, and flexible IP network have been identified, including:

- Separating traffic according to classification into queues
- A policy manager for managing QoS and SLAs and configuring routers and switches
- Traffic marking and policing mechanisms for entry traffic
- Filtering exit traffic for security and congestion control
- Active output queue management
- Packet discard algorithms
- Monitoring traffic levels at each outgoing interface
- Traffic policies to ensure the safety of premium traffic
- Leveraging of ATM switching and QoS technologies

THE FUTURE

Though IP QoS is in its infancy, it is quite clear that it will be an absolute requirement in commercial IP networks. Its evolution will be rapid, exciting, and rewarding.

Introduction

During the past twenty-five years, the Internet has evolved from a U.S.-government-sponsored research network to today's international, commercially operated network. The first grand-scale application of the Internet Protocol (IP), the Internet is driving the migration of other data traffic from voice, frame relay, asynchronous transfer mode (ATM), and other network architectures to IP networks.

IP technologies are now established as the fundamental platform for the world of webtone and are generally predicted to play a critical—and perhaps dominant—role in the evolution of the public network and private networks such as corporate intranets.

Migrating business network traffic onto public IP networks—including virtual private networks (VPNs)—presents great opportunities for business customers to reduce operating costs, investment risk, and operational complexity.

REMAINING ISSUES

Despite the Internet's rapid growth, implementation issues remain.

For example, the emergence of multimedia traffic over IP networks places great demands on *quality of service* (QoS) in the IP environment. Through the efforts of companies such as Intel and Microsoft, multimedia applications have become an integral part of PC architecture, driving both public and private networks even more rapidly toward a diverse and challenging traffic mix.

Voice and fax over the Internet also provide convincing cost savings and threaten to revolutionize the communications industry. All of these real-time multimedia applications demand better than the current *best-effort* Internet QoS.

The fact is that today's Internet falls far short of delivering the kind of reliability and performance guarantees that enterprises are demanding and are accustomed to in their private networks. Businesses will not place their mission-critical data, voice, and multimedia applications onto public IP networks until they receive secure, predictable, measurable, and guaranteed service.

Furthermore, during the period that the Internet was enjoying such rapid growth, intense competition was pushing margins extremely low in the traditional IP services market.

It is very difficult, if not impossible, to create a successful business model based on a $9.95 per month (with per-hour charges) or a $19.95 per month (unlimited hours) pricing structure. To improve this picture, service providers are now striving to find new sources of revenue and service differentiation that can improve their margins.

QOS OPPORTUNITIES

Moving business traffic—primarily data, but some IP—based voice traffic as well-onto public IP networks is one of the huge opportunities identified by providers in recent years.

A major prerequisite for attracting business customers with this type of mission-critical traffic is to offer alternative IP-based services with guaranteed QoS. By implementing IP QoS solutions, service providers can achieve:

- **Profitability**—improving top-line revenue by attracting high-margin business customers and offering higher-priced levels of services while reducing bottom-line cost by using bandwidth more efficiently.
- **Competitiveness**—enhancing service differentiation by offering multiple classes of *better-than-best-effort* service and by offering customized solutions based on individual requirements.

However, the path to profitability and competitiveness is not straightforward at this time. IP QoS is still a relatively new concept, with vendors offering different proprietary solutions while standards are still being developed.

In this currently uncertain environment, service providers should ask themselves these questions when implementing an IP QoS solution:

- What set of service levels should I offer my customers?
- How can I simplify my IP QoS offerings to communicate easily with my customers?
- How can I offer and cost-effectively manage IP QoS on an end-to-end basis?
- How can I take advantage of my existing IP or ATM infrastructure?

- How can I prepare for future growth and emerging IP QoS standards?
- How can I offer IP QoS in conjunction with Corporate Virtual Private Intranet services?

Service providers who weigh these questions carefully before planning and building IP networks will have a distinct advantage over their competition.

IP QoS Defined

Most industry experts agree that QoS can be a critical differentiator among service providers. However, general agreement on key concepts and terminology relating to service attributes—an important prerequisite for building standardized service offerings—still lags behind.

For example, the term *IP QoS* itself is frequently misused, even by people in the industry. What is advertised as IP QoS is often a set of features for implementing a *class of service* (CoS).

In general communications parlance, *CoS* is a broad term describing a more or less standardized set of features and other characteristics available with a specific service or service package.

QoS is a more precise term, chiefly used to measure a specified set of *performance attributes* typically associated with a service. In the IP network environment, *IP QoS* refers to the performance of IP packets flowing through one or more networks.

Given the current drive toward greater performance and reliability on the Internet, the ultimate aim of service providers is to deliver end-to-end, guaranteed IP QoS to user traffic on IP networks—including data, video, multimedia, and voice.

As a first step toward meeting this goal, a clear definition of QoS, within the context of a definable administrative authority (such as the network defined by a service provider's demarcation points), is a critical prerequisite.

With this aim in mind, QoS can be characterized by a small set of measurable parameters:

- **Service availability**—the reliability of the user's connection to the Internet service.
- **Delay**—also known as *latency*; refers to the interval between transmitting and receiving packets between two reference points.
- **Delay variation**—also called *jitter*, refers to the variation in time duration between all packets in a stream taking the same route.
- **Throughput**—the rate at which packets are transmitted in a network; can be expressed as an average or peak rate.
- **Packet loss rate**—the *maximum* rate at which packets can be discarded during transfer through a network; packet loss typically results from congestion.

With these definitions and parameters in mind, it is now time to look at a key mechanism that can help to ensure QoS in the IP network of the future.

Service Level Agreement

Service Level Agreements (SLAs), although usually thought of in conjunction with VPNs, can apply to all customers of a service provider, including dial-up, corporate, wholesale, or peer network users. An SLA could be a simple standard contract for mass consumers or customized and multidimensional for business customers.

An SLA defines end-to-end service specifications and may consist of the following:

- **Availability**—guaranteed uptime, service latency (where relevant, this is the delay accessing the network)
- **Services offered**—specification of the service levels offered
- **Service guarantees**—for each class; for throughput, loss rate, delay, delay variation, and class over-subscription handling
- **Responsibilities**—consequences for breaking the contract rules; location of the demarcation point; *24 x 7* support and customer service
- **Auditing the service**
- **Pricing**

TABLE 1. QUALITY OF SERVICE PARAMETERS

Service Level	Application	Priority Mapping
1	• Non-critical data • Similar to Internet today (see UBR on ATM) • No minimum information rate guaranteed	• Best-effort delivery • Unmanaged performance
2	• Mission-critical data • VPN outsourcing, e-commerce • Similar to frame relay CIR, ATM VBR	• Low loss rate • Controlled delay and delay variation
3	• Real time applications • Video streaming, voice, videoconferencing	• Low delay and delay variation • Low loss Rate

Central to the service level agreement are the service levels or classes that are available to the user's traffic. *Level of service* (LoS) and CoS are often used interchangeably. Traffic traveling under different service classes receives different levels of quality. An important function of the SLA is, therefore, to assign responsibility for mapping traffic to the different service classes offered.

Developing IP service levels is going to require a phased approach. In the first phases, very simple schemes will be implemented such as the two-bit differentiated services architecture (see reference 1) or the Assured Service (see reference 2), where only two to four service levels are defined. Subsequent phases will be evolutionary based on experience with early deployments and development of the market.

Another factor in favor of simplicity and a limited number of service levels is the user's perception of quality. Even when users can detect variations between the service classes through measurement and monitoring, they have not indicated the willingness to pay an incremental amount for the differences between highly granular performance variations. Early services will likely identify a premium service for mission-critical applications with guaranteed delivery and well-controlled delay, jitter, and throughput.

The next step may be to allow integrated services, with a low-delay, real-time service for voice applications. The natural environment to offer these services is within VPNs for intranet traffic. Table 1 shows an example of a simple set of IP QoS levels and their associated applications.

Note that the example in Table 1 represents current industry thinking about a simple move beyond the best-efforts-only Internet services that users are familiar with today. As the technologies, techniques, and service offerings mature, more sophisticated services will almost certainly be developed and marketed.

A final broad point should also be made about SLA. Because a legal contract is in place between the two parties, each desires to monitor the service performance and usage for different purposes.

The customer monitors to ensure the service provider is meeting the terms of the contract and to track utilization for it's own purposes, one of which may be internal accounting. The service provider monitors to verify any complaints made by the customer and for early detection of any potential violation in order to take preventative measures.

There is also monitoring to ensure that the customer is not over-subscribing services—although this is usually part of traffic conditioning at the trusted boundary point of the service provider's network (discussed later in detail).

IP QoS Architecture

A number of QoS architectures have been defined by various organizations in the communications industries (see reference 3). For IP QoS, the researchers are now focusing on two architectures developed by the

MORE ABOUT INT-SERV

The Integrated Services (*Int-Serv*) model for IP QoS architecture defines three classes of service:

- **Guaranteed**—with bandwidth, bounded delay, and no-loss guarantees.
- **Controlled load**—approximating best-effort service in a lightly loaded network.
- **Best-effort**—similar to what the Internet currently provides under a variety of load conditions, from light to heavy

Using a method similar to ATM's SVCs, Int-Serv uses RSVP between senders and receivers for per-flow signaling. RSVP messages traverse the network to request/reserve resources. Routers along the path—including core routers—must maintain soft states for RSVP flows.

Note: A soft state is a temporary state governed by the periodic expiration of resource reservations, so that no explicit path teardown request is required. Soft states are refreshed by periodic RSVP messages.

Internet Engineering Task Force (IETF)—the Integrated Services architecture (often referred to as *Int-Serv*), and the Differentiated Services architecture (often referred to as *Diff-Serv*).

INT-SERV

Int-Serv was defined in Request for Comments (RFC) 1633, which proposed the Resource Reservation Protocol (RSVP) as a working protocol for signaling in the Int-Serv architecture. This protocol assumes that resources are reserved for every flow requiring QoS at every router hop in the path between receiver and transmitter, using end-to-end signaling.

Scalability is a key architectural concern, since Int-Serv requires end-to-end signaling and must maintain a per-flow soft state at every router along the path. Other concerns are (1) how to authorize and prioritize reservation requests and (2) what happens when signaling is not deployed end-to-end.

It seems likely to current analysts that Int-Serv will be implemented at the edge of enterprise networks where user flows can be managed at the desktop user level. An important driver for Int-Serv in the vicinity of the desktop is Microsoft's implementation of RSVP and QoS capabilities in Windows 98 and NT 5.0.

MORE ABOUT DIFF-SERV

The Differentiated Services (Diff-Serv) model for IP QoS architecture uses a new implementation of the IP Version 4 type of service (ToS) header field. This field can now be marked, so that downstream nodes receive the information required to handle packets arriving at their entry ports and forward them appropriately to the next hop routers. Diff-Serv also renames the eight-bit ToS field as the DS field, with six bits available for current use and two reserved for future use.

Within the six available bits, only one mapping has currently been defined:

- **DE**—(Default), a best-effort class of service.

Another draft is proposing a second code point:

- **EF**—(Expedited Forwarding), not quantitatively defined at present; however, it is described as a forwarding treatment where the departure rate of the traffic from any Diff-Serv node must equal or exceed a configurable rate independent of the intensity of any other traffic attempting to transit the node; there are several implementation schemes that have been proposed but none is standardized yet.

DS Field

DSCP	**CU**

DSCP = Diff-Serv code point (6 bits)   CU = currently unused (2 bits)
DSCP = 000000 indicates DE   DSCP = 101100 indicates EF

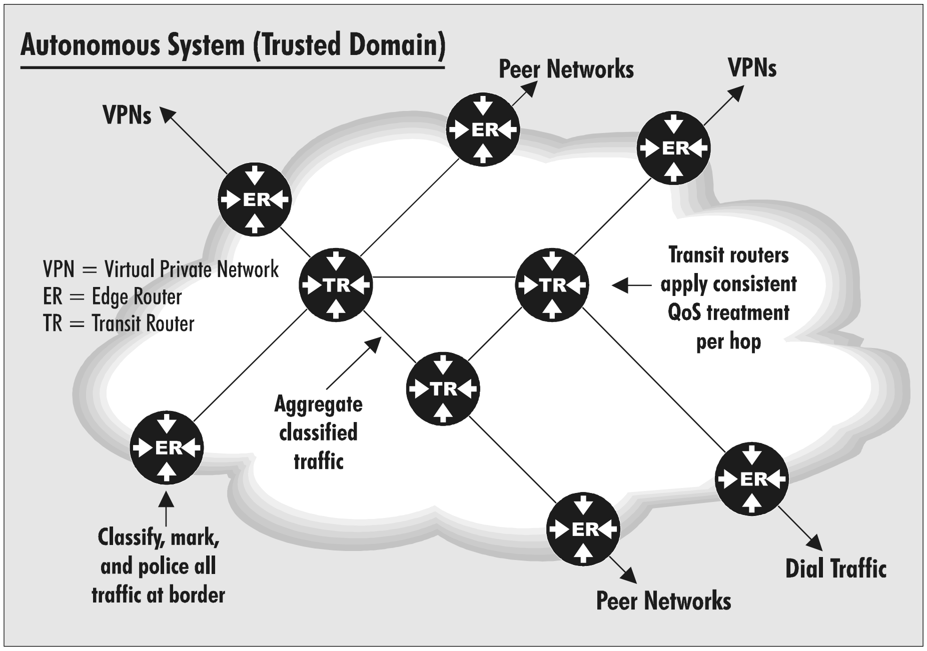

Autonomous System (Trusted Domain)
VPNs
Peer Networks
VPNs
ER
ER
ER
VPN = Virtual Private Network
ER = Edge Router
TR = Transit Router
TR
TR
Transit routers apply consistent QoS treatment per hop
TR
Aggregate classified traffic
ER
ER
ER
Classify, mark, and police all traffic at border
Dial Traffic
Peer Networks

Figure 1. Diff-Serv framework.

DIFF-SERV

Diff-Serv is a relatively new IETF working group that has defined a more scalable way to apply IP QoS. It has particular relevance to the service provider and carrier networks.

Diff-Serv minimizes signaling and concentrates on aggregated flows and per hop behavior applied to a network-wide set of traffic classes. Flows are classified according to predetermined rules, such that many application flows are aggregated to a limited set of class flows.

Traffic entering the network domain at the edge router (ER) is first classified for consistent treatment at each transit router (TR) inside the network (see Figure 1). Treatment will usually be applied by separating the traffic into queues according to the class of traffic.

The eight-bit IP Version 4 type of service (ToS) field is used as a marker to notify downstream routers which treatment to apply to each arriving packet. Diff-Serv has renamed this field the DS (Differentiated Services) field.

Diff-Serv takes control of the ToS field and gives it a simple role in a flexible framework, so that equipment providers can develop configurable QoS capabilities that can interpret bit patterns (code points) in this field as sophisticated per hop behaviors.

Diff-Serv also outlines an initial architectural philosophy intended to provide a framework for inter-provider agreements and make it possible to extend QoS beyond a single network domain (see Figure 2).

The Diff-serv framework is more scalable than Int-Serv because it handles flow aggregates and minimizes signaling, thus avoiding the complexity of per-flow soft state at each node. It will likely be applied most commonly in enterprise backbones and in service provider networks.

However, there will probably be domains where Int-Serv and Diff-Serv co-exist, so there is a need to interwork them at boundaries. This interworking will require a set of rules governing the aggregation of individual flows into class flows suitable for transport through a Diff-Serv domain. Several interworking schemes have been posited (see references 4 and 5).

The responsibility for mapping traffic to classes rests most logically with the customer. However, demarcation points can vary, so in some situations the service provider can manage this role on behalf of the customer. VPN services are particularly affected by such considerations, as will be discussed later in this paper.

REMAINING ISSUES

Diff-Serv lays a valuable foundation for IP QoS, but it cannot provide an end-to-end QoS architecture by itself. Effectively, Diff-Serv markings behave as a lightweight signaling mechanism between domain borders and network nodes, carrying information about each packet's service quality requirements.

Another set of requirements must be addressed before a workable implementation can be built. The principle requirements are:

1. A set of DS field code points in lieu of standards

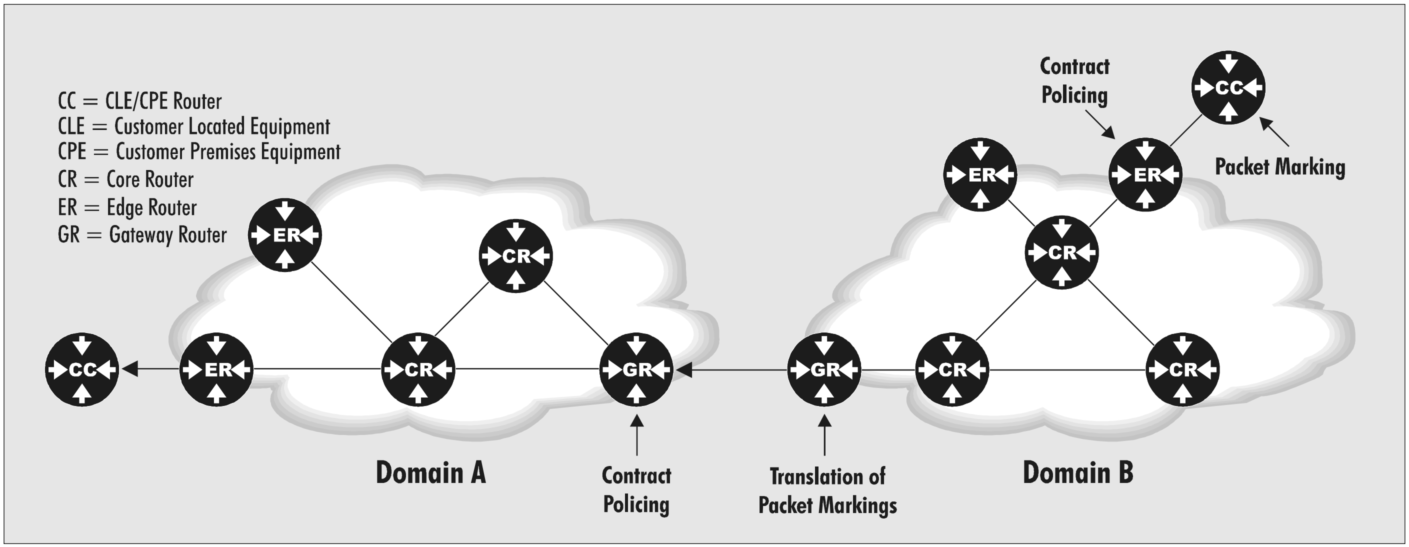


Figure 2. Diff-Serv inter-domain operation.

2. Quantitative descriptions of class performance attributes
3. A mechanism for efficiently aggregating the many sources of premium class traffic that can converge at transit routers
4. A solution to the single-ended SLA problem
5. An interworking solution for mapping IP CoS to ATM QoS
6. Management tools to facilitate deployment and operation

The first two points—standardized DS field code points and quantification of performance attributes—may not be as critical as some of the others in terms of developing standardized implementations. In fact, leaving these two issues unresolved will allow the service provider to develop proprietary solutions and achieve a competitive advantage.

However, lack of resolution in these areas is likely to slow down multi-domain service interworking. Moreover, providers may be able to negotiate agreements and service mappings at borders despite the lack of standardization.

Point 3—aggregation at transit routers—seems much more serious at this juncture of the evolution of IP QoS (see "Traffic management for IP QoS" later in this paper for potential solutions to this problem). It should be noted, however, that aggregation at transit routers is an issue that the communications industries have much to learn about. It will take some experimentation to find which levels of premium traffic can be handled safely. Initially, premium traffic may represent less than five percent of total traffic, but it may increase as confidence rises and new techniques emerge.

Point 4—the single-ended SLA problem—is also a serious challenge. Diff-Serv only manages traffic at the network entry points and does not provide a way to ensure appropriate exit capacity. This is particularly problematic in VPNs, where even high priority traffic might not terminate at a site if the access link is blocked by traffic from other sites. One solution is to over-dimension the access link. Another is to implement filtering (see "Traffic filtering" later in this paper).

Point 5—IP/ATM QoS interworking—is also challenging. Although ATM has excellent and well-defined QoS capabilities, they are path-based. Unfortunately, techniques for mapping IP packets to paths are still at an early stage of development, much like the Int-Serv and Diff-Serv QoS architectures. In addition, ToS-based routing has largely been unimplemented in routing protocols, since the IP ToS field has not been used by applications until recently.

Other ATM solutions are either scale limited—such as Multiprotocol Over ATM (MPOA)—or are proprietary and unlikely to be standardized. A scheme that many industry experts see as more promising in terms of standardization and scalability is Multiprotocol Label Switching

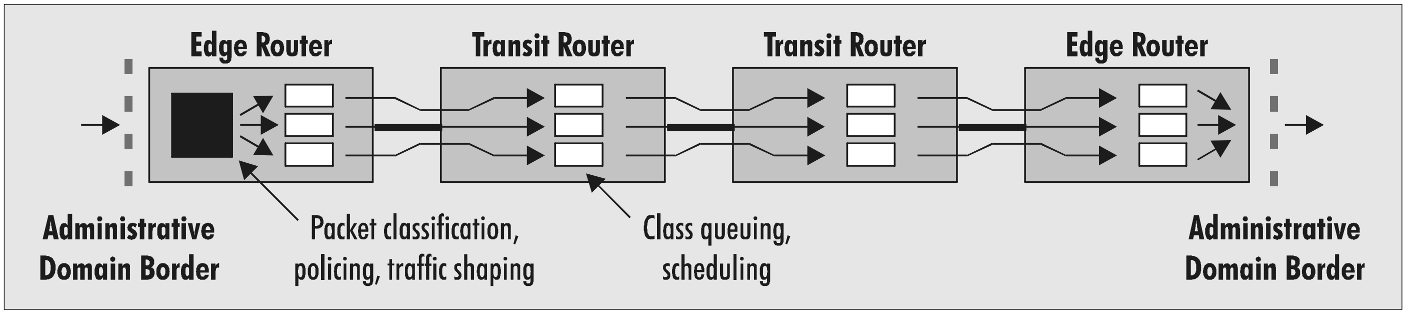

Edge Router
Transit Router
Transit Router
Edge Router
Administrative Domain Border
Packet classification, policing, traffic shaping
Class queuing, scheduling
Administrative Domain Border

Figure 3. Traffic flow across a domain.

(MPLS). See "Leveraging ATM Infrastructure" later in this paper for a discussion of MPLS.

Resolving the final point—the need for management tools—should also prove to be a formidable task. Note that IP QoS is a framework around which service quality can be designed and engineered. It requires a large number of other mechanisms and network elements to operate in harmony before end-to-end service quality can be delivered to users.

Because of the highly distributed nature of these components and the need to manage them centrally, a set of management tools is a critical requirement. The policy manager is the delivery vehicle for this tool set (see "Policy-based management" later in this paper for a discussion of this topic).

NETWORK DELAY

Four different types of delay have been identified in IP networks:

Propagation delay: An inherent delay associated with signals traveling on any physical medium. In the case of fiber optics, propagation delay is somewhat more than the speed of light delay (the theoretical minimum).

Link speed delay: Data transfer rate is determined by the bit rate of the link. A fast link will obviously transfer a packet much faster than a slower link, so the slower link introduces a relative delay. Link speed delay is independent of propagation delay and is by far the greater of the two components.

If traffic is allocated some share of a very fast link (such that its capacity is the same as if it fully occupied the capacity of a slower link), link delay can be reduced—provided that interleaving is at the packet level.

Queuing delay: Every switch and router employs queues, where packets can be stored until capacity is available to transfer them out to the link. Time spent in queues constitutes queuing delay, which accumulates with each device traversed.

Hop Count: Each switch or router traversed by a packet is considered a *hop*. Queuing delay grows as hop count increases, so hop count is an important metric to control.

Implementing IP QoS

Figure 3 shows how traffic flows across an IP network through queues at each node. Queues are provided at each outgoing interface, and, when appropriate, there is a dedicated queue for each traffic class.

The transit routers implement queuing at their output interfaces. Policing is not needed because traffic arrives only from reliable sources.

Based solely on a packet's DS marking, it is inserted into the associated class queue at the appropriate outgoing interface. The traffic in output queues is conditioned by traffic management mechanisms acting on each queue to create a well-defined class behavior. Key functions are allocation of the output bandwidth and establishing rules for how to drop packets when congestion occurs.

Edge routers have the same capabilities as transit routers, but use policing to monitor the customer contract and a classifier to classify and mark the traffic at the incoming interface. The packet arrival rate can be measured for each class to ensure compliance with the SLA. In most cases the average rate over a defined period is checked to minimize the effects of

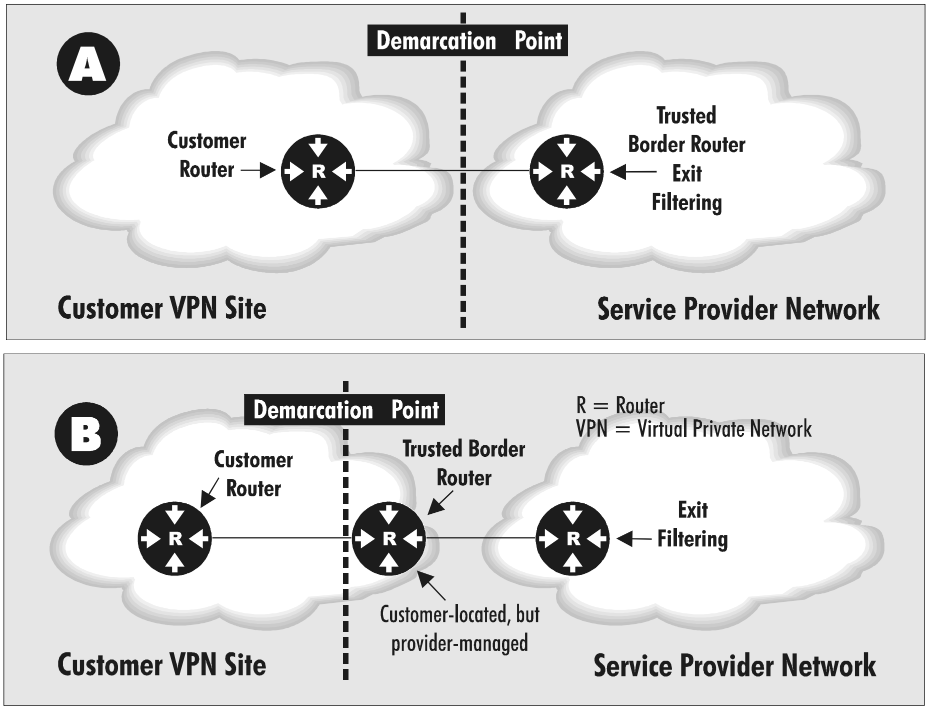


Figure 4. Traffic flow across a domain.

bursty traffic. Traffic can be classified in a number of ways, which are discussed later in this paper.

SLA AND NETWORK DESIGN

Earlier in this paper, the "Service Level Agreement" section discussed the various specifications of an SLA. The following sections discuss two specifications that relate directly to network attributes—availability and service guarantees.

Availability

Availability requires a network robust enough to survive failures such as a fiber cuts, port failures, or switch failures. Today, transport equipment often provides survivability of physical media failure that is almost transparent to higher network services.

Thus, in the case of a fiber failure, IP traffic may be totally unaffected. However, for equal service availability in the case of a non-transport related failure, the network must maintain services—particularly premium services—while minimizing service degradation overall.

One important part of managing service availability is ensuring that the traffic mix is composed of sufficient amounts of drop-tolerant traffic to prevent service degradation from affecting SLA traffic.

Service guarantee factors

The following paragraphs describe the challenges confronting the industry in the evolution toward reliable service guarantees.

Nodal Delay—such as propagation and link speed delay, which are relatively constant, and queuing delay are introduced into the network at each node (see the "Network Delay" sidebar on this page). Network design and planning can control link speed and minimize hop count.

Nodal delay can also be controlled in the queuing stages, where some traffic can be segregated by characteristics scheduling factors into queues, so that a share of the output link is allocated according to traffic engineering rules.

Delay variation—(or *jitter*) can be introduced by path variation, especially when poor network design is a factor.

However, most delay variation results from variations in queuing duration and packets getting stuck behind other long packets. Class-based queuing and output scheduling can be used to reduce jitter for premium types of traffic.

Loss Rate—defines the probability that a packet will be dropped before delivery to the destination. The transient nature of IP traffic patterns makes it difficult to eliminate packet loss.

Over-engineering link capacity is one solution but this may not be cost effective. It is virtually impossible to over-dimension links to the point that no traffic is ever lost. In addition, when failure conditions are taken into account, average utilization would be very low.

A reasonable solution is to implement *some* over-dimensioning and maintain a mixture of high- and low-value traffic, so that low-value traffic is potentially over-subscribed, but insensitive to loss in the event of failure or traffic surges.

QUEUING AND SCHEDULING MECHANISMS

First-In First-Out (FIFO)—the most straightforward approach and very simple to implement. However, with FIFO, a high-priority packet could be stuck behind thousands of best-effort packets.

Strict priority scheduling—where a class is served only if there are no queued packets belonging to any higher-priority classes. This is simple to implement but suffers from the problem that all but one class (highest-priority) could starve.

Fair Queuing or Round Robin (RR)—simple round robin scheduling from multiple queues. This helps in making the bandwidth availability fair to the different queues. One of the problems with fair queuing is that streams with large packets require a bigger share of the available bandwidth.

Weighted-Fair Queuing (WFQ)—an improvement to Fair Queuing. In this scheme, each queue is given a weight that determines the share of that queue to the link bandwidth.

Class-based queuing—uses several queues, each corresponding to a different traffic class (probably as defined by the PHB). Different methods for servicing or scheduling the queues can be used.

Hierarchical Class Based Queuing (CBQ)—Traffic is divided into classes and each class can have sub-classes. This hierarchy forms a tree. If a sub-class exceeds its share of link throughput, it will first try to borrow bandwidth from its sister sub-classes. This tree can be used to distinguish between types of traffic at many hierarchical levels.

Finally, this discussion points to two critical prerequisites for making service guarantees possible:

- *Good network design* is a pre-requisite for QoS delivery.
- *Queuing and scheduling mechanisms* in routers and switches play a vital role, which must be examined.

IP QoS Traffic Management

There are three distinct phases in the flow of every packet through a demarcated network (or *domain*). They are the *entry* phase, the *forwarding* phase, and the *exit* phase.

The network operator, whether a business customer or service provider, must first be concerned with how traffic enters its domain—typically via a trusted border router. This router applies appropriate traffic management processes (or *mechanisms*) to the traffic by agreement between the network operators on each side of the border. The agreed-upon mechanisms that control traffic entry and exit are the basis for the term *trusted border router*.

The network operator must also be concerned with how traffic flows within the domain (for example, is premium traffic handled consistently at each hop?), and with how traffic exits the domain (is exiting traffic marked appropriately for handling in the desired manner after leaving the operators domain of control?).

Thus, at each phase of the journey through a domain, packets may encounter multiple traffic management mechanisms—such as policing, security, filtering, conditioning, or classification mechanisms—that influence the quality of service during the journey.

ENTRY ARCHITECTURE

Upon entering a domain, a packet can be examined in a number of ways, not all of which are necessary for a particular type of traffic. Figure 4 shows two contrasting examples of traffic flowing between a business customer's VPN and a service provider network.

In Figure 4A, the customer owns and administers a WAN access router, typically shaping traffic into the link. Packets are classified by marking the DS field according to agreed-upon policies.

TABLE 2. TRAFFIC CLASSIFICATION

Network Layer	Application	Priority Mapping
4	Port Number	n/a
3	Type of transport protocol	ToS/DS field
2	n/a	Ethernet 802.1p, ATM, frame relay

PACKET DISCARD MECHANISMS

Tail drop—drops arriving packets only when the allocated buffer space is fully occupied. While being easy to implement, it is well known that this approach can lead to network collapse because it triggers the TCP global synchronization.

Random early detection (RED)—very effective at breaking TCP global synchronization. The idea is to try to maintain a small average queue size by randomly dropping arriving packets as the queue occupancy starts building up (but long before real congestion occurs). This causes only a few TCP sources to slow down and reduces the potential for congestion. The probability that an arriving packet will be discarded increases as the average queue size increases. Weighted RED (WRED) is a variant of RED that attempts to influence the selection of packets to be discarded. There are many other variants of RED.

Marking allows the network operators to aggregate individual flows from Int-Serv domains as discussed earlier in this paper. In this case, the trusted border router in the service provider's network *polices* the contract for compliance.

In Figure 4B, the service provider owns and administers the router collocated at the customer's site, so the demarcation and the policing points shift.

In this case, the service provider can shape traffic across the access link according to both the customer's policies and its own. Of course, this type of cooperative arrangement would depend on the level of trust between the parties.

In addition, the connection from the customer's network to the service provider's collocated router can be over Ethernet, instead of a WAN interface, as required in the architecture shown in Figure 4A. This gives the customer the option of classifying traffic using the Ethernet 802.1p priority scheme and letting the service provider map the priority to the packet's DS field according instructions in the SLA.

TRAFFIC FILTERING

Filtering is typically applied to traffic exiting a domain. Exit requirements may simply be filtering for security purposes and to prevent the access link from becoming blocked by low-value traffic.

For example, an exit-filtering policy might be used to dimension traffic termination capacity from other sites so that mission-critical traffic has priority to terminate over low-value traffic. This mitigates some of the problems of single-ended contracts alluded to earlier.

Security filtering might also be needed to prevent unauthorized traffic from entering a private domain. Filtering must be done at the service provider's end of the access link. Otherwise, malicious users could flood the link, causing denial of service for legitimate users. Thus, in Figure 4A and 4B, filtering is implemented on the edge router located at the service provider's premises.

Forwarding behavior in this case is different from classical IP forwarding in the sense that traffic is intentionally treated unequally so that packets marked for better treatment can be isolated and handled consistently at each hop. Forwarding treatment is applied at every stage including entry and exit.

TRAFFIC CLASSIFICATION

A network implementing Diff-Serv defines a standard set of classes throughout the domain. The number of classes may grow over time, but is relatively static and independent of the number of customer SLAs supported.

All traffic inside the network is treated as a standardized set of class flows. Customer service differentia-

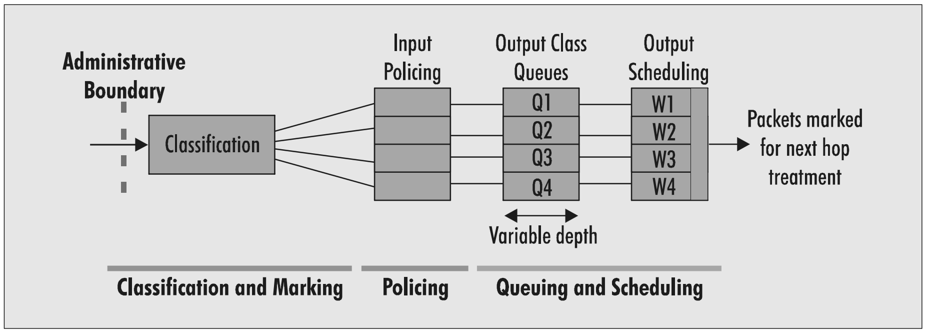


Figure 5. QoS functional model.

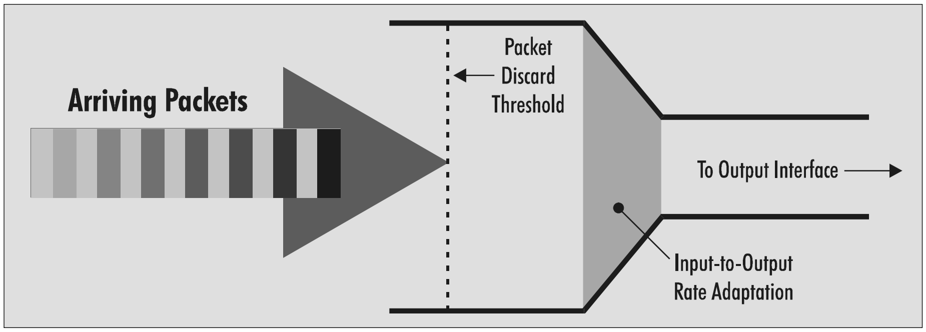


Figure 6. Functional model of output queuing.

tion is achieved entirely through contract negotiation and shaping at the point of entry. Typical customer-specific parameters might be price, penalty clauses, capacity per class, filtering or others.

Traffic entering a Diff-Serv domain must be classified for treatment inside the network. It must either be pre-marked by the customer or marked at the first router on the service provider's side of the demarcation point (see Figure 4).

Customer traffic classified by the service provider's edge router can be based on multiple criteria, ranging from the interworking of various priority schemes to application level analysis of traffic within the IP packet. Table 2 summarizes the options. It should be pointed out that security mechanisms, such as encryption and IPSec, will in some cases prevent application level analysis and classification of the traffic.

TRAFFIC POLICING

Traffic policing is implemented using a classifier (for classifying traffic), a token bucket or similar mechanism (for monitoring entry traffic levels at each class), and markers (for identifying or downgrading non-compliant traffic). Figure 5 shows the QoS functional model, including the policing segment.

Note that downgrading non-compliant traffic on a per-packet basis is not generally considered useful. Diff-Serv deliberately does not look at flows, so downgrading some packets from a premium flow would cause packet reordering—which defeats the purpose of enhanced service quality.

TRAFFIC CONDITIONING

Traffic at output interfaces is first classified and inserted into the correct output queues. Each queue will have selectable drop algorithms such as Random Early Detection (RED) or tail-drop, configurable by the requirements of the class. Each queue will also have programmable schedulers that implement algorithms such as Weighted Fair Queuing (WFQ), Round Robin (RR), and strict priority. These algorithms are also configurable by class requirements.

Figure 6 shows how queues adapt arrival rates to the output interface rate.

In addition, to accommodate different throughput and delay requirements of a class, queue depth is also a configurable parameter. However, there is a tradeoff to be aware of. Short queues can overflow quickly, but offer low delay. Longer queues are better at handling bursty traffic and provide enhanced throughput, but delay is correspondingly worsened. Queue depth must therefore be configured in conjunction with link scheduling and dimensioning in mind, as well as the characteristics of the traffic that will utilize the class.

Network Implementation

Network implementation can be just as complex as issues such as architecture, network design, standardization, and service levels. After all the industry standardization, planning, and development is done, networks must be built in a huge variety of environments, with complex hardware and software configurations, legacy devices, mixed technologies, and many other practical hurdles to overcome. This section provides some practical guidelines for network implementation.

TCP GLOBAL SYNCHRONIZATION

TCP Global Synchronization occurs when a large number of TCP sources lose packets at approximately the same time. This phenomenon leads to cycles of underload (when the involved TCP sources cut their rates simultaneously) and severe congestion (when the involved TCP sources ramp-up their rates simultaneously).

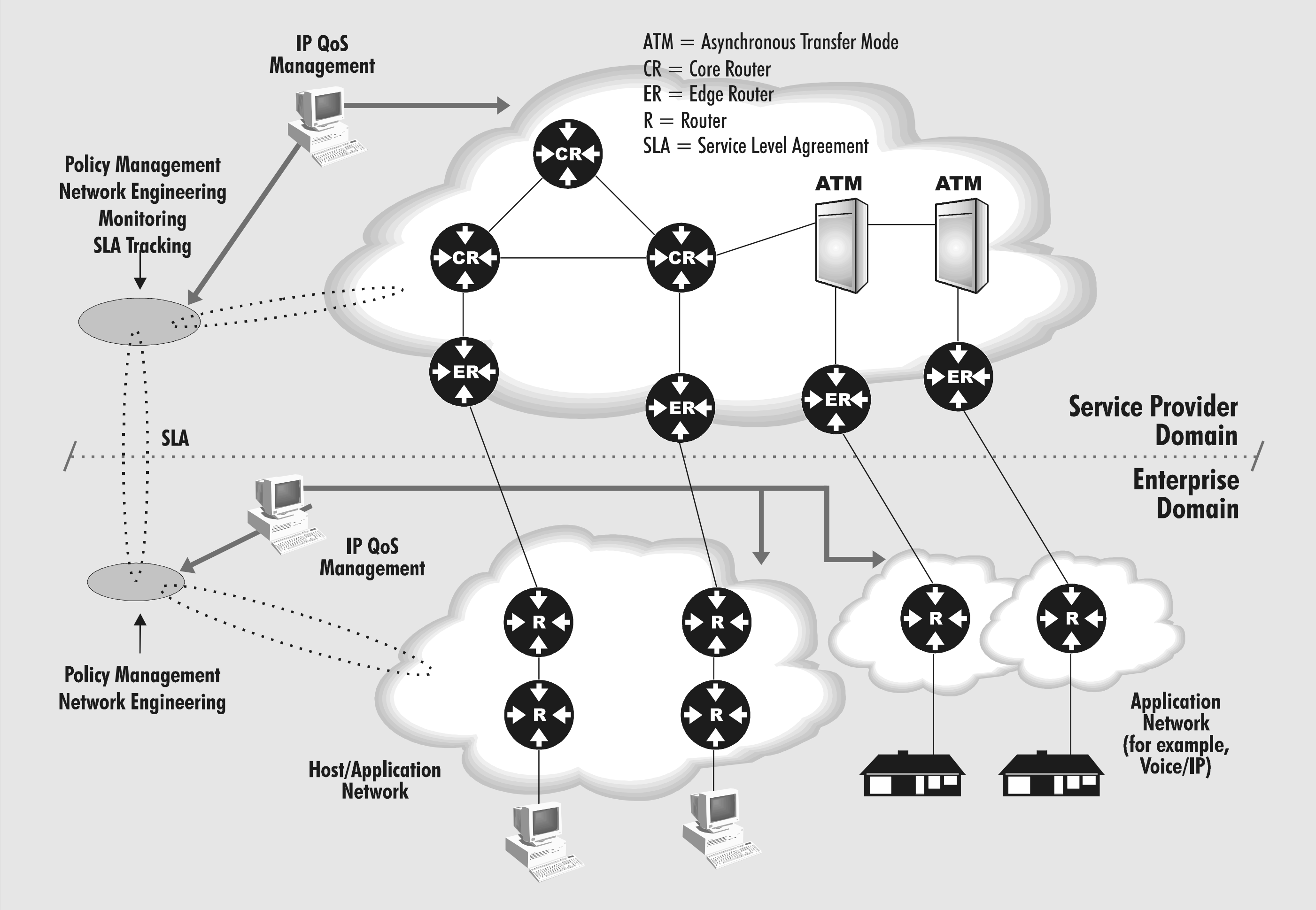


Figure 7. IP QoS architecture.

IP QOS ROUTER CHECKLIST

Router switches that can forward packets and apply traffic conditioning at wire speeds are going to be essential for IP QoS delivery. However, there are other important QoS-related factors to be aware of when selecting router products:

- **Carrier-class fault tolerance and reliability.** True carrier-class reliability will reduce routing instability and both support and improve availability guarantees to customers. The aim is to achieve the so-called *five nines* (99.999%) reliability.
- **Highly flexible QoS mechanisms.** QoS products should offer upwards of four queues (service classes) per interface with configurable discard and scheduling algorithms that can be selected independently for each queue. Look for a choice of mechanisms such as RED, WFQ, and strict priority, so that a rich set of service classes can be constructed.
- **Highly configurable QoS mechanisms.** QoS products should also be able to configure DS field code mappings flexibly to classifications that are user defined. Fixed or limited configuration capability could very quickly prevent service development and differentiation in both the current and future market environments, given the rapidly evolving standards that are predicted. Expect new mechanisms to emerge, such as the ability to create constant bit rate services by metering traffic onto the line.
- **Contract policing.** As service contracts become more complex, they should be rigorously checked for compliance. Token buckets or similar packet-counting mechanisms can be critical IP QoS components, since they allow traffic arrival rate to be verified for each class of service. This information

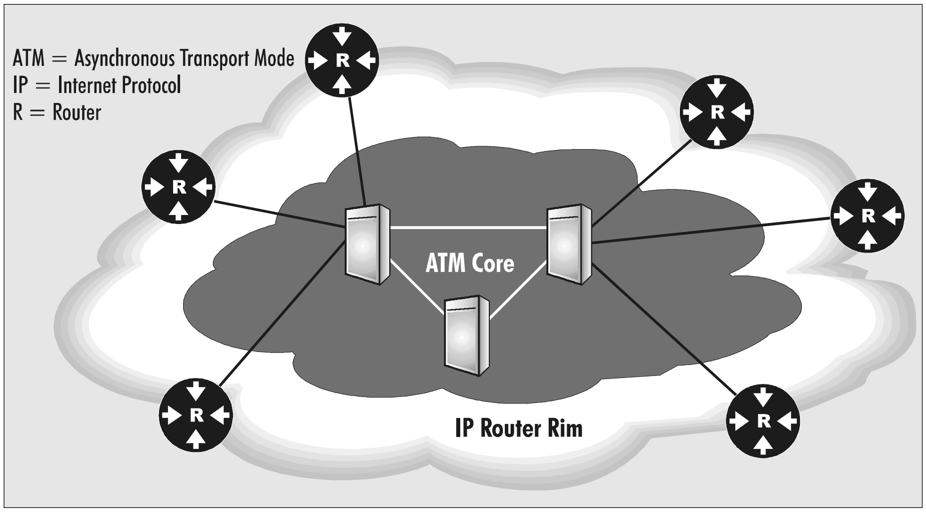


Figure 8. IP traffic channeled to an ATM core network.

can also be invaluable for billing and providing audit trails to customers.

- **Statistics gathering.** QoS products should offer a rich set of counters that can be configured to collect interface statistics on congestion and throughput by class. This information will be vital for traffic engineering and service monitoring.
- **Policy management.** Vendors who offer management tools that allow QoS to be configured and managed in multivendor installations will add value to their products and to customer and provider networks. IP QoS will be difficult to deploy in any reasonable-sized network without these tools.

COPING WITH LEGACY ROUTERS

As QoS services develop, routers will need to be able to process large numbers of packets at full wire-speed, which in the worst cases could be only 40 bytes long. However, legacy routers will be present in some networks, potentially limiting or complicating service offerings. Because of the per-hop behavior model of Diff-Serv, a network-wide set of QoS classes would have to default to the capability set of the lowest performing router.

A possible solution would be to confine legacy routers to best-effort only roles with policy- or QoS-sensitive routing techniques keeping valuable traffic away from them. For example, in a case where ATM is available alongside a legacy IP router network, the identified premium traffic can be groomed onto ATM virtual circuits with appropriate QoS attributes. Another option might be to redeploy the legacy devices to Internet traffic collector roles, feeding to new generation aggregation routers.

Hop-level packet re-marking is another potential limitation of legacy routers. Some routers assign hop behaviors to a non-user-configurable bit pattern in the IP precedence segment of the old ToS field. This would require packet remarking at the entry and exit of legacy environments within a network. Again, the impact could be limited by re-assignment to a best-effort role until scalability considerations allow these routers to be retired from the network economically.

LEVERAGING ATM INFRASTRUCTURE

Before delving into the technical issues of implementation, it is important to briefly consider the roles of ATM switches and IP routers and determine where and when they can be most effectively deployed.

Network solutions for adding QoS to IP traffic vary according to the needs of each service provider. When analyzed in detail, each proposed network has its own complex and subtle requirements, so a generalized approach can fail to find the optimum solution. With this caveat in mind, it is still useful to consider some general criteria involved in the decision process.

MPLS FOR IP AND ATM

Multiprotocol Label Switching (MPLS) labels are assigned at the network's edge router. Information from the routing protocols is used to assign and distribute labels to MPLS peers. In general, an MPLS node receives an outgoing label mapping from the peer that is the next hop for a stream, and allocates and distributes incoming labels to upstream peers for a given stream. The labels are extended into a switched path through the network (in a given service provider's domain) as each MPLS node *splices* the incoming to outgoing labels.

Figure 7 shows an architecture that includes IP routers and ATM switches at the core of an IP network, showing that either technology—and in some cases a mixed solution—is valid.

Considering the following factors can help a network planner decide the right implementation choice:

- Existing infrastructure
- Level of risk involved versus what is considered acceptable
- Time scale for maturity of products
- Amount of IP traffic and growth rate as a percentage of the total traffic mix in the network

As discussed earlier in this paper, there are areas independent of the technology where development and standardization are ongoing. There is, therefore, a choice to be made about how proven the technology is and whether it is standards-based or proprietary.

All of these factors affect risk. For example, choosing a new technique and an unproven product for the same network implementation raises the risk level—but could be acceptable for a new player seeking to steal market share from incumbent providers.

Another important factor is the percentage of IP traffic in the network. If the percentage is low and other types of traffic must be consolidated onto one network, ATM is a solid choice. Some of the more complex decisions arise for IP networks aiming to serve business markets. In this case, there is a particularly delicate trade off to be made between risk levels, time frame for network deployment, and the startup revenue needs of the business case.

Returning to technical consolidations, there are two primary *functional* roles to consider for ATM and IP router technologies—border traffic treatment and class handling inside the network. The model presented here allows Diff-Serv to be implemented over either an ATM- or IP-based core network.

IP can easily make use of the speed and performance of ATM at the core. Variable-length packet data can be adapted to the fixed-length cell transport using ATM adaptation layers (AALs). Both the adaptation to ATM and the switching of cells from one virtual circuit to another commonly take place in hardware. Figure 8 shows a rim of routers channeling IP traffic across ATM output interfaces towards an ATM core transport network.

MPLS can be implemented on ATM switches without modifying the hardware. Supporting MPLS on an ATM switch means that switch operation is controlled by the label switching component by running protocols such as OSPF, BGP, and PIM rather than protocols such as UNI and PNNI. RSVP is one of the methods for allocating QoS resources in IP networks.

More coarse-grained QoS capabilities can be supported by the Label Distribution Protocol (LDP). Such support would be more along the lines of *differentiated services*. The LDP provides the upstream node with Virtual Channel Identifier/Virtual Path Identifier (VCI/VPI) along with the CoS value. The VPI/VCI is used as a label, and QoS is signaled through LDP, based on the previously obtained CoS value in the IP header. The IP QoS Service Level is mapped into ATM as described in Table 1.

Handling of IP and ATM traffic will be based on common traffic management architecture. Some of the issues being investigated include MPLS/-ATM support for loop prevention. The interoperability between MPLS and the overlay ATM subnet require further investigation to eliminate the IP forwarding hop between the network boundary.

Traffic Engineering

For Diff-Serv to function, a traffic policy is required that allows relatively large amounts of traffic tolerant to packet loss to be dropped to ensure the safety of mission-critical and other highly valued traffic.

From the discussion of network issues in the previous section, it can be seen that network design and planning are an essential part of delivering service quality to users. Techniques such as policy or QoS-based routing can have tremendous value in networks with a diverse set of link media (such as wireless and satellite) such that application- and destination-based decisions allow traffic to be routed optimally.

However, path-based decisions have much less relevance to high-scale fiber networks, where delay and bandwidth are much less of a limitation. Path engineering in this type of network is more relevant for route diversity—independent of the routing layer.

Managing Quality of Service

So far, we have considered how SLAs can be implemented from IP QoS structures within a service provider's network—independent of provisioning or maintenance of device configurations. In fact, configuration is not a trivial task, especially when one considers the number of queues that must be configured at each interface and the translation of SLAs into policing contracts at customer interfaces.

Policy management is the solution to this administrative challenge.

POLICY-BASED MANAGEMENT

In fact, policy management, in solving QoS administration issues, enhances the service provider's ability to manage network resources efficiently and offer subscribers new service features. With policy-based management, it is suddenly possible to control bandwidth utilization based on dynamic factors—such as time of day, application priority, and conditions in the network—according to defined policies.

Policies are used to define and dynamically control traffic behavior within a network domain. The alternative to policies is nodal configuration, where intended network-level behavior must be manually translated down to device-level instruction sets.

It may be helpful to think of policies as analogous to high-level programming language statements. Extending this analogy, a device configuration is analogous to a set of machine code instructions. Thus, the relationship of policies to device configurations is high-level to low-level.

In practice, a set of policies effectively creates a device independent program for the network. The program is verified for errors, such as policy conflicts (for example, a local policy might contradict a global policy), and compiled into device specific instructions.

One departure from the programming language analogy is that a program compiler generates machine code for a particular processor, while the policy generator has to create sets of device-level instructions for potentially many different types of network devices.

Different network devices might have equivalent sets of traffic management capabilities but different configuration requirements, a configuration which is reasonably straightforward to manage.

However, complexity arises when the devices have very different capability levels. In some cases, it may be satisfactory to restrict policies to the lowest common set of capabilities. However, in others, some level of manual intervention might be required to address this issue.

In time, these compromises will be eliminated with equipment and network evolution, but for now, they are key issues.

Thus, a policy-based manager (PBM) acts globally across the network domain, supervising device configurations that pertain to traffic management of user SLAs.

The PBM consists of five functions:

- Policy editing
- Policy verification and conflict resolution
- Policy generation
- Policy distribution
- Policy evolution

The policy editor is used by a network administrator to create the network and subscriber policies. Subscriber services (SLAs in particular) need to be interpreted into policy statements, a process that can be performed manually or automated by using service templates from a service management system. The entered policies must be checked for errors and potential conflicts before the device-level instruction sets are created for all the network nodes.

PBMs work with network management to distribute the configurations to the network elements.

Some policies may have dependencies (for example, a dependency on the network state or the time of day might exist), which are sensed by the PBM and result in updates to the device configuration of some nodes. The policy evolution stage looks after these activities.

The PBM system must be robust to failure, so it should use a distributed architecture. Of course, the administrator control console can be centralized to a few locations or even driven from a Web-based terminal that can be accessed from almost anywhere. While security imposes some restrictions and authorization requirements, such an architecture permits a high degree of flexibility.

The distributed architecture also mirrors the nature of global and local policies. *Globally* refers to policies that affect traffic at a network level. *Locally* refers to policies that affect a sub-set of the traffic, as is the case with customer SLAs.

Global policies may pertain to traffic dimensioning rules, nodal QoS requirements for the network service classes, and response actions in the presence of fault conditions. Local policies may include time-of-day and day-of-week dependencies, filtering policies for security, and SLA policies.

MONITORING AND TRACKING

Earlier in this paper, it was mentioned that enterprise subscribers and service providers need to monitor and track service quality to confirm that it is contract-compliant. To facilitate this requirement, the network nodes can collect and store statistics from each node about the traffic flowing through each of the queues.

A large amount of valuable information is thus available from each output and customer interface. Statistics can reflect average and peak throughput and packet discard levels for each traffic class. The statistics can be periodically collected from each node via Simple Network Management Protocol (SNMP) for storage and later processing.

Measuring delay is much more difficult, since it needs to be calculated between end points across the network for a particular packet. It therefore needs to be calculated periodically for each customer's traffic classes with delay variation being discovered over time from the minimum and maximum measurements observed.

Functionality for delay measurements could be integrated into edge nodes or implemented as separate monitoring equipment. The customer could either trust the service provider and request tracking reports prepared by the service provider or implement its own monitoring and tracking solutions at its premises. In the latter case, equipment at customer end points can communicate periodically and sample network performance.

Some of the collected statistics have the more valuable role of charging. Billing might be at least partially flat rate rental and independent of usage. However, SLAs could be written to allow some or all of the service to be usage based. For example, in the case where a fractional service that only partially uses the available capacity is deployed, the contract may allow flat rate up to a certain level but per-packet or per-megabit billing thereafter.

A View into the Future

IP QoS will be the cornerstone of carrier-class IP networking solutions that can be trusted to carry business-critical applications alongside public Internet traffic. Many processes have already been set in motion that will, in turn, trigger other processes and accelerate the evolution toward carrier-class IP networks.

The engine of change is competition for lucrative markets opened by telecom liberalization and the wealth of opportunities afforded by the technology change to connectionless IP networks. The key areas that will feed each other are discussed in the following paragraphs.

IP TRAFFIC PATTERNS

Analysts generally agree that over the next two years IP traffic will grow rapidly and will be the dominant form of traffic in the majority of service provider networks—not just ISPs. The industry structure will change as IP services continue to commercialize and the drive to create profitable businesses intensifies.

Problems associated with *hot potato* routing affect public Internet traffic quality in particular, since the path taken by user traffic is determined by how the user is connected to the service provider, how the service provider is connected to regional, national, and international networks, and the entire network path from user to destination point.

Current commercial pressure seems to be leading to the development of a three-tier hierarchy of providers—from small, local ISPs through larger regional ISPs up to national scale providers. Local ISPs will need to connect to national networks via regional networks.

The rule of markets should eventually limit the players at each level to around three major providers (plus a number of niche providers), simplifying and improving interconnectivity between networks. The result of the simplified industry structure will be that traffic traversing multiple networks will be expedited by far better network performance.

As the industry structure changes, new content and service offerings with local and regional scope will emerge to take advantage of the improved connectivity between users and services.

In addition, new traffic patterns will result, based on communities of interest, so that most traffic will remain local—the reverse of the situation today. Improvements in caching techniques will also help to localize traffic patterns.

DEVELOPMENT OF IP QOS

IP QoS standards will be created both by standards organizations and by the process of *de facto* standards arising and gaining industry-wide adoption. Major areas in need of standardization will be traffic conditioning methodology, CoS definition, policy management protocols, and policy definition language.

Richer sets of QoS traffic conditioners will emerge to become the standard for routing and switching and to facilitate more advanced services and finer control. For example, traffic metering—where packets are transmitted at a fixed rate to break up packet trains and bursts—will help downstream aggregation and lead to more controllable traffic patterns.

Growing numbers of queues will be offered to facilitate finer service granularity. In the future, it could well be viable to define and allocate queues for specific flows.

Policy management algorithms will become fully tuned to network topology and other environmental factors to allow sophisticated high-level policies to be applied to the network, and more effectively govern SLAs, network and traffic engineering, and service restoration. For example, under certain failure conditions, some users may have the option of paying extra to receive the highest priority for early and preferential restoration.

The result of more well-defined traffic patterns and an enhanced ability to control IP traffic is that service quality levels will evolve rapidly and become available to subscribers in increasing numbers and richness.

References

1 "A Two-Bit Differentiated Services Architecture for the Internet," L. Zhang, V. Jacobson, K. Nichols, December, 1997.

2 "IP Precedence in Differentiated Services Using the Assured Service," S. Brim, F. Kastenholz, F. Baker, J. Renwick, T. Li, S. Jagannath, April, 1998.

3 "A Survey of QoS Architectures," C. Aurrecoechea, A. T. Campbell, L. Hauw, Center for Telecommunication Research, Columbia University, New York, NY 10027, USA.

4 "A Framework for Use of RSVP with Diff-serv Networks," Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, K. Nichols, M. Speer, Internet Draft, June, 1998.

5 "A Framework for End-to-End QoS Combining RSVP/Intserv and Differentiated Services," Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, Internet Draft, March 1998.

Glossary

24 x 7	24 hours, 7 days a week
AAL	ATM Adaptation Layer
Administrative Domain	An administrative partition of a network
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CBQ	Class-Based Queuing
CIR	Committed Information Rate
CLE	Customer Located Equipment
CoS	Class of Service
CPE	Customer Premises Equipment
CU	Currently Unused
DE	Default
Diff-Serv	Differentiated Services
Domain	Architectural partition of a network
Downstream	Network element or other network component that follows an upstream element
DS	Differentiated Services
DSCP	Differentiated Services Code Point
EF	Expedited Forwarding
ER	Edge Router
FIFO	First In, First Out
IETF	Internet Engineering Task Force
Gatekeeper	Element that manages addressing, admission, and bandwidth
IETF	Internet Engineering Task Force
Int-Serv	Integrated Services
IP	Internet Protocol
IPSec	Internet Protocol Security protocols
ISP	Internet Service Provider
LDP	Label Distribution Protocol
LoS	Level of Service
LSP	Label-Switched Path
MPLS	Multiprotocol Label Switching
MPOA	Multiprotocol Over ATM
n/a	Not Applicable
OSPF	Open Shortest Path First
PBM	Policy-Based Manager
PC	Personal Computer
PHB	Per-Hop Behavior
PIM	Protocol-Independent Multicast (both Sparse and Dense modes)
PNNI	Private Network-Network Interface
QoS	Quality of Service
RED	Random Early Detection
RFC	Request for Comments
RR	Round Robin
RSVP	Resource Reservation Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SVC	Switched Virtual Circuit
TCP	Transmission Control Protocol
ToS	Type of Service
TR	Transit Router
UBR	Unspecified Bit Rate
UNI	User-to-Network Interface
Upstream	Network element or other network component that precedes a downstream element
VBR	Variable Bit Rate
VCI/VPI	Virtual Channel Identifier/Virtual Path Identifier
VPN	Virtual Private Network
WAN	Wide Area Network
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection

Information subject to change since Nortel reserves the right to make changes, without notice, in equipment design or components as engineering or manufacturing methods may warrant. Product capabilities and availability dates described in this document pertain solely to Nortel's marketing activities in the United States and Canada. Availability in other markets may vary.

For more information, or to order more copies of this document, contact your Nortel sales representative or call 1-800-4 NORTEL (1-800-466-7835) from anywhere in North America. Product and service information is also available on the Internet at Nortel's World Wide Web home page (http://www.nortel.com).

Address correspondence to:

In the United States:

Nortel
P.O. Box 13010
Research Triangle Park, NC 27709
USA

In Canada:

Nortel
8200 Dixie Road, Suite 100
Brampton, Ontario L6T 5P6
Canada

Published by:

Nortel
Marketing Publications, Department 4262
PO Box 13010
Research Triangle Park, NC 27709

74003.25/09-98 Issue 1
Printed in USA September 1998