Traffic Management in Integrated Services Networks

SYNOPSIS

of a thesis to be submitted for the degree of

Doctor of Philosophy

by

Rahul Garg

April 1999

Department of Computer Science and Engineering

Indian Institute of Technology, Delhi

New Delhi-110016, India

1 Introduction

Traditional means of communications employ different networks for different services. For instance, there is a separate telecommunications network for providing voice telephony services. Similarly, there are different networks for television, radio and data transport services. These networks were independently developed and are engineered for the specific service they provide.

With the recent rapid advances in computer and communications technology, different communications services are converging into some form of *Integrated Services* communication. Today, computers can generate and process information in the form of different media like voice, video, text and data and use the same communication networks to transport this information. In such a scenario, different applications can potentially provide different services using the same underlying Integrated Services Networks (ISN).

In addition to emulation of traditional services like voice telephony, video broadcast and data transfer, future Integrated Services Networks are expected to support several novel applications like video conferencing, distributed games, real-time monitoring and control of remote systems etc. Most of these applications need to synchronize their action in real time, and for this they require that each of their sub-tasks complete in a timely fashion. For instance, a video display application displaying 30 frames per second require that the display of a frame does not take more than 33ms in the worst case. Similarly these applications need a performance bound (also called Quality of Service requirement) from the underlying communication network. Such a performance bound is critical for correct functioning of the applications.

Consider a computer telephony application. Voice digitized at one end is packetized and sent to the other end via a packet network. At the receiving end, the application receives the voice packets and reconstructs the original voice. For such an application to work correctly, additional constraints on network performance are needed. If voice is digitized (without any compression) at telephone quality, its transmission would require a dedicated bandwidth of $64Kbps$. Even if the network makes the required bandwidth available to the application, the end-to-end delay of successive packets across the network may be highly variable because of variation in queue lengths at intermediate nodes in the network. If the packet delay is not bounded, then the application would not know how long to wait for a packet before reconstructing the voice. If the network guarantees a bound on the maximum delay incurred by the packets, the application can buffer the packets appropriately before regenerating the voice. However, if the bound on the delay is large, the human users, who are talking using this application would perceive a large delay which may not be acceptable to them. In general a small bound on maximum delay is desirable. Network may drop packets because of various reasons, and it is also desirable to have a bounded packet loss rate.

Providing guaranteed QoS is vital to support various real-time services in Integrated Services Network. The telephony application will not be able to send voice if the desired bandwidth is not available. Distributed real-time games will become highly unpredictable if there is no bound on packet delays. They may even become unfair to players having large packet delays. A lost packet or a delayed packet can play havoc on a real-time control system.

The QoS parameters are defined for a connection (also called flow or session in a connection-less network) between two (or more in case of multicast) communicating applications. Some of the important QoS parameters are:

Bandwidth. It identifies the amount of bandwidth which should be exclusively reserved for the connection. Applications like video conferencing or voice telephony require that a minimum bandwidth be available to them even during peak congestion periods. If this bandwidth is not available, the applications will simply fail to work.

Delay. Packets sent by applications incur a delay before they are received at the other end. This delay has a fixed component and a variable component. The fixed component is because of the signal propagation delay and constant processing delay at intermediate switches and routers. The variable part of the delay is due to the packet queues at various contention points in the network [1]. Most of the applications require the delay to be as small as possible. The *delay* QoS parameter, bounds this packet delay for a connection. Depending upon the definition of QoS parameters, the delay parameter could bound the maximum packet delay of a connection, or the mean packet delay, or a percentile of the delay. A low value of delay results into a faster response time between communicating applications, and results into better interactive performance.

Delay-jitter. It is defined to be the difference between the maximum and the minimum delay packet delay of a connection. For some applications such as video on demand, the delay jitter is more important than the absolute value of the maximum delay. The delay jitter bounds the maximum amount of (playback) buffering needed in these applications.

Loss-rate. Some of the packets sent on a connection may become corrupted because of transmission errors and need to be dropped. Some other packets may also have to be dropped because of buffer overflows caused by transient congestion in the network. The QoS parameter loss rate bounds the loss rate of packets because of these problems.

QoS guarantees can be provided using enhanced traffic management functions. Traffic management is concerned with the problem of managing heterogenous traffic in Integrated Service Networks such that a diverse range of service requests are satisfied as efficiently as possible. In traditional data networks, most of the service requests are homogeneous and congestion is the most severe impediment to good performance. Therefore, most of the traditional research on traffic management has focused on the problem of congestion control. However, in the context of Integrated Services Networks, traffic management has assumed a new significance, as the issues related to QoS need to be solved. In this thesis we examine the traffic management issues for efficient provisioning of QoS in Integrated Services Networks. The key components to provide QoS are traffic characterization, admission control and scheduling.

We first examine bounding traffic models for traffic characterization. We characterize several long traces of compressed video (MPEG, JPEG, NV) using the leaky bucket model [2], X_{min}, X_{avg}, I model [3, 4] and generalized burstiness function [5]. We give qualitative insights for choosing appropriate traffic descriptors. Main results and related work is outlined in Section 2.

We next focus on admission control algorithms for deterministic QoS guarantees. Deterministic guarantees bound the worst case behavior and are therefore expected reserve resources for the extreme case which may be very unlikely. For instance, they may reserve bandwidth for a connection at its peak rate. This may result into lower network utilization. We found the the admission control tests proposed in [3, 4] were suboptimal. We propose improved admission control tests. The admission control tests depend upon the traffic model and scheduling algorithms. A poor traffic model may results into overallocation of resources and low network utilization. We examined the effect of using different bounding traffic models on overall performance. Long traces of compressed video were used for this analysis. We conclude that the leaky bucket traffic model outperforms the X_{min}, X_{avg}, I

traffic model and our admission control strategy can accept more connections while providing deterministic guarantees. Section 3 outlines these results and related work in more detail.

A major contribution of this thesis a new packet scheduling algorithm called the *recursive round robin scheduler* (RRR). We propose two variants of this algorithm, one for scheduling fixed sized cells and another for scheduling variable sized packets. The hardware implementation of the fixed sized cell scheduler can potentially schedule one cell at every clock cycle. Therefore it is particularly suited for implementation in ATM switches. Even the software implementation can potentially operate at high speeds. Delay and fairness properties of this scheduler have been analytically derived for both the variants. These properties are described in more detail in Section 4.

Finally we look at the problem of QoS provisioning in virtual networks. We show that traditional work conserving schedulers cannot provide good end-to-end delay bounds when used in virtual networks. This is because some of the schedulers in a virtual network work on aggregated traffic. We introduce the concept of output burstiness and show that schedulers having low output burstiness in virtual networks have good delay properties. We extend the theory of latency rate servers for virtual networks and present two variants of the RRR scheduling algorithm for virtual networks. Section 5 gives an overview of virtual networks and presents more details of this work.

The output burstiness constraint limits the rate at which traffic may be sent on virtual links of a virtual network. As a result, some packets may have to wait in queues even while the physical link is idle. This reduces the overall throughput of the network. Our suggested solution to this problem is to modify the admission control procedure in the virtual network. We propose a novel technique called stochastic fair sharing (SFS) in which bandwidth reservation of several logical links which are created from the same physical link are adjusted dynamically during session arrivals. This adjustment is done such that the free capacity of lightly loaded logical links is fairly redistributed to heavily loaded logical links, without significantly affecting the session blocking probability of lightly loaded links. With this readjustment, traffic may be sent at a higher rate on virtual links which are heavily loaded, increasing the overall utilization of the network. Section 5.1 gives more details of this work.

2 Traffic Characterization

The network control must know typical behavior of its traffic in order to efficiently manage the traffic and provide the requested QoS. A traffic model summarizes the expected behavior of an application or an aggregate of applications. A single parameter traffic model, like one containing just the bandwidth required by an application is not sufficient. An application may send a burst of data at a very high rate and then become inactive for long time. This makes its traffic bursty. The behavior of traffic flow is characterized more accurately by using a more elaborate traffic model. Some of the proposed traffic models are based on worst case behavior such as burstiness function of Cruz [5], X_{min}, X_{avg}, I model of the Tenet group [3, 4] and leaky bucket model of the ATM Forum [2]. Statistical characterizations are usually based on Markov modulated Poisson process [6] or long range dependent model [7].

Video is a very important component of multimedia applications and the video traffic is expected to take a significant share of real-time traffic on Integrated services networks. There has been a lot of work in statistical characterization of video traffic. A first simplistic model of variable bit-rate (VBR) video traffic appears in [8] where the video traffic is modeled as a first order autoregressive process with marginal Gaussian probability density function and an exponential autocorrelation function. A more sophisticated autoregressive moving average process (ARMA) was used to model the video traffic in [9]. Similarly, there are number of other traffic models for VBR video have been proposed [10, 11, 12, 13, 7, 14]. Most of this work is based on statistical modeling of video traffic and cannot be applied directly to carry out traffic policing. The work on bounding traffic models (the models characterizing the worst case behavior) is limited. Moreover, some of the work used video traces of small duration for characterization and some have only characterized one particular coding algorithm.

In the first chapter of this thesis, a more extensive video traffic characterization has been attempted. Video traffic generated by three different coding algorithms namely MPEG, JPEG and compression algorithm of software NV has been examined. One to two hour long traces of five video sequences of different types, ranging from a lecture in a classroom to a basketball match were considered.

We have characterized the video traffic using a widely known traffic descriptor - Leaky Bucket. Bandwidth and buffer assignment for leaky bucket model has been studied in great detail. We found that leaky bucket is an appropriate descriptor to characterize the traffic. Burstiness of the traffic, has been characterized at various time scale using the burstiness function as defined by Cruz [5].

We found that JPEG compressed video has very little short term burstiness. MPEG and NV trace shows high burstiness over small time scales. JPEG and MPEG video exhibit burstiness over long time scales, whereas NV shows no burstiness over long time scales. It is found that for constant quality JPEG and MPEG compressed video, the leaky bucket parameters depend upon the contents of the video. For JPEG video, the service rate is mainly determined by the peak rate and for MPEG the service rate is given by the peak rate of the smoothed MPEG stream. However, the traffic generated by NV can be characterized independent of the actual video. The target sending rate of the software NV determines the service rate for its traffic.

Characterizing traffic is not an easy task. It is quite intuitive that characteristics depend upon the application, but for same application, say video server the traffic characteristics depend on whether the video is encoded by JPEG, MPEG or another algorithm. Not only this, the characteristics also depend upon the the content of the video. If an MPEG encoded video is full of motion, its bit rate will be high, whereas for a slow and calm video, the bit rate will be low.

3 Admission Control

In Integrated Services Networks, it is necessary to reserve resources before an application begins its data transfer. In a connection oriented network, the resources required to ensure specified QoS for a connection are reserved during the connection establishment phase. Since the traffic descriptor of all connections are known before they are established, the total load on the network is known and the QoS for each connection can be computed.

It may happen that there are not enough resources in the network to provide required QoS to a newly arrived connection request. In this case the connection should not be setup, since the desired QoS cannot be provided, and the requesting application must be informed about inavailability of resources. This is called *admission control*. Admission control enables the network to shed its load by refusing to setup connections, in case of overload and continue to provide the promised QoS to its existing connections. This is similar to what happens in the telecommunication network. No call in a telecommunication network is allowed unless there is a free circuit from the caller to the called party.

There are two components of admission control. Firstly, there must be a network level signalling protocol which checks if adequate resources for the new connection are available throughout its path [15, 16, 17, 18]. The second component is a local test at each node in the connection path. The test determines if the node has adequate resources for the new connection. Given a set of traffic descriptors and QoS requirements of existing connections, the test determines if the new connection with given traffic descriptor and QoS requirement can be accommodated at the node. This test is highly dependent on the switch's internal architecture, its scheduling algorithms and the traffic model.

One of the very common QoS model is the deterministic service model. Deterministic service model and corresponding admission control algorithms have been proposed in [3, 4]. The deterministic service model provides bounds on the worst case behavior. A major criticism of deterministic guarantees is that the network has to reserve the resources for the worst possible case. This translates to peak rate reservation for bandwidth. Thus the network utilization is contemplated to be low. However, this is not true. In [19] it has been shown that for variable bit rate traffic, peak rate reservation is not required.

We show that by choosing a proper traffic model and optimal admission control tests, high network utilizations can be achieved. The earlier admission control tests for deterministic guarantees were suboptimal. We suggest new optimal admission control tests for the earliest deadline first (EDF) scheduling algorithm. Using these tests and video traffic traces we analyzed the performance of deterministic QoS guarantees. The performance of these admission control test using actual real-time data was evaluated. We have shown that the video traffic alone could result into high utilization of network. There is a clear improvement in network performance by using leaky bucket traffic descriptor instead of Tenet group's X_{min}, X_{avg}, I traffic descriptor. This improvement is due to better admission control tests which make use of statistical multiplexing to give better network performance.

4 RRR Scheduler

Different connections may have different QoS requirements. Thus, packets of two connections arriving at a node in the network must be treated differently according to the QoS of corresponding connection. This is done using a scheduling algorithm designed for the purpose. A scheduler decides the order in which packets from different connections are transmitted such that each connection gets the requested

QoS.

Recently, a number of new scheduling algorithms that are aimed to provide per connection QoS guarantees, have been proposed in the context of Integrated Services Networks [20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33].

Many of these scheduling algorithm proposed, were first designed for scheduling packets of variable sizes and were complex [21, 24]. Some of them were later adopted for scheduling ATM cells [30]. However the inherent complexity remains and typical operations needed to schedule a cell are addition, multiplication and division. In addition they also need a multi-level priority queue. Among the schedulers optimized for scheduling ATM cells, some have poor delay and fairness properties [34], some are not scalable for fine rate granularity [33] and some may need to over-allocate rate resulting into poor utilization of link bandwidth [31].

In this thesis we propose and describe a new scheduling algorithm called *recursive round robin scheduler* (RRR), which is optimized for scheduling fixed size packets. Most of the properties of the scheduler depend upon the count c of number of one in the binary representation of normalized rate. For a compliant stream of ATM cells of rate r, and bucket size σ, the scheduler provides a delay and jitter bound of $\frac{1}{r}(\sigma + c)$. We show that work conserving version of the RRR algorithm is fair. The residual link capacity is evenly divided among active streams. We analytically derive fairness bounds for the scheduler. The algorithm has good scaling properties. We also generalize the RRR scheduling algorithm for scheduling variable sized packets. This variant has a little worse delay and fairness properties as compared to the fixed size packet scheduler. Important concepts such as link sharing [35], class based queuing etc. can be implemented using the RRR scheduler.

5 Virtual Networks

Virtual networking is an important step in the evolution of data networks. It allows quick deployment of new services over legacy networks, eases network operation and management by hiding the unnecessary details and presenting a simplified topology, allows development of experimental protocols in a controlled and safe environment and eases interoperability between networks of different types.

A virtual network has virtual nodes overlaid on physical nodes. Pairs of virtual nodes are connected to each other by virtual links, which are realized by tunnels traversing a path between them. Packets arriving on a virtual node are forwarded to the next virtual node via a virtual link based on the routing table of the virtual network. Before forwarding, packets of virtual network are encapsulated [36] into packets of the physical network. These packets carry the required header to route them in the physical network. At the destination node, the encapsulated packet of the virtual network is reconstructed and given to the virtual node.

Some examples of virtual networks are ATM network carrying virtual connections over virtual paths [37] (for simpler network operation and management), IP over ATM networks (for interoperability), virtual private networks in the Internet [38] (for enhanced security), and the MBONE and 6-bone virtual networks [39, 40] over the Internet (for deploying multicast services over the legacy network).

Providing QoS guarantees to real-time applications in virtual networks is as important as in physical networks. For example, future corporate virtual private networks (VPN) will carry real time voice and video conference data along with the regular application data. The research related to scheduling algorithms for virtual networks has been limited. In [35, 26, 25] the concept of link sharing and packet scheduling algorithms to implement link sharing have been proposed. These algorithms hierarchically partition the link bandwidth into several classes while providing bounded delay and rate guarantees to traffic from each subclass. Application of link sharing to build QoS capable virtual networks, where a virtual link is realized by a sequence of logical links, has not been discussed.

We show that traditional scheduling algorithms are inappropriate for providing bounded delay service in a virtual network. The problem arises because traffic of a number of sessions sharing a virtual link in a virtual network is aggregated and tunneled through the physical network, which cannot isolate the traffic of well behaved sessions from that of misbehaving sessions.

We define the concept of output burstiness and show that this problem can be solved by regulating output burstiness on virtual links. Let r be the rate of a virtual link and let $W(t_1, t_2)$ be the amount of the traffic sent by a scheduler on the virtual link in the interval $(t_1, t_2]$. The output burstiness of the scheduler is defined as $b = \max_{t_1,t_2} W(t_1, t_2) - r(t_2 - t_1)$. The output burstiness is a measure of the maximum excess traffic sent on a virtual link than that allowed by the virtual link rate.

Using the theory of latency rate servers [41], we show that latency rate servers with bounded output burstiness, may be used in a virtual network to provide bounded delay service. We show that a network of two latency rate servers, where the downstream server acts on aggregate traffic is also a latency rate server, if the output burstiness of the first scheduler is bounded. The latency of the equivalent server is sum of the latencies of the two servers and b/r where b is output burstiness and r is the rate of the virtual link. This gives a method to design a generic class of scheduling algorithms for virtual networks. We discuss how variants of RRR scheduling algorithm may be used in virtual network.

5.1 Stochastic Fair Sharing Approach

In order to keep low output burstiness, the schedulers in virtual network should not send traffic on a virtual link at a rate larger than the rate of the virtual link. As a result packets may have to wait in the queues corresponding to the virtual link even if the physical link on which they have to be transmitted is idle. In general, the real-time traffic of a virtual link cannot make use of the idle bandwidth of other virtual links, which share the same physical link. This may result into lower network utilization.

We propose a novel technique called stochastic fair sharing (SFS) to increase the bandwidth sharing among different virtual links. Instead of having a fixed bandwidth allocation to a virtual link, the idea is to dynamically adjusts the bandwidth allocation depending upon the current utilization of virtual links. By doing this, the free capacity of lightly loaded virtual link is redistributed to heavily loaded virtual links. This reallocation is done at time scale of session arrivals. Upon arrival of a session, the rate reserved for a virtual link may be changed depending upon the current utilizations of all the virtual links sharing the same physical link. If the entire free capacity of a link is redistributed, then future sessions of the link may get blocked because of bandwidth inavailability. In order to avoid this, some capacity (called trunk reservation) in the virtual links is left unused. Thus, new sessions may reserve bandwidth from this unused capacity. Only small amount of trunk reservation is sufficient to keep the session blocking probabilities to reasonable limits.

Assume a physical link of capacity C is partitioned into N logical links of capacities c_i. Assume that sessions arriving on these logical links require certain bandwidth to be guaranteed to them. If the link was statically partitioned, then upon arrival of a new session, the admission control procedure would simply check if unreserved capacity of the session's logical link is greater than or equal to the bandwidth requirement of the session.

In SFS, the admission control procedure also checks the unreserved capacity of other logical links before making the final decision. Let r_i be the amount of this capacity already reserved by logical link i. The normalized usage of link i is given by $n_i = r_i/c_i$. Assume that the logical links are relabelled in increasing order of their normalized usage. Logical link i has an associated truck reservation capacity t_i. A new session of link i with bandwidth requirement r is accepted iff:

$$\sum_{j=1}^{N} r_j + r + \sum_{j<i} t_j \leq C$$

The amount of trunk reservation needed is fairly small. Our simulations indicate (under the assumption of possion arrivals and exponential holding time) that SFS achieves free capacity redistribution to a reasonable extent without significantly affecting the session blocking probabilities of lightly loaded logical links.

The SFS technique is generic enough to achieve fair sharing of a resource in a *loss network* [42, 43]. SFS is well suited to achieve better link sharing [35] for a single link and can also be applied to improve sharing properties in virtual networks.

6 Summary of Contributions

We characterize burstiness of video traffic at different time scales using the burstiness function. We found that the constant quality video is inherently bursty in the long term because of changes in its scene complexity and the amount of information present in different scenes. The MPEG compression algorithm adds short term burstiness which is specific to the algorithm. Similarly the compression algorithm of the software NV adds the algorithm specific short term burstiness, but removes the long term burstiness by reducing the scene quality of complex scenes.

We compare the X_{min}, X_{avg}, I traffic model with the leaky bucket traffic model and found that the leaky bucket model results into better network utilization. We also discuss some insights gained for choosing leaky bucket traffic descriptor for video traffic. Our studies suggest that the knee point in the leaky bucket traffic descriptor graph of JPEG and MPEG video traffic is a good candidate for its traffic descriptor.

The earlier admission control tests for deterministic guarantees were suboptimal. We suggest new optimal admission control tests for the EDF scheduling algorithm. Using these tests and video traffic traces we analyze the performance of deterministic QoS guarantees. We show that the video traffic alone could result into high utilization of network.

We developed a new scheduling algorithm called the recursive round robin scheduler (RRR) for very high speed networks. Specialized high speed hardware implementation of the RRR scheduling algorithm is possible if the scheduler works on fixed size packets. A general software implementation can be speeded up to take four to six memory accesses to schedule a packet. The delay and fairness properties of the scheduler are analyzed analytically. The scheduler is shown to have all the properties needed to provide QoS in an Integrated Services Network.

Traditional scheduling algorithms do not maintain their end-to-end delay and buffer bound properties when used in a virtual network. This is because some of the schedulers in a virtual network work on aggregated traffic. We introduce the concept of output burstiness and show that schedulers having low output burstiness in virtual networks have good delay properties. We extend the theory of latency rate servers for virtual networks and we present two variants of the RRR scheduling algorithm for virtual networks.

The output burstiness constraint limits the rate at which traffic may be sent on virtual links of a virtual network. As a result, some packets may have to wait in queues even while the physical link is idle. This reduces the overall throughput of the network. Our suggested solution to this problem is to dynamically adjust the rate of logical links depending upon their current usage. The admission control procedure in the virtual network is modified to carry out this adjustment. We propose a novel technique called stochastic fair sharing (SFS) in which bandwidth reservation of several logical links which are created from the same physical link are adjusted dynamically during session arrivals. This adjustment is done such that the free capacity of lightly loaded logical links is fairly redistributed heavily loaded logical links, without significantly affecting the session blocking probability of lightly loaded links. With this readjustment, traffic may be sent at a higher rate on virtual links which are heavily loaded, increasing the overall utilization of the network.

References

[1] A. Banerjea and S. Keshav, "Queueing delays in rate controlled networks," in *Proceedings of INFOCOM*, (San Francisco), Apr. 1993.

[2] "User network interface specifications 3.1." ATM Forum, Prentice Hall, 1993.

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

[4] D. Ferrari, A. Banerjea, and H. Zhang, "Network support for multimedia: a discussion of the Tenet approach," *Computer Networks and ISDN Systems*, vol. 26, pp. 1267–1280, July 1994.

[5] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Transaction of Information Theory*, vol. 37, no. 1, pp. 114–131, 1991.

[6] D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," in *Proceedings of INFOCOM*, 1994.

[7] M. W. Garrett, *Contributions Toward Real-Time Services on Packet Switched Networks*. PhD dissertation, Columbia University, 1993.

[8] B. Maglaris, D. Anastasiou, P. Sen, G. Karlsson, and R. J., "Performance models of statisitcal multiplexing in packet video communications," *IEEE Transactions on Communications*, pp. 834–844, 1988.

[9] R. Gurenefelder, J. P. Cosmas, S. Manthrope, and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE Journal on Selected Areas of Communications*, pp. 284–293, 1991.

[10] P. Skelly, M. Schwartz, and S. Dixit, "A histogram based modeling for video traffic behavior in an ATM multiplexer," *ACM/IEEE Transactions on Networking*, pp. 446–459, 1993.

[11] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 49–59, 1992.

[12] D. S. Lee, B. Melamed, A. Reibman, and B. Sengupta, "Analysis of a video multiplexer using TES as a modeling methodology," in *Proceedings of IEEE Global Telecommunications Conference*, pp. 131–135, 1991.

[13] B. Melamed, D. Raychaudhuri, B. Sengupta, and J. Zdepski, "TES-based traffic modeling for performance evaluation of integrated networks," in *Proceedings of INFOCOM*, pp. 75–84, 1992.

[14] P. Pancha and M. E. Zarki, "A study of MPEG coding for VBR video transmission," *IEEE Communication Magazine*, 1994.

[15] "Private network to network interface specifications 1.0." ATM Forum, Prentice Hall.

[16] I. Cidon, T. Hsiao, P. Jujjavarapu, A. Khamisy, A. Parekh, R. Rom, and M. Siddi, "The OpeNet architecture," Tech. Rep. SMLI-TR-95-37, Sun Microsystems Laboratories, Mountain View, CA, USA, December 1995. http://www.sunlabs.com/technical-reports/1995/smli_tr-95-37.ps.

[17] A. Banerjea and B. Mah, "The real-time channel administration protocol," in *Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video* (R. Herrtwich, ed.), (Heidelberg, Germany), pp. 160–170, Springer-Verlag, Nov. 1991.

[18] L. Zhang, S. Deering, D. Estrin, S. Shenker, and et. al., "RSVP: A new resource reservation protocol," *IEEE Network*, vol. 7, pp. 8–18, September 1993.

[19] D. Ferrari and H. Zhang, "Improving utilization for deterministic services in multimedia communication," in *IEEE International Conference on Multimedia Computing and Systems*, (Boston, Massachusetts), pp. 295–305, ACM, Prentice Hall, May 1994.

[20] L. Georgiadis, R. Guerin, and A. Parekh, "Optimal multiplexing on a single link: Delay and buffer requirements," in *Proceedings of INFOCOM*, (Toronto, Canada), pp. 524–532, June 1994.

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

[22] K. Lindberger and S. E. Tidblom, "Weighted fair queueing, a method to control integrated heterogeneous traffic streams, having different QoS demands," in *Twelfth Nordic Teletraffic Seminar NTS12 (VTT Symposium 154)*, (Finland Technical Research Centre, Finland), pp. 47–58, Finland Technical Research Centre, August 1995.

[23] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control - the single node case," in *Proceedings of INFOCOM*, pp. 915–924, IEEE, May 1992.

[24] L. Zhang, "Virtual clock : A new traffic control algorithm for packet switching networks," *ACM Transactions on Computer Systems*, vol. 9, pp. 101–124, May 1991.

[25] J. C. R. Bennett and H. Zhang, "WF2Q: worst-case fair weighted fair queueing," in *Proceedings of INFOCOM*, (San Fransisco, California), pp. 120–128, Mar. 1996.

[26] H. Zhang and J. C. R. Bennett, "Hierarchical packet fair queueing algorithms," in *Proceedings of ACM SIGCOMM*, pp. 143–156, ACM, September 1997.

[27] S. J. Golestani, "A self-clocked fair queuing scheme for broadband applications," in *Proceedings of INFOCOM*, pp. 636–646, April 1994.

[28] D. Stiliadis and A. Varma, "Design and analysis of frame-based fair queuing: A new traffic scheduling algorithm for packet switched networks," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, (New York), pp. 104–115, May 1996.

[29] D. Stiliadis and A. Varma, "Rate-proportional servers: a design methodology for fair queueing algorithms," *ACM/IEEE Transactions on Networking*, vol. 6, pp. 164–174, Apr. 1998.

[30] A. Varma and D. Stiliadis, "Hardware implementation of fair queueing algorithms for asynchronous transfer mode networks," *IEEE Communications Magzine*, pp. 54–68, December 1997.

[31] C. R. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks," in *IEEE Global Telecommunications Conference*, (San Diego, California), pp. 300.3.1 – 300.3.9, Dec. 1990.

[32] S. J. Golestani, "A Stop-and-Go queueing framework for congestion management," in *Proceedings of SIGCOMM*, (Philadelphia Pennsylvania), pp. 8–18, Sept. 1990.

[33] S. Panwar, T. Philips, and M. Chen, "Golden ratio scheduling for flow control with low buffer requirements," *IEEE Transactions on Communications*, vol. 40, pp. 765–772, April 1992.

[34] S. S. M. Katevenis and C. Courcoubeti, "Weighted round robin cell multiplexing in a general purpose ATM switch chip," *IEEE Journal on Selected Areas of Communications*, vol. 9, pp. 1265–1279, October 1991.

[35] S. Floyd and V. Jacobson, "Link sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, August 1995.

[36] K. Hamzeh, "Ascend tunnel management protocol - ATMP," Feb. 1997. RFC 2107.

[37] M. C. Chan, H. Hadama, and R. Stadler, "An architecture for broadband virtual networks under customer control," in *Proceedings of NOMS '96 - IEEE Network Operations and Management Symposium*, vol. 1, (Kyoto, Japan), pp. 135–144, Apr. 1996.

[38] D. Shah and H. Holzbaur, "VPNs: security with an uncommon touch," *Data Communications International*, vol. 26, pp. 53–63, Sept. 1997.

[39] H. Eriksson, "MBone: The multicast backbone," *Communications of the ACM*, vol. 37, pp. 54–60, Aug. 1994.

[40] D. Lee, D. Lough, S. Midkiff, I. Davis, N.J., and et. al., "The next generation of the Internet: aspects of the Internet protocol version 6," *IEEE Network*, vol. 12, pp. 28–33, January - February 1998.

[41] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," in *Proceedings of INFOCOM*, (San Fransisco, California), pp. 111–119, Mar. 1996.

[42] D. Mitra, R. J. Gibbens, and B. D. Huang, "Analysis and optimal design of the Aggregrated-Least-Busy-Alternative routing strategy on symmetric loss networks with trunk reservations," in *Teletraffic and Datatraffic in a Period of Change, Thirteenth International Teletraffic Congress*, pp. 477–482, Elsevier Science Publishers B. V. (North-Holland), 1991. Copenhagen.

[43] R. J. Gibbens and F. P. Kelly, "Network programming methods for loss networks," *IEEE Journal on Selected Areas in Communications, invited paper for special issue on Advances in the Fundamentals of Networking*, vol. 13, no. 7, pp. 1189–1198, 1995.