Performance Measurements of a Simple Hierarchically Coded Image Animation over Various Network Testbeds

Kirk Taylor and George C. Polyzos

ktaylor@cs.ucsd.edu, polyzos@cs.ucsd.edu.

Computer Systems Laboratory
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0114, U.S.A.

December 1993

Sequoia 2000 Technical Report 93/39

UCSD Technical Report CS93-346

Abstract

Hierarchical coding is a promising technique for handling real-time continuous media (i.e., video, audio, etc.), where timely presentation might be more important than totally lossless presentation. Hierarchical coding allows the receiver to have, and if desired present, a progressively better image as data is received and also allows the network to intelligently discard less important packets when congestion is encountered.

We present the results of a simple hierarchically encoded image animation in various environments (hosts and networks). Presented here are results of the quality of the animation when loss was experienced, throughput rates achieved by the animation, near maximum user-to-user throughput rates of the environments tested, and maximum and average packet loss per frame experienced in the experiments. The highest visually-tolerable packet loss per frame was found to be about 2-3% for this animation and these environments. The highest animation rates and maximum throughput rates were measured between two DEC Alpha workstations connected via an FDDI ring. We were able to achieve animation rates of 4.4-5.2 Mbps with 0-1.3% loss, and maximum throughput rates of 33.2-42.1 Mbps with 0-3.4% loss. The maximum throughput rates are much higher because no image processing was done in those experiments, only reception of the image data and recording of statistics. However, in the animation experiment images are processed and displayed which reduces the throughput achieved.

1 Introduction

Images have been transmitted over conventional communication channels for a long time. The only real communications issue is the transmission delay of high resolution images over low capacity channels.[1] For example a 2000 by 2000 full-color (i.e., 24 bits per pixel), uncompressed image requires the transmission of 96 Megabits (Mb). Using a modern, standard, voice-quality telephone line (a Narrowband-ISDN B-channel at 64 Kb/s) it takes 1500 seconds, i.e., 25 minutes! Even a 1.5 Mb/s T1 line induces a 64 second delay. Compression becomes then essential for interactive applications.

When sequences of images, rather than individual images, are transmitted, new problems arise. When there is a specific temporal relationship between images in the sequence, we refer to such sequences as Continuous Media (CM). Examples of CM are video and image animations (however, CM also include non-visual media such as voice, and in general, audio). Traditionally, for transmission of continuous media constant bit rate circuit-switched channels have been used. Therefore, source coding schemes for such media were designed specifically to produce output at a specific Constant Bit Rate (CBR), typically, the rate of the channel to be used, independent of the instantaneous information content of the signal to be transmitted. This coding typically results in variable signal quality, even though when bandwidth is not severely limited, potential temporary quality degradation is usually engineered to be imperceptible.

For transmission over packet switched networks, the problem of continuous media coding changes. There is typically no a priori constraint on the maximum instantaneous bit rate produced by the coding scheme, but instead, an effort is made to produce constant quality signal at a given average bit rate (or level of quality) by using Variable Bit Rate (VBR) coding schemes. For example, video using interframe compression produces much lower information rates when there is little motion or change between frames, but generates very high bit rate peaks when there are scene changes or fast motion. Peak-to-mean bit-rate ratio for VBR codecs is usually high. In [4] the value 4.7 is reported for one particular codec.

VBR coding is used in order to take advantage of the capability of packet switching networks to use statistical multiplexing for efficient utilization of network resources. One can then economize on resources when the information content of the signal is low and expend them when it is high, achieving the best overall (constant) quality at a given cost. A more complete discussion of VBR video coding in a high-speed network environment is provided in [8].

1.1 Hierarchical Coding

Hierarchical coding techniques (also referred to as component, layered, or sub-band coding) split signals into components of varying importance [4, 5]. The aggregation of these components reconstructs the original data, but subsets of them can also provide various degrees of approximation to the original signal. For example, a simple form of hierarchical coding could decompose a video frame into two sub-frames: (1) a low resolution component containing one quarter of the pixels, and (2) a high-resolution component containing the remaining three quarters of the pixels. For a receiver which uses a presentation window of size one quarter the size of the frame generated by the transmitter, receiving the second part of the signal would not be useful, and can actually be counter-productive because it can be a severe strain on local resources. This type of image coding is known as *pyramid decomposition* [6].

Many different proposals have been made for hierarchical coding of video (and continuous media in general). In [5] a general approach to video transmission over packet- switching networks, emphasizing the role of hierarchical coding, is presented. One of the most basic and conceptually simple methods for hierarchical or layered coding is bit-plane separation [5]. This method applies to images with representations that use multiple bits per pixel. It encodes first, separately, subsets of the image that contain the most significant

[1] There are also many compatibility related issues–format, aspect ratios, etc.–which are not addressed here.

bits (MSBs) of each pixel, which have the most important image information. Then, it progresses down through the bit-layers, towards the least significant bits, which provide the high resolution sub-signal. Thus each bit-plane (or each bit-plane subset) can be separately encoded, progressively transmitted, and decoded independently from the others. More elaborate hierarchical coding techniques are discussed in [11].

Interestingly, with bit-plane separation, the most (visually) important components (i.e., those produced by the MSBs), are also the more highly compressible ones. A reasonable strategy then is to transmit the low resolution, high priority components through connections with explicit performance guarantees, and send the other components on connections without explicit guarantees or require less stringent guarantees.

There are, of course, tradeoffs between the benefits derived from the relative independence of the components produced by hierarchical coding and the total volume of data generated by compressing components individually (in order to achieve their independence). Interestingly, many compression standards support various forms of hierarchical coding [11].

Video conferencing for scientific collaboration and animation of scientific visualization sequences, are Sequoia 2000 applications that might benefit from the use of hierarchical coding for communication over packet-switching networks. Image database browsing can also take advantage of the reduced perceived latency of either progressive transmission or transmission of only the most important signal components. Another latency-hiding technique appropriate for image browsing (which can be combined with compression) is the use of abstracts [3]. Abstracts are more flexible than compression and hierarchical coding and can provide arbitrary compression ratios, however, they require application dependent computation to achieve these advantages. On the other hand, hierarchical coding is perception oriented and uniform across applications.

1.2 Implications of Hierarchical Coding for Network Design

Many of the techniques used in conventional packet-switched networks are based on assumptions of relatively slow, essentially non-interactive communication. It is important to re-examine those assumptions and reformulate the problems in the context of real-time, high-volume traffic. For example, one such assumption is the requirement for error-free transmission. The delay constraints of continuous media do not permit retransmissions for error recovery (at least end-to-end for transcontinental links). Therefore, the (open loop, end-to-end) packet error rate of the transport mechanism becomes an issue. Fortunately, however, real-time audio and video usually exhibit some tolerance to transmission errors, depending on the particulars of their encoding scheme and the level of compression. Thus, audio and video may be characterized as *soft real-time* traffic, i.e., they require some kind of statistical delay guarantees, such as percentile of packets (or bits) received within the given delay tolerance [1].

This property has the potential to diminish the effects of communication errors, such as those leading to packet loss due to congestion. This is possible with protocols that support priorities based on the type of information transferred and which use these priorities to control congestion by “dropping” the least important packets, rather than being indiscriminate. Congestion control schemes based on these ideas have been proposed and are now under consideration for B-ISDN using ATM [7]. For example, *graceful* video quality degradation in the presence of congestion is usually more acceptable than audio degradation, and can be facilitated through the use of hierarchical coding.

For the *Sequoia 2000 Network* guaranteed performance protocols [1] being developed at UC Berkeley will provide the necessary functionality. In this case, hierarchical coding should be able to significantly increase the available capacity by relaxing the performance requirements of a significant proportion of the (less important components of the) traffic, while the most important signal components take advantage of the performance guarantees.

A form of packet-switching, called the Asynchronous Transfer Mode (ATM) has been chosen as the universal switching method for the future Broadband Integrated Services Digital Network (B-ISDN), which promises to integrate the traditional telephone network, current data networks, and provide effective and efficient support for the newly arising multimedia applications. Even though there is still debate over the form and the implementation details, statistical multiplexing is a central feature of ATM (and packet-switching more generally), and one of the arguments for its selection over the alternative, Synchronous Transfer Mode (STM), a form of circuit-switching.

Statistical multiplexing is based on the premise that for independent sources the total amount of resources required (or used) to satisfy the traffic demands at any time is considerably less than the sum of the peak demands of the sources. Statistically, this is explained by the law of large numbers. Therefore, it is expected that applications that require high Quality-of-Service (QoS) will be able to reserve resources, not at the level of their peak demands, but at a much lower level, hopefully close to their average demand. This difference has significant economic implications.

Of course, there will be instances where peak demands are presented to the system simultaneously. In most of these cases the system will be unable to immediately satisfy these demands. In traditional packet-switching networks these situations have been handled through queueing of the requests. With real-time services, however, queueing might introduce unacceptable delays. Therefore, one of the techniques considered for traffic smoothing and controlling user perceived latency in times of high congestion in high-speed networks transporting continuous media, is the dropping or delaying of parts of the signal that might be of secondary importance. This approach is made possible by the use of hierarchical coding and appropriate traffic labeling.

1.3 The *Sequoia 2000* Network and CSL

The *Sequoia 2000* Network (S2Knet) is a private, wide area packet switched network providing high throughput for distributed scientific applications, and real-time services for multimedia collaboration tools such as video-conferencing [2].

It consists mainly of a T1 (1.5 Mb/s) backbone connecting the Sequoia 2000 campuses, and 100 Mb/s FDDI local area networks for local distribution at each campus. The bacbone will soon be upgraded to DS3 (45 Mb/s); the first DS3 link is already in operation connecting UC Berkeley to UC San Diego. The Computer Systems Laboratory (CSL) at UCSD, in collaboration with the Tenet Group at UC Berkeley, is responsible for designing the S2Knet [2]. The actual construction of the S2Knet is a large distributed engineering effort involving all the Sequoia 2000 sites; a "handbook" serves as an implementation guide as new sites are added, links are upgraded and is continually updated to reflect the current status of the network [9].

The S2Knet became operational in 1992, and now provides the experimental infrastructure for us to carry out network research (e.g., to evaluate new network protocols), while at the same time providing the project's Earth scientists with a fast production network (during normal working hours) that is rarely down and is well utilized. This network provides a significant improvement in performance over what was previously available through the *Internet* [10].

CSL is equipped with a variety of workstations, including DEC 3000 Model 500 AXP Alpha deskside workstations, DEC 3000 Model 400 AXP Alpha workstations, DECstation 5000/200 workstations, DECstation 5000/240 workstations, and Sun SPARCstation 2 workstations, Networks within the laboratory include Ethernet and FDDI local area networks. CSL is connected to various UCSD campus networks, as well as a private campus-wide FDDI network which is part of the Sequoia 2000 Network. There is also an ATM network consisting of a FORE System's ASX-100 ATM switch connected to DECStation 5000/200 workstations with FORE System's TCA-100 ATM adapters.

2 Experimental Environment

2.1 Animation and Throughput Experiments

Two experiments were performed for each environment tested. The primary experiment was to determine throughput and acceptable loss of the hierarchically coded image animation (referred to as animation). The secondary experiment was to determine maximum achievable throughput for the animation experiment (referred to as throughput). The animation speeds were based on (and should be compared to) the throughput results. In both cases, throughput and loss statistics were collected. The acceptable loss value for the animation experiment was derived by observing the animation as it proceeded.

Both experiments consisted of a constant stream of packetized video data sent from one system and received and processed on the other system. The experiments were virtually identical and differed only in the following manner. In the animation experiment received packets are reconstructed into frames and displayed. In the throughput experiment, packets are counted and discarded, yielding a much higher possible throughput, which is considered the maximum possible for the animation.

Both experiments send groups of packets over the network in a time-paced manner governed by the packet send time, in a manner similar to the leaky bucket algorithm. Packets are sent using a datagram socket, which in turn sends the data as UDP packets. No acknowledgement or flow control is performed in order to maintain a strict open-loop control, send-receive paradigm, as proposed in [13].

More specifically, each video frame is fragmented into 264 packets. The packets are then sent in small groups periodically based on an interval time, t, in milliseconds. The packet group size, g, was generally small, 2 or 3, and an integral divisor of 264. A larger packet group size was needed at times on the DECStation 5000s due to the resolution of the interval timer available. The sender sends g packets and then checks if the interval timer has expired. If it has not, then the sender waits for the timer to expire. If the timer has expired, the timer is reset and the transmission of the next group of packets begins immediately. This guarantees that packets will never be sent faster than g packets per time t. However, it doesn't guarantee that packets are not sent slower than this rate. In the experiments, the group size and interval are adjusted to increase the sending rate (thus decreasing the frame time) to the point that packets start to be lost. This rate is the limit of the network facility, the host network interface, or typically in this experiment, the application speed.

In the results reported in Section 3, the frame time is determined from $264 * g/t$; and the packet time (packet send time) is calculated as g/t. Throughput is calculated from the perspective of the receiver, and is the number of bits of good data divided by the time observed by the receiver for the experiment. Therefore, the throughput figure does not include lost or late packets. Maximum and average packet loss per frame are the actual number of packets that were observed lost or late by the receiver.

2.2 Hierarchical Coding Technique

The hierarchical coding technique used in this experiment is known as bit-plane separation. Ideally, the packets in the video stream would be assigned priorities in the following manner. The frame is divided into y groups of bit-planes, each group receives a priority from 0...y, 0 being the highest priority. Each packet in the x^{th} group (x: 0...y) receives priority x. Bit-planes are always sent in ascending order from group 0...y (in other words from the most significant bit-plane to least significant bit-plane). In this experiment, no explicit priorities are assigned, since the underlying networks do not support priorities. However, an implicit priority is used due to the send order. Bit-planes are sent in order from most significant to least significant, therefore, more significant bit-planes are more likely to arrive on time than less significant bit-planes. (This didn't always prove to be a benefit as will be discussed in the results).

The experimental video animation is a sequence of 550×550 8-bit color images produced from satellite data. This sequence was generated by members of Catherine Gautier's lab at UCSB. This sequence and other images are available on `heel.s2k.berkeley.edu` in the showcase directory (`/data/10/showcase`). This directory contains sample images contributed by Sequoia 2000 researchers.

Each frame is approximately 300 kilobytes of data. Each frame is fragmented into 264 packets each containing 1224 bytes of video data and application layer header; 32 bytes of UDP/IP header is also added. Images are sent uncompressed, although, it has been noted that this form of coding could yield very high compression rates, particularly in the higher priority bitplanes. Experiments consisted of sending 100 or 200 frames, which is 32 or 64 megabytes of data, respectively.

2.3 Systems and Networks

The systems and networks used were two DECStation 5000/200 systems communicating over Ethernet and ATM networks; a DECStation 5000/240 and a DECStation 5000/200 communicating on an FDDI network; and two DECStation 5000/240 systems communicating on the Sequoia 2000 Network (between UC Berkeley and CSL-UC San Diego); lastly, two DEC 3000/500 Alpha AXP systems communicating on Ethernet and FDDI. The DECStation 5000s in CSL are running Ultrix version 4.2A. The DECStation 5000 at UC Berkeley is running Ultrix version 4.3A. The DEC Alphas are running DEC OSF/1 version 1.3. The Ethernet and FDDI are local to the CSL. The ATM fiber links are dedicated to the CSL. All networks including the ATM switch are generally lightly loaded and experiments were performed during non-peak hours in lightly-loaded conditions.

3 Experimental Results

Our experiments led to the results shown in the following tables. Note that the maximum throughput rates reflect of our specific non-optimized applications. The experiments were performed using solely application level processes and X-window display functions available to the user program. No kernel level optimizations were made or required. As was discussed earlier, the maximum throughput experiment was simply the animation experiment without the image processing; and as such was intended to be an upper limit for performance of the animation experiment and not necessarily for the hardware or system itself.

3.1 Sequoia 2000 Net and ATM

Sequoia 2000 Net Maximum Throughput — DECStation 5000/240					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
1980.000	7.500	1.34	0.0	0	0.0

Sequoia 2000 Net Animation Statistics — DECStation 5000/240					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Kb/s)	Loss	Maximum	Average
6230.000	23.435	425	0.15	2	0.4

ATM Maximum Throughput — DECStation 5000/200					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
515.590	1.950	5.08	0.0	0	0.0
343.728	1.300	7.65	0.0	0	0.0
257.796	0.975	10.15	0.0	0	0.0
249.744	0.946	10.34	2.05	24	5.5
234.168	0.887	9.93	13.4	85	35.5

ATM Animation Statistics — DECStation 5000/200					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Kb/s)	Loss	Maximum	Average
5191.070	19.529	511	0.0	0	0.0
4671.575	17.576	568	0.0	0	0.0
4152.077	15.623	638	0.2	2	0.4
3632.579	13.670	713	2.4	11	6.2

3.2 Ethernet

Ethernet Maximum Throughput — DECStation 5000/200					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
346.066	1.301	7.69	0.0	0	0.0
281.160	1.065	9.53	1.2	31	3.2
257.664	0.976	9.45	7.4	28	19.5

Ethernet Maximum Throughput — DEC Alpha AXP 3000/500					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
330.000	1.250	8.02	0.0	0	0.0
277.200	1.050	9.32	1.4	150	3.6
224.400	0.850	9.31	19.1	58	50.5

Ethernet Animation Statistics — DECStation 5000/200					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Kb/s)	Loss	Maximum	Average
6108.984	22.459	434	0.05	1	0.1
5500.000	20.675	477	1.14	10	3.1
5312.150	19.530	493	1.23	8	3.3
5194.900	19.525	505	1.17	10	3.1

Ethernet Animation Statistics — DEC Alpha AXP 3000/500					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
750.000	2.817	3.54	0.0	0	0.0
660.000	2.480	3.96	1.45	10	3.8

3.3 FDDI

FDDI Maximum Throughput — DECStation 5000/240					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
171.864	0.650	15.6	0.0	0	0.0
140.610	0.530	19.6	0.01	3	0.02
128.890	0.488	17.9	15.8	200	41.8
124.992	0.473	16.3	23.4	133	61.7

FDDI Maximum Throughput — DEC Alpha AXP 3000/500					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
79.200	0.300	33.2	0.0	0	0.0
66.000	0.250	39.8	0.21	87	0.5
52.800	0.200	31.4	34.7	147	91.5

FDDI Animation Statistics — DECStation 5000/240					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Kb/s)	Loss	Maximum	Average
5195.070	19.529	511	0.03	4	0.15
4152.077	15.623	625	2.2	10	5.9
3632.579	13.670	693	5.2	20	13.6

FDDI Animation Statistics — DEC Alpha AXP 3000/500					
Transmission Time (ms)		Throughput	%	Packet Loss Per Frame	
Frame	Packet	(Mb/s)	Loss	Maximum	Average
600.000	2.254	4.42	0.0	1	0.005
550.000	2.066	4.67	3.2	21	8.5
500.000	1.878	5.24	1.3	7	3.5

3.4 Comments

From observations of the animations, the limit of packet loss was 2-3% before it became obvious and annoying. Previous simulations of this animation yielded a higher tolerance of loss. Those simulations consisted of three systems, a sender, a receiver and a subnet simulator. The subnet simulator artificially injected loss into the stream by not forwarding packets to the receiver. Packet loss was simulated as random loss only in the low priority component of the video stream.[2]

There appears to be two reasons for the lower loss tolerance in this experiment. The first is that in the simulations, we, of course, were able to control the component which would be subjected to loss. In current networks, there is not yet an ability to handle packets marked with priorities, therefore, loss was more indiscriminate than in our simulations.

The second reason is due to the fact that in a LAN environment, network bandwidth is abundant (this may not be true in heavily used networks) and is not the bottleneck. Rather, the bottleneck is host system performance, and the hierarchical coding is masking the loss due to the overflow of buffers in the host systems as they try to process packets (rebuild and display frames) near the limits of the system and/or application speed. This was observed in the animation as loss appeared in the same place in the frame consistently. This was due to the socket buffer overflowing while the system was displaying the just received and rebuilt frame. The maximum socket buffer size is approximately 50K bytes, which is sufficient for approximately 40 packets (1224 bytes each) in this animation. This is precisely where loss was observed. So, contrary to the earlier statement that an implicit priority is given due to the send order, it appears that packets 40...40+x actually received a lower priority due to the tendency to be dropped during a buffer overflow. Many things could be changed to improve or alleviate this, such as, increasing the socket buffer size, reordering the transmission of packets etc. But, it appears that the hierarchical coding will be better suited to the WAN environment where network throughput and jitter are the dominating factors of the animation, and delay and random loss/discard are more frequent as a result of network conditions rather than host buffer overflow. In our case of buffer overflow, the loss was not random at all, but occurred in bursts, just after the buffer overflowed; and typically, in the same point in the stream as data was being buffered while a frame was being displayed.

4 Conclusions

Hierarchical coding is now catching the attention of network designers who are trying to reconcile the great economic advantages of statistical multiplexing in high-speed packet- switching networks with the complexity of controlling the congestion phenomena that it produces. It promises to provide the necessary freedom to design networks that provide the needed quality of service for applications, without reverting into an absolute, tight control of all network resources.

In order to take full advantage of the potential benefits in the case of image browsing, in addition to the necessary application modifications, it is important that the database and the storage system is organized in a way that individual media components be directly accessible. This probably entails restructuring existing databases, plus incorporating new functionality into database management systems in order to provide the retrieval operations efficiently and according to the spirit of hierarchical coding.

To further reduce latency for applications retrieving images across wide area packet-switching networks, the components of high visual importance should be transmitted as high-priority traffic. Also, hardware support for video organized in a bit-plane format is necessary to increase the possible frame rates.

[2] Images from the sequence used in this experiment, subjected to simulated loss in the low priority component, are available and show good tolerance of loss. Because of their size and the need for a color printer, they are provided in a separate file, so that this report can easily be printed. They are not available in the hardcopy version of this document, but they can be obtained by anonymous ftp from `cs.ucsd.edu` as `/pub/csl/Papers/cs93-346-pict.ps.Z`.

References

[1] D. Ferrari and D. C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 8(3), pp. 368-379 (April 1990).

[2] D. Ferrari, J. Pasquale, and G.C. Polyzos, "Network Issues for Sequoia 2000," *Proc. IEEE Compcon Spring '92*, San Francisco, CA, February 24-28, 1992.

[3] J. A. Fine, T. E. Anderson, M. D. Dahlin, J. Frew, M. Olson, and D. A. Patterson, "Abstracts: A Latency-Hiding Technique for High-Capacity Mass-Storage Systems," Sequoia 2000 TR, March 1992.

[4] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 7(5), pp. 771-781 (June 1989).

[5] G. Karlsson and M. Vetterli, "Packet Video and Its Integration into the Network Architecture," *IEEE Journal on Selected Areas in Communications*, Vol. 7(5), pp. 739-751 (June 1989).

[6] A. Lippman and W. Buttera, "Coding Image Sequences for Interactive retrieval," *Communications of the ACM*, Vol. 32(7), pp. 852-860 (July 1989).

[7] Congestion Control in ATM Networks, Special Issue on, *IEEE Network*, Vol. 6(5) (September 1992).

[8] M. Nomura, T. Fujii, and N. Ohta, "Basic Characteristics of Variable Rate Video Coding in ATM Environment," *IEEE Journal on Selected Areas in Communications*, Vol. 7(5), pp. 752-760 (June 1989).

[9] J. Pasquale, K. Fall, and J. Forrest, "The Sequoia 2000 Network (S2Knet) Handbook," Sequoia 2000 Technical Report 92/5, U. C. Berkeley, November 1992.

[10] J. Pasquale, G. Polyzos, K. Fall, and V. Kompella, "Internet Throughput and Delay Measurements Between Sequoia 2000 Sites," Sequoia 2000 Technical Report 91/7, U. C. Berkeley, December 1991 (also available as U. C. San Diego Technical Report CS91-219, December 1991).

[11] G.C. Polyzos, "Hierarchical Coding of Images and Continuous Media for Transmission over Packet-Switching Networks," Technical Report CS 92-273, Dept. of Computer Science and Engineering, University of California, San Diego, La Jolla, CA, January 1993.

[12] L. Trajkovic and S. J. Golestani, "Congestion Control for Multimedia Services," *IEEE Network*, Vol. 6(5), pp. 20-26 (September 1992).

[13] J. Pasquale, G. Polyzos, E. Anderson, and V. Kompella, "The Multimedia Multicast Channel," *Proc. Third International Workshop on Network and Operating System Support for Digital Audio and Video*, 1992.