Proceedings of the 22nd IEEE Conference on Computer Networks (LCN'97), November 1997.

Adaptive Streaming of MPEG Video over IP Networks[1]

Ranga S. Ramanujan, Jim A. Newhouse, Maher N. Kaddoura, Atiq Ahamad, Eric R. Chartier, and Kenneth J. Thurber

Architecture Technology Corporation
P. O. Box 24344, Minneapolis, MN 55424
Tel: (612) 935-2035
FAX: (612) 829-5871
ranga@atcorp.com

Abstract

We present the design of an adaptive service for streaming MPEG video over a best-effort IP network environment. The adaptive video streaming service exploits the inherent adaptiveness of video applications to perform controlled and graceful adjustments to the perceptual quality of the displayed MPEG video stream in response to fluctuations in the QoS delivered by the three primary components in the end-to-end path of a video stream, i.e., the video server, the network, and the receiver. The design supports multiple video adaptation techniques that can be applied individually or in combination to adapt the transmitted video stream in response to fluctuations in the QoS provided by the underlying system. A novel aspect of the design is a video adaptation algorithm that selects the adaptation delivering the best perceptual quality for the video playback for a given QoS delivered by the underlying infrastructure. We describe a prototype implementation of the adaptive video streaming service and present the results of a performance evaluation of this prototype system.

1. Introduction

Several networked multimedia applications require on-demand retrieval and playback of stored compressed video. Essential to the functioning of such applications is an end-to-end video streaming service that streams stored compressed video over the network from a video server to a remote client (or receiver) for real-time playback. In an ideal scenario, the streaming service delivers the stream of video packets associated with a video clip to the receiver at a data rate enabling video playback at its encoded frame rate. Furthermore, ideally, no video information should be lost during network transport. In this case, we say that the video playback occurs at the quality at which it was encoded, or the *encoded quality*.

However, in a heterogeneous best-effort IP network environment, a video streaming service may have to operate under conditions that are less than ideal for playing back video at the encoded quality. The sustainable throughput supported by the network for the video stream may fluctuate over a wide range during the life of the video playback session. Under network congestion conditions, IP packets belonging to a video stream may encounter unacceptable delays and losses. Although the use of TCP may alleviate the problem of packet losses under network congestion conditions, it does not address the problem of timely delivery of video packets to the playback client. In fact, its slow-start mechanism for recovering from congestion exacerbates this problem by introducing additional delays [1]. Consequently, the delivered video is susceptible to uncontrolled degradation in perceptual quality.

Another problem encountered by a video streaming service in a heterogeneous network environment is the variation in the receiver's video playback capability. Not all receivers may have the resources necessary to play video at the encoded quality. The rate at which a receiver can play back an incoming video stream (i.e., *the displayed frame rate*) may be less than the encoded frame rate of the video. Moreover, the displayed frame rate may vary over time if the receiver application must compete for shared resources. A third problem is contention for CPU and I/O resources at the video server. This may cause variations in the transmission rate supportable by

[1] This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under contract number DAAH01-96-C-R048 monitored by the U.S. Army Missile Command. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of DARPA.

the server. Thus, to prevent uncontrolled degradation of the delivered video, a video streaming service for a heterogeneous IP network environment must be able to effectively handle dynamic variations in the quality of service (QoS) provided to the video stream by the underlying system consisting of the server, the IP network, and the receiver.

We present the design of an adaptive video streaming service that facilitates the streaming of MPEG video using the best-effort services of an IP network. The adaptive video streaming service exploits the inherent adaptiveness of video applications to perform controlled and graceful adjustments to the perceptual quality of the displayed MPEG video stream in response to fluctuations in the quality of service (QoS) delivered by the underlying system to the video stream. The two parameters of the transmitted video stream that may be adjusted by the adaptive video streaming service to adapt the perceptual quality of the delivered video are the signal-to-noise ratio (SNR) of the video signal and the displayed frame rate of the video.

SNR Adaptation: The signal-to-noise ratio of the video signal in an MPEG-encoded video clip is primarily determined by two factors: 1) the range of DCT coefficients used to encode the MPEG stream, the maximum range being 0 through 63; and 2) the precision with which the values of each of the 64 possible DCT coefficients are encoded, the highest precision being 8 bits for MPEG-1. Thus there are two major techniques for adjusting the SNR of a transmitted video stream, i.e., *spectral filtering* and *quantization filtering*. The spectral filtering technique discards a range of DCT coefficients from the transmitted video stream. The quantization filtering technique discards a given number of high precision bits (i.e., least significant bits (LSBs)) from the eight bit representation of the value of DCT coefficients.

SNR adaptation, using spectral filtering or quantization filtering or a combination of the two, can be used to implement controlled and graceful degradation of the delivered video quality when the data rate supported by the server or the network falls below that required for the encoded quality of the video or when the frame processing rate at the receiver drops below the encoded frame rate of the video. In the former case, reducing the SNR of the transmitted video reduces the amount of encoded information per video frame, thereby reducing the data rate required to deliver the adapted video stream to the receiver. In the latter case, reducing the SNR of the transmitted video stream reduces the amount of video information that must be processed (decoded) by the receiver, allowing the receiver to play back the adapted video at a faster frame rate. Other SNR adaptation techniques (e.g., spatial filtering and color filtering) are also available but are not further considered in this paper.

Displayed Frame Rate Adaptation: There are two major techniques for reducing the rate at which video frames are processed at the receiver: *frame dropping* and *playback dilation*. With frame dropping, a subset of the frames associated with a video clip are discarded at the server end. These frames are thus skipped by the player, reducing the player's frame processing rate. Note that frame dropping is in reality another form of SNR adaptation, since the receiver obtains less accurate data about the video sequence, and must either repeat preceeding frames or interpolate, both of which result in a reduction in perceptual quality. In a generalization of this technique, the server may drop data at the macroblock level, so that a better name for this technique would be *macroblock filtering*.

With playback dilation, all the frames of the source video are delivered to the receiver. However, the rate at which video frames are processed at the receiver is intentionally reduced below the encoded frame rate of the video. This results in the dilation or expansion of the playback time of the video. Displayed frame rate adaptation, using macroblock filtering or playback dilation or a combination of the two, can be used to handle reductions in the data rate supported by the server or the network for the video stream as well as reductions in the frame processing rate supported by the receiver.

Although any of the four major techniques described above (i.e., spectral filtering, quantization filtering, macroblock filtering, and playback dilation) may be used individually or in combination to adapt the transmitted video stream to changes in the QoS delivered by the server, the network, and the receiver, the impact of each of these choices on the perceptual quality of the delivered video may be significantly different. Also, different video clips may undergo varying perceptual quality degradation levels for a given adaptation choice. Furthermore, perceptual quality is a subjective phenomenon which may vary among human observers. The choice of adaptation technique(s) also has a varying impact on the server/network throughput and receiver frame processing rate required to support the adapted video stream.

For any given QoS delivered by the underlying system for a video session, the adaptive video streaming service selects an adaptation that achieves:

1) A data throughput appropriate for the

underlying system's current QoS level
2) The "best" feasible perceptual quality for the video stream played by the receiver
3) Operator discretion in formulating the definition of what constitutes the "best" perceptual quality.

Section 2 of this paper discusses related work and outlines the significance of this work. Section 3 presents an overview of the architecture of the adaptive video streaming service. Section 4 describes a prototype implementation of the adaptive video streaming service and presents a performance evaluation of this prototype system. Section 5 presents our conclusions.

2. Significance and Related Work

The ability of the adaptive video streaming service to accommodate multiple adaptation techniques for the transmitted video stream, and to make a judicious choice among the adaptation techniques for each situation, is what distinguishes it from existing approaches for adaptive video streaming. Existing approaches generally use a single technique for adapting the video to variations in the QoS supported by the underlying system [2, 3, 4, 5, 6, 7, 8]. In contrast, the adaptive video streaming protocol allows for a dynamic choice among four adaptation techniques that may be applied in any combination to optimize the perceptual quality of the delivered video for a given QoS level supported by the server, the network, and the receiver for the video stream. This versatility of the adaptive streaming service makes it much more effective than existing video streaming implementations in accommodating a wide range of variations in the QoS of the underlying system without appreciably degrading the quality of the delivered video.

The ability to handle variations in resource availability of all three components of the underlying system also distinguishes our approach for video streaming from most existing techniques for streaming of MPEG video. Existing approaches have primarily focused on QoS variations within the network and/or the receiver. However, the source of the bottleneck may vary among the server, the network, and the receiver during the course of any given streaming video clip. Ideally, the QoS delivered by all three components of the underlying system should be dynamically factored into the adaptation strategy of the streaming video protocol.

Probably the most important factor that distinguishes our adaptive streaming video protocol from other streaming video research is its perceptual-quality-centric approach. In order to characterize the relative perceptual quality of an adapted video stream, we have developed a quantitative approach to video perceptual quality [9]. Although it is not the primary focus of this paper, briefly, our approach generates a real number in the range (0, 1) representing the perceptual quality of an adapted video segment relative to the encoded quality of the segment. The technique starts with the relative SNR of the adapted versus unadapted video segments, and also incorporates the first derivative of relative SNR to discourage rapid fluctuations in quality which could be annoying to the viewer. The raw SNR factors are mapped through a function approximating perceptual quality as perceived by humans. The resulting SNR-based perceptual quality can then be scaled appropriately to reflect any additional perceptual degradation due to playback dilation. The quantitative method was calibrated and verified via perceptual quality experiments using human subjects.

One of the earliest implementations of networked streaming MPEG video was the Berkeley Continuous Media Player (CMP) [3]. The CMP uses an ad hoc software feedback mechanism to adjust the frame rate sent by the server. It supports a single technique for video adaptation, i.e., playback dilation to address resource depletion in the network and receiver only. Recent CMP performance results for various adaptive frame rate policies are given in [10]. Another example of real-time multimedia transport over the Internet is the distributed MPEG video and audio retrieval system developed at the Oregon Graduate Institute of Science and Technology [4]. This system includes real-time synchronized playback of MPEG video and audio streams, user specification of presentation quality in number of frames-per-second (fps), and simple best-effort QoS control. A single adaptation technique, i.e., frame dropping, handles network or receiver resource decreases during a video playback session. The specialized video datagram protocol implemented within Vosaic, at the University of Illinois, employs a similar strategy for streaming MPEG video over the Internet [5].

The Distributed Multimedia Research Group at Lancaster University has focused on the QoS aspects of multimedia [6, 7]. In conjunction with this research, the group has developed a filter-agent toolkit for filtering MPEG streams along the quantization, spectral, macroblock, and color dimensions. Unlike our approach, this technique does not support dynamic selection of the adaptation technique to maximize perceptual quality of the delivered video for a given operational scenario. Also, server resource contention is not addressed. Performance results for their approach in a multicast environment are given in [11]. StreamWorks [8] is a commercial product that streams MPEG video and audio

from a StreamWorks server to a StreamWorks client in real-time via a UDP channel. At session set-up time, it determines the sustainable throughput of the end-to-end connection and the frame rate supported by the receiver. It then employs frame dropping to accommodate the video stream within the supported network and receiver throughput. There is no mechanism to handle changes in the system QoS after the video playback has started. Other commercial streaming video implementations include VDOnet's VDOLive, VXtreme's Web Theater, and Progressive Networks' RealVideo. VDOLive utilizes wavelet compression with dynamic quality adjustment, whereas our protocol is MPEG-based. Web Theater uses Real-Time Protocol (RTP) to ensure timely video packet delivery, whereas our approach does not require RTP. RealVideo uses Real Time Streaming Protocol (RTSP).

3. Architecture of Adaptive Video Streaming Service

Figure 1 presents an implementation model of the adaptive video streaming service. As shown in the figure, it may be viewed as an application that uses TCP and UDP services to implement an adaptive application-level protocol for streaming video from a video server to a playback client (i.e., receiver). The receiver end maintains a video buffer from which the video player retrieves video frames for playback. The player consumes frames at a rate no greater than that at which the video was encoded. The receiver buffers a predetermined number of video frames before allowing the player to start playback in order to help smooth small jitters in the inter-frame arrival intervals of the video stream.

The adaptive video streaming service continuously monitors the QoS supported by the video server, the network, and the receiver for the video stream. The server end of the streaming service dynamically performs any necessary adaptation to the transmitted video stream to maximize the perceptual quality that can be delivered for the end-to-end QoS supported by the underlying infrastructure. To reduce the computational requirements required to perform on-the-fly adaptations of the video stream, an off-line process is used to convert a standard MPEG file into a customized *layered MPEG* format. Spectral, quantization, and/or macroblock layering is utilized by the transcoder in order to greatly facilitate the implementation of spectral filtering, quantization filtering, and macroblock filtering by the server end of the adaptive video streaming service.

The layered MPEG format produced by the transcoder may be viewed as a compound video stream consisting of a base layer video stream of minimal-acceptable SNR quality and one or more enhancement layers that successively enhance the SNR quality of the base layer video. The SNR quality associated with a video stream consisting of all the layers is the same as that delivered by the unlayered, source MPEG clip from which the layered video was derived. A subset of the stored layers is transported to the receiver end of the adaptive video streaming service where it may be converted to a standard MPEG stream before being delivered to a hardware or software video player.

There are several feasible alternative playback and transport methodologies. It would typically be a waste for a software-based receiver/player to re-encode the merged, partially-decoded data into a standard MPEG stream only to immediately decode it again. Instead, the player might simply continue the decoding and displaying process. This is the approach utilized by our prototype player implementation. In an alternative transport approach the server might reassemble the adapted MPEG video stream into standard MPEG format and then transport the MPEG video over the network, thus alleviating the player of the reassembly burden. However, it is our opinion that typical end-user workstations will soon be capable of handling this task in addition to the playback task. Therefore, the adaptive streaming video protocol was designed to off-load the reassembly task from the possibly-overloaded server. This approach is also more flexible in the sense that it allows the server to send excised video layers to the receiver at a later time if the system QoS should improve dramatically (given sufficient receiver buffering) and it facilitates extension of the protocol to a variable-QoS multicast environment.

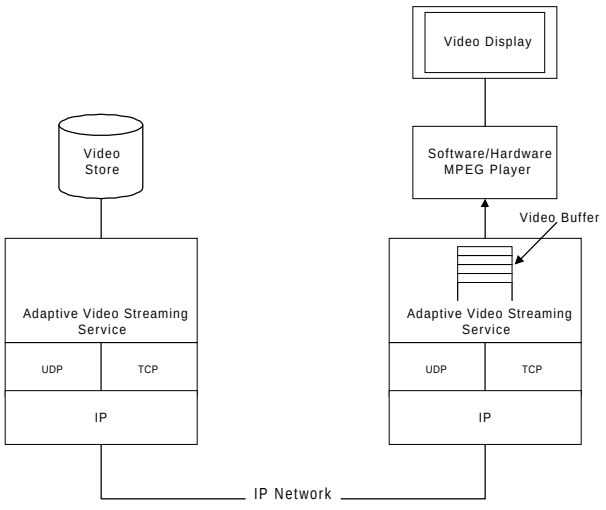

Figure 1: Implementation model of adaptive video streaming service

Section 3.1 describes the perceptual-quality-based

adaptation methodology. Section 3.2 describes the off-line processes that are used to convert an MPEG file into a layered MPEG format. Section 3.3 presents the protocol that is used by the sending and receiving sides of the adaptive video streaming service to transport the video stream from the video server to the video player. It also describes how the adaptation information derived by the off-line process is used at run-time by the adaptive video streaming service to perform its functions.

3.1 Adaptation Methodology

An off-line process is also used to derive information that can be used by the server to determine which adaptation to apply to optimize the perceptual quality of the video for the QoS that is supported by the server, network and receiver (i.e., the end-to-end QoS). Conceptually, this information is in the form of the table shown in Figure 2 which lists a finite set of adaptations, ranking them in terms of their impact on the perceptual quality of the delivered video stream. For instance, the video stream with no adaptation (Adaptation 1) provides 100% of the encoded quality, whereas Adaptation *n* provides a baseline perceptual quality beyond which further degradation is deemed unsatisfactory. Each table entry contains the required system QoS in terms of sustainable server transmission rate, network data rate, and receiver frame processing rate, as well as the relative perceptual quality. Each adaptation ID identifies a specific combination and adaptation level of the four adaptation techniques employed by the protocol.

Adaptation ID	Required Server QoS	Required Network QoS	Required Receiver QoS	Relative Perceptual Quality
Adaptation 1				1.0
Adaptation 2				.95
. . . .				. . . .
Adaptation *n*				.2

Figure 2: Structure of adaptation table used by the streaming service

The adaptation table could conceivably be either static or dynamic. The latter offers the advantage of being able to adapt to varying end-user definitions of what constitutes perceptual quality. However, dynamic adaptation table construction may require extensive server computations. Since there is a four dimensional *adaptation space* associated with the four types of adaptations, and there could be many discrete adaptation levels in each of these dimensions, the adaptation table may contain many entries. The perceptual quality for each entry would need to be calculated in real-time to enable dynamic end-user definition of perceptual quality.

Our initial experimental prototype implementation of the adaptation table utilizes a compromise hybrid approach which effectively linearizes the three SNR dimensions of spectral filtering, quantization filtering, and macroblock filtering into a single SNR adaptation dimension, leaving playback dilation as an independent dimension. When the video clip is layered and installed on the server, a separate off-line process derives a fixed perceptual-quality ordering of all SNR-only adaptations in conjunction with a parameterized definition of SNR-based perceptual quality. This quality information may be stored with the installed video along with associated data characterizing each SNR adaptation's required QoS.

The experimental implementation layers the video in a manner such that each of the *m* entries in the stored SNR-adaptation table corresponds to a contiguous subset of the *m* video layers. Full-quality SNR-Adaptation 1 corresponds to the encoded-quality unfiltered video consisting of all layers; SNR-Adaptation 2 corresponds to filtering out a single enhancement layer; and SNR-Adaptation *m* corresponds to transmitting a single base layer. The server uses this static table to control adaptation in the SNR dimension and dynamically factors-in the remaining playback dilation dimension. The end-user thus has a measure of dynamic control over the effective adaptation table based upon their relative preference for playback dilation versus SNR degradation (e.g., via a slider control on the playback interface), while the installer controls the preference for spectral versus quantization versus macroblock filtering. A fully-dynamic implementation is also feasible.

3.2 Off-line Transcoding of MPEG Video

The layered MPEG video stream may be derived using either spectral layering, quantization layering, macroblock layering, or a combination of all three techniques. Consider the following example of a layered MPEG video stream that is derived using both spectral layering and quantization layering. The example layered MPEG stream consists of three layers where layer 3 contains the LSB of the DCT coefficients of the source MPEG file, layer 2 contains the 7-bit encoding of the MSBs of the 40 higher order DCT coefficients (i.e., coefficients 24 through 63) and layer 1 (or the base layer) contains the 7-bit encoding of the MSBs of the 24 lower order DCT coefficients (i.e., coefficients 0 through 23).

The number of candidate layerings for a given video

that may be obtained using a larger number of layers and all three SNR adaptation techniques is potentially enormous. It is therefore essential to select the layering scheme that provides the best perceptual quality for a given QoS supported by the underlying system. The task of selecting an effective layering scheme is prohibitively time consuming and extremely tedious. We have therefore developed a Video Quality Analysis (VQA) tool that automates some of the time consuming aspects of this task. The VQA tool automatically analyzes the source MPEG video and a given candidate layering of the video and estimates the perceptual quality for each filtering level of the candidate layering relative to the source video for a given frame-rate [9]. These quantitative perceptual quality estimates may be used both to select an optimal layering structure as well as to derive the SNR-adaptation table for the selected layering.

The layering of the MPEG video must be achieved in a manner that preserves the compression efficiency of the original video. That is, the size of the full-quality layered video stream that is transmitted over the network should not exceed that of the original, unlayered MPEG video file. Also, the layering scheme should facilitate a computationally efficient implementation of the decoder for the layered stream.

We have implemented a tool that accomplishes an efficient transcoding of an MPEG video into a layered stream. For a typical layering, the coding efficiency results in a layered MPEG stream whose size is no larger than that of the source MPEG file. The transcoder implements an efficient two-pass Huffman algorithm which is applied in parallel to each layer's {run-length, value} pairs. A novel parse-treelet technique is used in the Huffman codec in order to achieve an average computational complexity per parsed symbol of O(1.05) while at the same time requiring only a modest amount of memory beyond a minimal-size binary-parse-tree implementation. Further details of the implementation are beyond the scope of this paper.

The output of the transcoder consists of two components, i.e., the layered video data and the video descriptor. An appropriate subset of the video layers may be transmitted to the receiver by the sending end of the adaptive video streaming service. The video descriptor contains information describing the layered video data, including the system QoS required by each layer. This information allows the sending end of the adaptive video streaming service to determine which video data layers must be filtered out when SNR adaptation must be applied to the transmitted video stream. This meta-data is not transmitted over the network.

Figure 3 is a conceptual view of the structure of the output produced by the transcoder for an example three-layered video stream. For each frame of the video, the video descriptor contains information identifying the frame type (e.g., I, P, or B frame), the beginning and end of the frame in the byte aligned video data stream, and the beginning and end of each of the three layers associated with the frame. If an adaptation calls for a single layer of the video stream, the adaptive video streaming service need only retrieve the layer 1 data associated with each frame from the layered video data before transmitting it over the network.

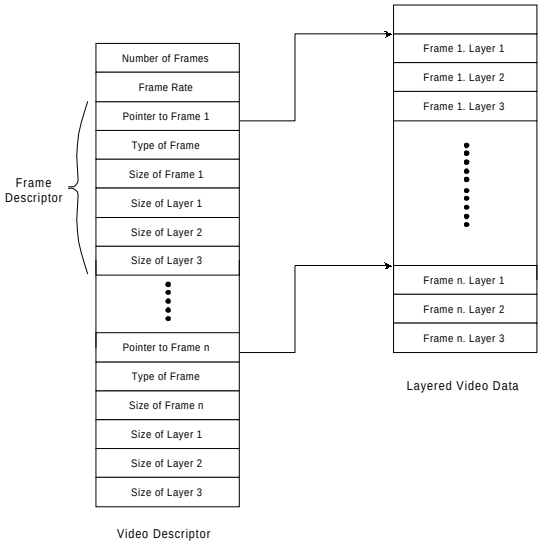

Figure 3: Structure of transcoder output

3.3 Adaptive Video Streaming Protocol

The receiver and server ends of the adaptive streaming service communicate through two channels: a uni-directional video streaming channel, implemented over UDP, that is used by the server end to stream layered video data to the receiver; and a bi-directional signaling channel, implemented over TCP, that is used by the receiver and server ends to exchange signaling or control messages. The server end of the adaptive streaming service employs a dynamic rate-based scheme for controlling the data transmission rate. The following paragraphs describe the mechanisms implemented by the adaptive video streaming protocol to monitor the QoS supported by the server, network, and the receiver, and to dynamically adapt the transmitted video stream to variations in the QoS supported by them.

Adaptation ID	SNR Adaptation (video data transmitted)	Temporal Dilation (fps)	Required Server/Network Throughput (Mbps)	Relative Perceptual Quality
Adaptation 1	All 8 bits, 64 coeff, I/P/B MBs	30	2.0	1.0
Adaptation 2	7 MSBs, 64 coeff, I/P/B MBs	30	1.0	.85
Adaptation 3	7 MSBs, 64 coeff, I/P/B MBs	$25 \leq N1 < 30$	N1/30	$.71 \leq Q1 < .85$
Adaptation 4	7 MSBs, 64 coeff, I/P MBs	25	.6	.5
Adaptation 5	7 MSBs, 64 coeff, I/P MBs	$20 \leq N2 < 25$	(N2/25)*.6	$.4 \leq Q2 < .5$
Adaptation 6	7 MSBs, 24 coeff, I/P MBs	20	.4	.35
Adaptation 7	7 MSBs, 24 coeff, I MBs	20	.25	.2

Figure 4: Adaptation table for example video stream

Whenever the operation of an adaptation mechanism needs illustration, we will use an example video stream consisting of three spectral/quantization layers similar to the previous example, but augmented with the addition of I/P/B macroblock layering and playback dilation. The adaptation table for this example video stream is shown in Figure 4. The encoded frame rate of this video is 30 frames/second. For simplicity, only a subset of useful adaptations are considered for this example. In the table of Figure 4, the adaptations are ranked based on the relative perceptual quality of the adapted video stream. Thus, Adaptation 2 which drops all LSBs is more desirable than Adaptation 3 which drops LSBs and also dilates the playback time of the video stream by up to 20% (i.e., when the video is played at 25 fps). However, Adaptation 3 delivers better perceptual quality than Adaptation 4 which also drops all B macroblocks.

3.3.1 Adaptation to Server Throughput Variations

The server uses rate-based flow control to space out packet bursts. The time interval between two consecutive bursts is set at the beginning of the session and remains fixed until the end of the session. However, the number of bits that are sent in each burst varies depending on the desired transmission rate. The number of bits to be sent in each burst is computed as (*desired data rate * burst interval - adjustment*). The variable *adjustment* is used to account for the fact that the actual number of bits sent during each burst interval may be more or less than the desired burst size because the burst's size is restricted by the granularity of the packet sizes. For example, if the desired burst size was 10,000 bits and the sizes of the next two available packets were 6,000 and 5,000 bits, then an additional 1,000 bits will be sent. In this case, the value of the *adjustment* variable is set to the number of extra bits sent during this interval, i.e., +1,000. During the next interval these extra bits are debited from the desired burst size for the interval. Thus, the desired burst size for the next interval would be 9,000 bits in this example. This method also compensates for any shortage, which is credited during the next interval, as well as for any delay in scheduling the server process. Thus, the *adjustment* variable used in the calculation of the burst size for each interval averages out variations due to packet granularity and processor workload.

The server end of the streaming service monitors its recent transmission rate, processor utilization, and disk buffer occupancy to form an estimate of its sustainable transmission rate. A sustainable throughput level that is less than that required for the current adaptation of the video stream is indicative of the inability of the server to support the video stream at this quality. The server end of the adaptive video streaming service then consults the adaptation table for the video stream to determine which adaptation would provide the best perceptual quality for the throughput currently supported by it and applies this adaptation to the video stream. Suppose, for the example video described above, the measured sustainable transmission rate of the server while streaming the full-quality video is 1.1 Mbps instead of the required 2 Mbps. In this case, the streaming service will implement Adaptation 2 by not retrieving the layer(s) containing the LSB data from the layered video file residing in secondary storage.

If the server can maintain the throughput for the adapted video over a predetermined amount of time, the adaptive streaming service probes the server to determine whether it might support additional throughput. If the server appears able to sustain a throughput that would support an adaptation with a better perceptual quality than the current one, the video stream is adapted accordingly.

3.3.2 Adaptation to Network Throughput Variations

The adaptive video streaming protocol uses additive increase and multiplicative decrease schemes to adapt the

data rate of the transmitted video to variations in the throughput supported by the network. These schemes have been shown to ensure fairness in shared network environments such as the Internet.

The receiver end of the adaptive video streaming service periodically sends a *status report* to the server end of the service over the signaling channel. Among other things, the status report includes the average data rate of the received video stream over a preset number of intervals. The status report also contains the packet loss rate observed by the receiver over this interval as well as an indication of whether the current packet loss rate exceeds the receiver's threshold of acceptable packet losses. To facilitate detection of packet losses by the receiver and its calculation of the packet loss rate, each UDP packet carrying the layered video data carries a sequence number in its payload portion. To prevent fragmentation of packets that may complicate the detection of packet losses, the sizes of the packets are kept below the MTU for the network path between the server and the receiver.

Adaptation of the transmitted video is initiated by the streaming protocol when the number of receiver-initiated status reports indicating unacceptable packet losses exceeds a preset threshold within a monitoring time interval at the server. When this occurs, the server selects an adaptation for the video stream that delivers the best perceptual quality at a data rate that is no more than half of the current transmission rate of the video. For the example video, suppose unacceptable losses occurred while the full-quality video was being transmitted by the server at 2 Mbps. In this case, the adaptive streaming protocol will apply Adaptation 2 that reduces the required network throughput to 1 Mbps.

If unacceptable packet losses persist after this initial adaptation, the protocol applies another adaptation that reduces the data rate of the stream at least another 50%. In the case of the example video, this second adaptation would be Adaptation 5 where a playback dilation will be applied to play a video stream consisting of all MSBs of all coefficients for I/P macroblocks at about 21 fps. The adaptation table shown in Figure 4 yields a required network throughput of .5 Mbps for this adaptation.

On the other hand, if the adaptation operates without experiencing unacceptable packet losses for a preset amount of time, the adaptive video streaming protocol probes the network to determine whether it can support additional network throughput at this time. It uses an additive increase scheme to gradually increase the transmission rate of the video stream on a temporary basis. After each additive step, the server monitors the data arrival rate reported by the receiver in its status reports. If the network cannot support a throughput at the current level, then the protocol falls back to the transmission rate from which the probing process started. The server end ignores the arrival rate reported in the first few receiver status reports after each additive increase. This is needed to prevent oscillations in the protocol. A similar method is used to respond to variations in sustainable network throughput which are not accompanied by significant packet losses. However, in this case an additive increase and subtractive decrease methodology is appropriate.

At the beginning of a video session, the receiver and server ends of the adaptive video streaming service perform a brief experiment to determine the network and server throughput that can be initially supported. The receiver end of the video streaming service buffers a predetermined number of frames. It also monitors the rate at which the data was received and reports it to the server end along with an indication of whether unacceptable packet losses were encountered. The adaptive streaming service selects an initial adaptation to match the sustainable throughput level, under the default assumption that the receiver will be able to handle the implied playback rate. The receiver requests retransmission of all lost packets and allows the video playback to start only after the initial set of buffered video frames has been reliably received. This ensures that the initial portion of an MPEG video stream, which contains critical header data for video playback, is reliably delivered to the receiver.

3.3.3 Adaptation to Variations in Receiver Frame Processing Rate

The video packets transmitted by the server contain information identifying the boundaries of frames carried within the packet, i.e., the offsets to the beginning of a frame and the end of a frame are carried within the payload of each UDP packet. Of course, if the size of a frame is larger than the MTU for the network path, then it may be fragmented across multiple packets. In this case, the frame boundary information may straddle multiple packets. The frame boundary information allows the receiving end of the adaptive video streaming service to monitor the rate at which frames are consumed from its video buffer by the video player.

The receiver status report, that is periodically sent to the server, carries the average rate at which frames were processed by the MPEG player over the last few reporting periods. The server end of the adaptive video streaming service uses this information to determine whether the

receiver can process frames at the rate required to support the current adaptation level, say Adaptation *j*. If the frame processing rate is less than that required for adaptation *j*, the adaptive streaming service consults the adaptation table and selects a new adaptation which is appropriate for the currently supported receiver frame processing rate. Although it may initially appear that only a displayed frame rate adaptation would suffice to properly adapt to the supported receiver playback rate, experience has shown that required receiver processing resources are roughly linearly proportional to received data rate. Therefore, any Adaptation *k* (where $k > j$) which requires a data rate equivalent to a straightforward dilation of adaptation *j* would typically suffice. Thus, the protocol simply consults the adaptation table to select the highest quality adaptation satisfying this constraint.

For the example video, suppose the server is streaming full-quality video to the receiver. Let the current frame processing rate reported by the receiver in its status reports be 10 fps. Rather than dilating the video stream to 10 fps in order to immediately solve the rate mismatch problem, Adaptation 4 is selected from the adaptation table because it has a required throughput of .6 Mbps which is less than 1/3 of the full 2 Mbps throughput. The selected adaptation would consist of the MSBs of all coefficients for I/P macroblocks at 25 fps. This would typically have a higher perceptual quality than full-SNR-quality video dilated to 10 fps.

3.3.4 Variations in QoS of Multiple Elements

The above discussion focused on handling of QoS variations for each of the three system elements, i.e., the server, the network, and the receiver, considered separately. If resource depletion is experienced by more than one system element simultaneously, the adaptive video streaming protocol uses the adaptation table to select an adaptation that would effectively handle QoS deterioration in all of the affected resources. Thus, the protocol effectively selects the adaptation required to accommodate the system element that is currently the bottleneck. Considering the example video, suppose a situation arises where the server is initially streaming full-quality video to the receiver at 2 Mbps and the throughput offered by the server suddenly drops to 1.1 Mbps. At the same time, the throughput rate offered by the network to the video stream decreases to .9 Mbps and the frame processing rate at the receiver decreases from 30 fps to 15 fps. In this case, the receiver was still processing full-quality frames when it reported its processing rate so that it has an equivalent throughput capacity of 1 Mbps. Therefore, the network is the bottleneck and the new target system throughput would be .9 Mbps. The streaming video protocol would select Adaptation 3 from the adaptation table which would transmit the MSBs of all coefficients for I/P/B macroblocks at a temporal dilation of 27 fps. Thereafter, it will use a combination of probing techniques described in Sections 3.3.1 through 3.3.3 to determine whether Adaptation 2 and ultimately Adaptation 1 can be supported by the entire system.

4. Performance Assessment

We have implemented a subset of the capabilities of the adaptive video streaming service in a laboratory network testbed environment that supports tools and mechanisms for simulating various network congestion scenarios and delays encountered in an operational network. The prototype implementation of the adaptive video streaming service supports video adaptations in response to variations in network throughput and frame processing rates of receivers. The streaming protocol uses simplified adaptation rules. It exclusively applies SNR adaptations to the transmitted video stream to recover from network congestion and uses playback dilation for handling slow receivers. The utilization of playback dilation is primarily due to the fact that the prototype player is rather slow, as opposed to any inherent technical hurdles. The prototype adaptive streaming service supports up to three SNR layers for layered MPEG video data. These layers are derived through quantization filtering. Software for spectral filtering of MPEG video has also been implemented. SNR layering using a combination of these two techniques and/or macroblock filtering is to be implemented.

The component elements of the software implementing the experimental adaptive video streaming service include:

1. A transcoder that converts a standard MPEG file into a layered MPEG format. The tool is hosted on an HP/UX machine but can be easily ported to other UNIX platforms.

2. An automated Video Quality Analysis (VQA) tool, running on an HP/UX machine, that automates the estimation of the relative perceptual quality of a given SNR layering of an MPEG video sequence. An overview of the operation and performance of this tool is presented in reference [9].

3. A video server that implements the server end of the adaptive video streaming service. It runs on an HP/UX platform, but can be ported with little effort to other platforms.

4. A video player derived from the Berkeley MPEG Player [3] that decodes and plays back a layered MPEG stream from the network in real time. The functions of the receiving end of the adaptive streaming service are integrated within the video player. The player software is designed to run in a Linux environment on a PC platform.

The following paragraphs present the results of preliminary testing and evaluation of the various components of the prototype adaptive video streaming service.

4.1 Protocol Performance

Our analysis of the performance of the adaptive video streaming protocol shows that the dynamic bandwidth management scheme successfully adapts to wide variations in available network capacity. In particular, the performance data confirm that the protocol exhibits a rapid and successful response to the onset of congestion, reducing its transmission rate to a level sufficient to alleviate the problem of network packet loss. The protocol also successfully resumes the target transmission rate after the congestion has disappeared. Figure 5 shows network performance data for a network testbed congestion experiment with three SNR layers. The primary curve plots total traffic rate on a single Ethernet segment downstream of the router separating server from receiver. This total includes traffic associated with the TCP signaling channel, the UDP video streaming channel, the artificially induced congestion, and a small amount of network management traffic. The vertical scale measures traffic rate in units of bytes per second. The horizontal scale shows elapsed time in seconds. Overlaid on top of this plot are vertical bars representing the timing of various signaling messages associated with the adaptive video streaming protocol.

At the beginning of the experiment, we see that traffic associated with the video stream initially ramps up from zero during the video session setup stage. At point A, video playback begins and the adaptive video streaming stage begins. At point B, a stable streaming rate for the layered video data is reached. The network is then artificially loaded with traffic from a traffic generator, resulting in a network traffic spike. However, this traffic is insufficient to induce either lost packets or sufficient throttling of network throughput for the video stream, thus indicating that the protocol is able to cope with normal network traffic fluctuations. Severe artificial traffic is then injected, resulting in a larger network traffic spike which exceeds the forwarding capacity of the router. Near point C, shortly after the onset of congestion, the protocol initiates several signaling messages. Upon detection of congestion, the server drops a layer of the video data, resulting in a constrained transmission rate. This is not quite enough to prevent the router from losing packets, so near point D the server drops an additional layer in response to the player's reports of continued unacceptable packet losses.

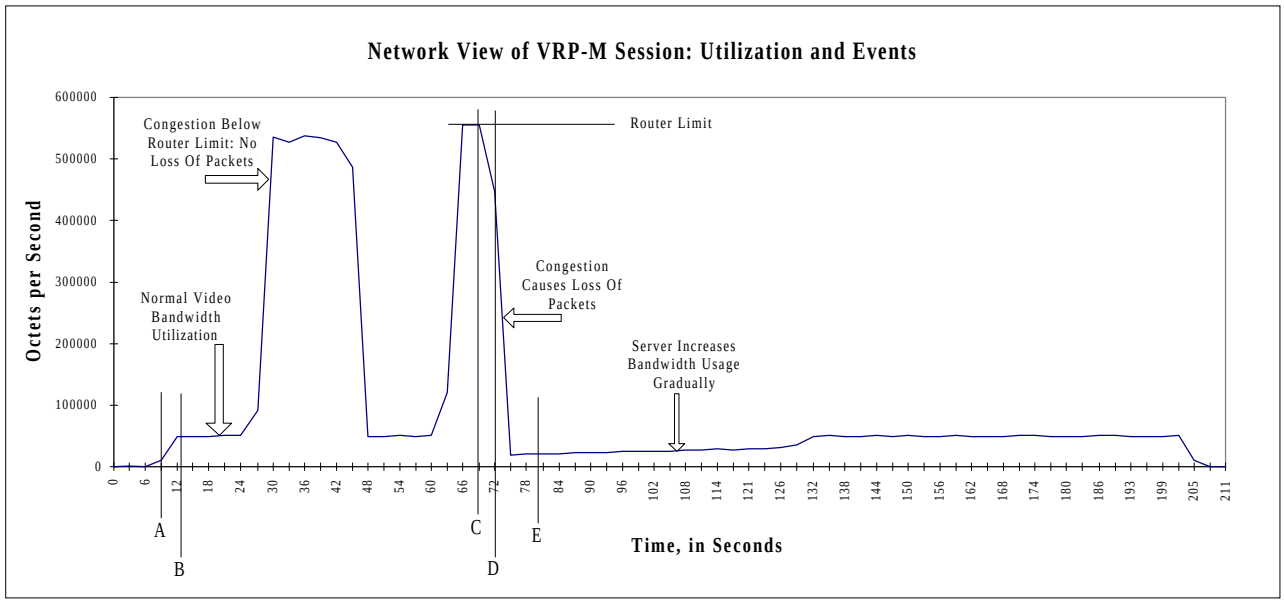

Figure 5: Protocol performance during congestion experiment

It is difficult to determine the exact timing of these events from the raw traffic curve per se which consists of one-second moving averages, so that these events were confirmed using traces from a traffic analyzer. By this time, the artificial congestion has been removed, leaving only the reduced level of video traffic, which is roughly 30% of the normal transmission rate. Near point E, the player sends the server a periodic status message that causes it to initiate an adaptation experiment to detect whether the congestion has possibly disappeared. The gradual ramp-up in the traffic curve starting at this point shows the server gradually increasing its transmission rate without increasing the number of layers. Although the traffic curve appears to exhibit a continuous increase at this point, it is really a fine-grained step-wise increase. Since the original congestion has in fact disappeared and no further congestion has been introduced, the server eventually increases the number of layers transmitted by one, beginning a second adaptation experiment to see if it can be increased further. This experiment is successful as well and the number of layers is increased to three (i.e., full-quality video). Network traffic continues at the full-quality level until the end of the video session.

Our experiments with a sample set of 30 video clips revealed that removing the least significant enhancement video layer (layer 3) typically results in a 40-60% reduction in required video throughput and an associated reduction in perceived video quality of 5-20%. Removing the next enhancement layer (layer 2) typically results in an additional 15-25% throughput reduction (relative to the original) and an additional 15-25% quality reduction. The resulting video quality versus required throughput function varies widely between different video clips, but our research has shown that it is generally a strongly convex curve. Thus, a very large reduction in required throughput is usually possible without severe degradation in video quality, simply by selecting an adaptation that dynamically filters one or more enhancement layers. This phenomenon contributes in large part to the protocol's ability to achieve graceful degradation.

4.2 Transcoder Efficiency

The off-line transcoder tool was evaluated in terms of the compression ratio of the transcoding process, i.e., the ratio of the size of the layered video stream transmitted over the network to the size of the source MPEG video from which the layered stream was derived. For a sample set of 30 MPEG video clips that were transcoded into three video layers, the average compression ratio of the transcoding process was 98%. Thus, on average there was a 2% reduction in the size of the layered video file versus the source MPEG video file.

Reductions of up to 8% were achieved for several videos. However, a few clips (typically clips containing only I frames) increased slightly in size after the transcoding process. The minimum and maximum compression ratios obtained for the sample set of clips were 92% and 103%, respectively. The efficiency of the transcoding process can be attributed to the two-pass Huffman compression technique that is employed by the transcoder to produce layered video. Our research suggests that increasing the number of layers results in slightly degraded compression efficiency. However, we anticipate even better transcoder compression efficiency in a future transcoder revision which will Huffman-encode various side information.

4.3 Server Performance

We measured the average CPU utilization of the video server process for the set of sample clips used during the experiment. Server utilization was measured on an HP/UX machine containing an 80 MHz PA-RISC processor. Server CPU utilization for each video session averaged between 4% to 5% for typical 320x240 pixel clips on this 1994-vintage HP workstation. By extrapolation, this machine could potentially support up to 20 simultaneous streaming video sessions. We believe that with current processor, storage, and network access technology, a video server machine can be configured at low cost to support several hundred concurrent video streams using the adaptive video streaming protocol.

4.4 Player Performance

Our experimental implementation of the video player for handling layered video streams was derived from the existing Berkeley MPEG player. By leveraging existing player technology, we were able to focus on issues of protocol design, implementation, and efficiency as opposed to player implementation issues. On the 200 MHz Linux PCs where the prototype player was evaluated, the original Berkeley player was incapable of playing MPEG video from local disk at the encoded frame rate for all but a few of the sample videos used for the experiments. Since our prototype player adds additional video streaming service and client application layer overhead to the existing playback code, it requires even more processing time per frame, and thus the attainable playback rate is degraded slightly from that of the stock Berkeley player for these clips. On a given PC, we observed up to 10% reduction in the playback rate of three-layered streaming video when compared to the fastest playback rate achievable for the source MPEG clip when played back from local disk by the original Berkeley MPEG player.

5. Conclusion

In this paper, we presented the design of an adaptive video streaming service for streaming MPEG video over a best-effort IP network environment. The adaptive video streaming service exploits the inherent adaptiveness of video applications to perform controlled and graceful adjustments to the perceptual quality of the displayed MPEG video stream in response to fluctuations in the QoS delivered by the three primary components in the end-to-end path of a video stream, i.e., the video server, the network, and the receiver. The approach supports four orthogonal video adaptation techniques that may be selected dynamically and applied in any combination to adapt the transmitted video stream in response to fluctuations in the QoS provided by the underlying infrastructure. For a given QoS, the adaptive streaming service selects the adaptation that delivers the best feasible perceptual quality for the video stream.

We presented the results of an implementation and evaluation of a subset of the capabilities of the adaptive video streaming service in a local area laboratory testbed environment. Initial experiments with the prototype implementation of the adaptive video streaming service have clearly demonstrated the potential of this approach. The protocol has been shown to handle a wide range of variations in the QoS delivered by the network and the receiver without causing appreciable degradation in the perceptual quality of the delivered video stream. Although the implementation is successful, how well the design will translate to a wide area environment with large QoS fluctuations remains to be seen. We plan to test the protocol in a WAN environment later this year.

There are several areas of future work. The specification of both the layering structure and the adaptation table, that is used at run time by the adaptive streaming protocol to select the video adaptation most appropriate for a given operation environment, is currently a manual process. Although the VQA tool facilitates this process, it is still a very tedious and challenging task to arrive at an optimal choice given the number of feasible layering alternatives and adaptation tables. We are developing the design of an off-line tool called the Video Application Development (VAD) that would automate some of the difficult aspects of this task. In conjunction with this goal, further research is warranted into 1) enhanced methods for deriving quantitative perceptual quality estimates and additional studies using human subjects to calibrate and validate these methods; 2) techniques for achieving a fully dynamic adaptation table based upon end-user preference for temporal dilation versus spectral versus quantization versus macroblock filtering; and 3) the typical convex nature of the video perceptual quality versus required throughput function and how to select a layering structure which maximizes the convexity of this function.

The focus of this work has been on point-to-point streaming of video. We are developing an approach for extending the adaptive streaming service to a multicast environment where a receiver-selectable mix of resource reservations (using RSVP) and best-effort service can be used to deliver the video stream [12].

References

[1] R. Ramanujan, J. Bonney, C. Manson, J. Mack, and K. Thurber, "Media Transport Service for Video Retrieval Application over ATM Networks," *Proc. 20th Conf. on Local Computer Networks*, October 1995.

[2] A. Eleftheriadis and D. Anastassiou, "Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video," *Proc. 5th Int'l. Wkshp. Network and Op. Sys. Support for Digital Audio and Video*, Durham, NH, November 1995.

[3] L.A. Rowe and B.C. Smith, "A Continuous Media Player," *Proc. 3rd Int'l. Wkshp. Network and Op. Sys. Support for Digital Audio and Video*, San Diego, CA, November 1992.

[4] S. Cen, C. Pu, R. Staehli, C. Cowan, and J. Walpole, "A Distributed Real-Time MPEG Video Audio Player," *Proc. 5th Int'l. Wkshp. Network and Op. Sys. Support for Digital Audio and Video*, Durham, NH, November 1995.

[5] Z. Chen, S. Tan, R. Campbell, and Y. Li, "Real Time Video and Audio in the World Wide Web," *Proc. 4th Int'l. World Wide Web Conf.*, 1995.

[6] A.T. Campbell and G. Coulson, "QoS Adaptive Transports: Delivering Scalable Media to the Desktop," *IEEE Network*, March/April 1997.

[7] N. Yeadon, et. al., Computing Department, Lancaster University, Distributed Multimedia Research Group, "Filtering for Multipeer Communications," www.comp.lancs.ac.uk/computing/users/njy/demo.html

[8] Xing Technologies Corporation, www.xingtech.com.

[9] L.Wang, R. Ramanujan, J. Newhouse, A. Ahamad, M. Kaddoura, K. Thurber, and H.J. Siegel, "An Objective Approach to Assessing Relative Perceptual Quality of MPEG-Encoded Video Sequences," *Proc. Int'l. Conf. on Multimedia Computing and Systems*, Ottawa, Canada, June 1997.

[10] K. Patel and L.A. Rowe, "Design and Performance of the Berkeley Continuous Media Toolkit," *Proc. SPIE Conf. on Multimedia Computing and Networking 1997*, February 1997.

[11] N. Yeadon, F. Garcia, D. Hutchison, and D. Shephard, "Continuous Media Filters for Heterogeneous Internetworking," *Proc. SPIE Conf. on Multimedia Computing and Networking 1996*, March 1996.

[12] R. Ramanujan, A. Ahamad, and K. Thurber, "Traffic Control Mechanism to Support Video Multicast over IP Networks," *Proc. Int'l. Conf. on Multimedia Computing and Systems*, Ottawa, Canada, June 1997.