Continually Traffic Accommodating Internet Streaming Video

**Yon Jun Chung, Hwang Jun Song, Tien Ying Kuo,
Jong Won Kim and C.-C. Jay Kuo**
Integrated Media Systems Center
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, California 90089-2564

Abstract - A new way to implement the Internet modem video transmission is presented in this research. Our system is capable of continually accommodating its bitstream size in response to changing network conditions. The key idea is to adopt an adaptive least mean squares (LMS) controller to orchestrates an H.263+ encoder rate control at the server end as well as a fast frame interpolation at the client end. It is demonstrated that the UDP packet loss is significantly reduced and a smooth video display can be achieved.

Introduction

The number of UDP (User Datagram Protocol) packets lost during transmission is largely a function of network traffic. To cope with lost UDP packets, we should consider the problem from two aspects. The first one is to minimize the number of future packets that will be dropped. The second one is to reduce the effect of reduce load both in terms of dropped packets and decreased video resolution. Most previous efforts have approached this problem from just one of these two angles [3]. Either the application in question could not tailor their transmission stream for one specific receiver, e.g. Internet multicast, or there is no feedback channel for the receiver to signal back the state of network traffic to the transmitter.

Such signaling back is possible for applications such as the Internet videophone, where there is just one receiver for every transmitter. The packet loss minimization and frame rate post-processing can be coordinated jointly between the receiver and transmitter via a feedback channel. In this work, we employ the real-time streaming protocol (RTSP). RTSP has a control connection which handles the command communication between the transmitter (or the server) and the receiver (or the client). We extend the functionality of this control connection to serve as a feedback channel between the server and the client. We present an Internet video transmission that will continually adapt its bitstream size to accommodate the network traffic condition. The presented solution has multiple components: an adaptive LMS (Least Mean Squares) bandwidth controller, real-time H.263+ rate controller and fast motion-compensated frame interpolation (FMCI) all working in unison as presented in the following sections.

RTSP Video with Feedback Channel

There are numerous Internet videophones that employ point to point video transmission available today. Unfortunately, the performance of these products is at the mercy of network traffic. Especially for modem Internet access, their performances range from marginal in low traffic to utter breakdown in heavy traffic. Our demo, as an alternative, must have components capable of continually adapting to changing network conditions and do so in a strict environment.

The task of any BW controller is to instruct the server whether to increase or decrease the bitstream size according to network traffic. The pivotal point for the task is the socket at the client. It essentially acts as our observation portal. The information we relay back to the server must be based on the observations made at this locale. We can observe the size and the transmission time of successfully transmitted real time protocol (RTP) packets. From these two pieces of information, we can gauge the bandwidth the network is capable of sustaining for a particular instance in time. Also by examining the sequence number of arriving packets, we can determine which packets were lost and consequently the packet loss rate.

Some UDP packet, which RTP packets are built upon, loss is accepted as unavoidable. This loss can be reduced to a tolerable level if the client can continuously update the server on the size of bitstream the network can support. A feedback channel controller residing on the client side is proposed to perform this task. Most controllers need a model of the system they are trying to control. However, it is difficult to model the Internet traffic. In this work, we consider a model-free controller borrowing from statistical signal processing. One well-known example is the adaptive LMS controller. The control is performed based only on output observations. The robust nature of the LMS controller makes it a suitable candidate to control a nonlinear and nonstationary system like the Internet. In the current context, our observations are the sequence numbers and interarrival times of packets arriving at the client sockets. The client can relay back any number of data to the server including packet loss rate, delay and available BW. Among these the available BW is the most usable parameter to a real time rate controller. It can use the available BW as a budget constraint in its calculations. We quantify the ratio of successfully arriving packet size over its transmission time as the instantaneous arrival time, denoted as ABW. The formulation for the LMS control can be described as

$$ABW_{k+1} = ABW_k + 2\mu(\rho_{TH} - \rho_k) \cdot \alpha_{pkt}/\tau, \tag{1}$$

where k denotes a time index according to some internal control clock, which has a comparable resolution to the packet transmission clock, ρ_k is the packet loss rate at time k, ρ_{TH} is threshold for the maximal acceptable packet loss rate, α_{pkt} is the size of the last successfully transmitted packet and τ is the inter-arrival time between the last two packets, and μ is the adaptation parameter determined through empirical tuning. Because an overactive feedback channel may erode the data connection bandwidth, ABW updates are not continuously sent. Only when a change in ABW greater than 10% of its previous value is detected, does the server receive an ABW update. For example, if there is a change in the packet loss rate (*i.e.* the packet loss rate ρ_k increases/decreases beyond the acceptable loss rate ρ_{TH}), the controller will enlarge/reduce the amount of the available bandwidth ABW_{k+1} of the next instance. This method was initially tested in simulation with ns (Network Simulator, [1]) with success and the results are shown in [4].

Real-Time Frame Rate Control

The real time rate controller at the server must quickly respond to changes in ABW received from the client with a corresponding adjustment in its bitstream size. The proposed frame rate control is compatible with current macroblock layer rate control. Within this framework, the encoding frame rate control is adopted as a main control mechanism with the macroblock layer rate control as an auxiliary control tool. The encoding frame rate control seeks to tradeoff between spatial with temporal quality to improve perceived video quality.

In this work, we consider a frame layer R-D modeling approach which constructs both the rate and distortion models with respect to the averaged quantization parameter (QP) of all macroblocks in each frame. It can be viewed as a hybrid statistical/experimental method. In terms of mathematics, the rate and distortion models can be written, respectively, as:

$$\begin{aligned}\hat{R}(\bar{q}_i) &= (a\bar{q}_i^{-1} + b\bar{q}_i^{-2})MAD(f_{ref}, f_{cur}),\\ \hat{D}(\bar{q}_i) &= a'\bar{q}_i + b',\end{aligned}$$

where a, b, a' and b' are model coefficients, f_{ref} is the reconstructed reference frame at the previous time instance, f_{cur} is the uncompressed image at the current time instance, $\bar{q}_i$ is the average QP of all macroblocks in a frame, $\hat{R}(\bar{q}_i)$ and $\hat{D}(\bar{q}_i)$ are the rate and distortion models of a frame, respectively, and $MAD(f_{ref}, f_{cur})$ is the mean of absolute difference between f_{ref} and f_{cur}. Note that $MAD(f_{ref}, f_{cur})$ takes into account the dependency among frames. Coefficients a, b, a' and b' are determined by using the linear regression method.

First, we approximate the available Internet channel bandwidth with piecewise constant channel bandwidth. Then, we can estimate the distortion of current frame based on the rate and distortion model. Next, let us consider the rate control scheme. If the spatial quality is below a tolerable level due to fast motion change or sudden channel bandwidth decrease, we should reduce the temporal quality and improve the spatial quality in order to reduce the flickering artifact. At the same time, it is still desirable to control the temporal quality degradation. On the contrary, if the spatial quality is above a certain level, we should increase the temporal quality. Based on the discussion, the encoding frame rate control algorithm can be stated as follows:

- If $\hat{D} > TH_{D1}$, increase the encoding frame interval by $\Delta F^{int}(F^{int}_{cur})$.
- If $\hat{D} < TH_{D2}$ and the current frame interval is greater than the frame capturing interval, the encoding frame interval is decreased by $\Delta F^{int}(F^{int}_{cur})$.

TH_{D1} and TH_{D2} are two threshold values to be selected. By adopting this rate control scheme, we can avoid the abrupt change of the encoding frame rate and improve the spatial quality. This algorithm can be applied in real-time processing since the computational complexity is very low and low latency can be guaranteed.

Fast Motion-Compensated Frame Interpolation

The jerkiness accompanying low frame rate playback is an annoying artifact. However, the proposed rate controller in the H.263+ standard has no recourse but to vary the frame rate from time to time. Maintaining a high frame rate via interpolation is an excellent way to remove the jerkiness. This can be thought of as temporal video post processing. Recourse is taken at the client side in the form of a frame interpolation algorithm, a deformable block based fast motion-compensated frame interpolation (DB-FMCI). Regardless of the variable frame rate received, a constant playback rate is maintained with a minimal computational complexity.

The complexity of DB-FMCI is reduced since no additional motion search in the decoder is needed as required by standard MCI. It has been observed from experimental results that the visual quality of coded low-bit-rate video is significantly improved at the expense of a small increase in decoder's complexity. The proposed DB-FMCI is implemented in the decoder as a video post-processing unit, which is cascaded with the standard H.263/H.263+ decoder without changing the bitstream syntax. DB-FMCI consists of three main units, i.e. motion-preprocessing, segmentation and MCI prediction. The motion-preprocessing unit is used to modify the block-based motion field to achieve a better frame interpolation result. Once the post-processed motion field is obtained, we map it to the pixel-based motion field for MCI prediction. The adopted mapping strategy is the deformable block mapping, that is, the block mapping from the decoded frame is deformed with an affine transform. The second unit of FMCI performs object segmentation of decoded frames, which is useful to provide the moving object location for MCI. We do not use any complicated segmentation procedure, partly because we do not want to increase the computational load in the decoder and partly because the segmentation result is rough due to the use of the block-based motion field

Method	Avg PSNR	STD of PSNR	NO of Enc. frms
TMN8(2-frame skip)	31.2348	1.3278	132
TMN8(1-frame skip)	30.3696	1.2894	197
Proposed method	30.8446	1.0200	167

Table 1: Performance comparison with TMN8.

only. For the third unit, classification of regions into stationary, covered and uncovered backgrounds and the moving object, which are used in standard MCI, is still adopted here.

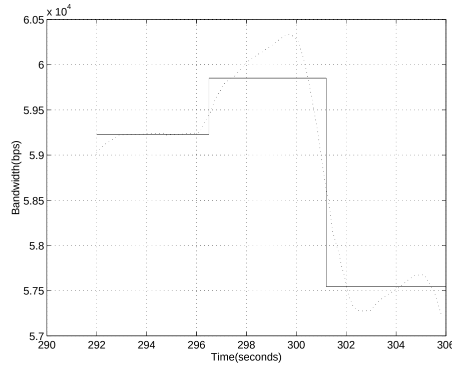
Even though H.263/H.263+ provides the optional PB-frame mode to achieve a similar goal of frame interpolation, the PB-frame still requires extra bits to encode the B frame overhead and the optional B-frame motion vector (MVDB). Besides, the PB-frame mode can only interpolate one B frame between two P frames. The proposed FMCI scheme is capable of inserting as many frames as needed at any time reference. Furthermore, they require no extra bits and conform to the bitstream syntax of the H.263/H.263+ standard.

Experimental Results

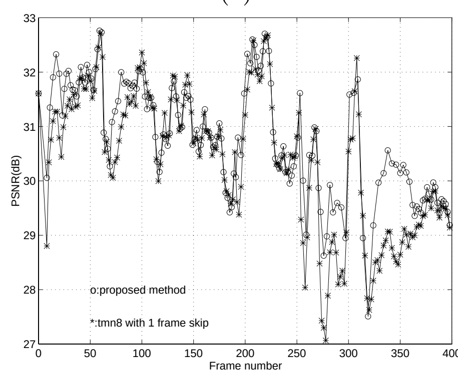
These experimental results were performed with not only talking heads but with footage containing fast motion. The adaptive LMS controller relays to the real time rate controller at the server the available Internet channel bandwidth and its approximation shown in Fig. 1(a). The proposed real time controller employs the channel bandwidth approximation as a bitstream budget and performs the necessary rate adjustment. Under the approximated channel bandwidth, we can see the performance comparison with TMN8 in Table 1. The proposed frame rate control algorithm increases the average PSNR and reduces the standard deviation of PSNR. Furthermore, the degradation in motion smoothness caused by encoding in variable frame rate is not made obvious since the encoded frame interval changes very gradually as shown in Fig. 1(b).

Conclusions

The demonstration presented in this paper was conceived as a total solution to the problems associated with VBR video transmission over a non-centralized, loss prone, single packet class network such as the Internet. The adaptive LMS BW controller instructs the packet loss minimizing BW consumption level to the server. The real time rate controller at the encoder/server responds by tailoring the bitstream to fit the instructed available BW. Last, DB-FMCI at the client side, restores the frame rate reduced albeit intact video bitstream, via interpolation.



(a)



(b)

Figure 1: (a) Available channel bandwidth and its approximation and (b)PSNR comparison with TMN8.

References

[1] http://www-mash.cs.berkeley.edu/ns

[2] Tien-ying Kuo and C.-C. Jay Kuo, "Motion-Compensated Interpolation for Low-Bit-Rate Video Quality Enhancement", SPIE's 44th Annual Meething, July. 1998.

[3] Y. Chung, J. Kuo, "Non-Disruptive RTSP Video Over the Internet using a Modem Connection" , *SPIE Visual Communication and Image Processing*, San Jose, January 1998.

[4] Y. Chung, J. Kuo, "Network Friendly Video Streaming via Adaptive LMS Bandwidth Control" , *SPIE Proceeding of International Symposium on Optical Science, Engineering, and Instrumentation*, San Diego, CA, July 1998.