An End-to-End Video Transmission Framework with Efficient Bandwidth Utilization

Hongli Luo[1], Mei-Ling Shyu[1], Shu-Ching Chen[2]
[1]*Department of Electrical and Computer Engineering*
University of Miami, Coral Gables, FL 33124, USA
h.luo@umiami.edu, shyu@miami.edu
[2]*Distributed Multimedia Information System Laboratory, School of Computer Science*
Florida International University, Miami, FL 33199, USA
chens@cs.fiu.edu

Abstract

We present a new framework for streaming video over Internet in this paper. The sending rate is dynamically adjusted to obtain a maximal utilization of the client buffer and minimal bandwidth allocation. To guarantee a more reliable and better quality of delivery of the video frames, retransmission and selective drop of different frame packets are integrated into our framework. Under severe network congestion, an adaptive playback schedule can provide relative high video quality at a low frame rate. Comparisons are made with the most current streaming approaches using the H.26L video coder to evaluate the performance of the framework. Simulations results show that PSNR is increased in our approach, which provides a better quality of the decoded frames, and the quality of the decoded frames also changes more smoothly.

1. Introduction

Video transmission over the Internet has been a challenging task. To guarantee the presentation quality of multimedia applications, QoS requirements such as packet loss rate, bandwidth, delay, and jitter must be satisfied. Normally, the requirements of video transmission include high bandwidth and low delay, while a certain amount of packet loss can be tolerated.

In network congestion, besides decreasing the transmission rate, scalable video coding is also used. The scalability of video consists of SNR scalability, spatial scalability, and temporal scalability [7]. Hence, how to achieve a satisfactory frame quality with the limit of the bandwidth involves the consideration of the importance of different packets. Different schemes are now available, such as forward error correction (FEC), retransmission, and selective drop [7]. Since a higher quality frame with a lower frame rate is more acceptable to the users than a lower quality frame with a higher frame rate, in our approach, we adopt the temporal scalable video playback technique.

H.26L [2] video coder has group-of-pictures (GOPs) similar to MPEG, where each GOP starts with an I-frame, followed by the interleaving P-frames and B-frames, which enables the decoding of any GOP without the decoding of the previous GOPs. I-frames are intraframe coded without a reference to other frames. P-frames are coded using motion compensation prediction based on the previous I-frames and P-frames. B-frames are coded using bi-directional prediction or interpolation from the past and future I-frames and P-frames. Motion estimation is done on the macroblocks in the size of 16 x 16 pixels. Thus the encoder generates packets of different importance for decoding. The transmission of the packets should be guaranteed according to their priorities.

The playback buffer at the receiver stores the video packets before they are used for decoding. Maintaining a little playback delay also allows the retransmission of the lost frame packets before the decoding deadline [3]. The buffer can be used to smooth the video stream and reduce the jitter introduced by the changing network delays. The fully utilization of the limited buffer at the receiver is especially helpful for mobile terminals such as PDA and cellular phone.

In our previous work, an end-to-end optimal rate control scheme was proposed in [4][5], which aims at achieving maximal utilization of the client buffer and at the same time minimal bandwidth allocation. The optimized rate is obtained based on the simple feedback information sent from the client. In this paper, we present an end-to-end video transmission framework by integrating our rate control scheme with I-frame packet retransmission, selective B-frame packet dropping, and adaptive playback rate adjustment algorithms. To provide a more acceptable video quality when packet loss ratio is too high under severe congestion, an adaptive playback schedule will generate a relatively high quality video sequence with a reduced frame rate.

The paper is organized as follows. Section 2 gives the entire video communication system, including the coder, optimal bandwidth allocation scheme, retransmission and selective dropping. In Section 3, we present the simulation scenario and simulation results. Conclusions are given in Section 4.

2. The proposed framework

A brief introduction of the optimal rate algorithm [4][5] is given below. Assume an end-to-end connection between the server and the client and let Q_r be the allocated buffer size for each client at the setup of the connection. At each time interval k, let R_k be the number of packets transmitted from the server, P_k be the number of packets arriving at the client buffer, and L_k be the number of packets used for playback. In addition, let Q_k and Q_{k+1} denote the numbers of packets in the client buffer at the beginning of time intervals k and k+1, respectively. Hence,

$$Q_{k+1} = Q_k + P_k - L_k \tag{1}$$

The packets arriving at the client buffer at k comprise of those packets transmitted from the server at time intervals k-d, k-d-1, …, and k-d-i+1. $b_{i,k}$ is used to denote the corresponding percentage of the packets transmitted at k-d-i+1 that arrive at the client buffer at k. Thus we have

$$P_k = b_{1,k} R_{k-d} + b_{2,k} R_{k-d-1} + \ldots + b_{i,k} R_{k-d-i+1} \tag{2}$$

where the subscript k-d is to denote the closest time interval when the transmitted packet can arrive at the buffer, and k-d-i+1 denotes the farthest time interval when the transmitted packet can arrive at the buffer at time interval k. Similar to $b_{i,k}$, the value of d can also be obtained at the client by checking the timestamp of the arriving packets.

To make optimal utilization of the network resource, we try to keep the transmission rate small, and at the same time keep the difference between the total size of the packets in the buffer and the allocated buffer size small. The optimization object is to find a suitable sequence of R_k that can minimize the following quadratic performance function:

$$J_k = (w_p Q_{k+d_0} - w_q Q_r)^2 + (w_r R_k)^2 \tag{3}$$

where w_p, w_q, and w_r are the weighting coefficients. Because of the network delays, the change of transmission rate at k (R_k) will result in a change of the packets in the client buffer at k+d_0 (Q_{k+d0}). Therefore, d_0 is introduced in the performance index as a transmission control delay parameter. Here, we have $d_0 = d + 1$.

Using time domain representation to combine Equations (1) and (2), we have the following equation:

$$A(z^{-1})Q_k = z^{-d_0} B(z^{-1}) R_k - L_k, \tag{4}$$

where $A(z^{-1}) = 1 - z^{-1}$ and

$$B(z^{-1}) = b_{1,k} + b_{2,k} z^{-1} + \ldots + b_{m+1,k} z^{-m} \tag{5}$$

z^{-1} is the delay operator (i.e., $z^{-1}Q_k = Q_{k-1}$). Here we assume that $b_{1,k} \neq 0$, and d_0 is decided during the computation. When 1 is divided by $A(z^{-1})$, we get the quotient and remainder as follows.

$$1 = A(z^{-1})F(z^{-1}) + z^{-d_0} G(z^{-1}) \tag{6}$$

The optimal transmission rate R_k is as follows.

$$\begin{aligned}&(w_p^2 B(z^{-1})F(z^{-1}) + \frac{1}{b_{1,k}} w_r^2) R_k \\ &= -w_p^2 G(z^{-1}) Q_k + w_p w_q Q_r + w_p^2 F(z^{-1}) L_{k+d_0}\end{aligned} \tag{7}$$

In response to the network congestion, w_r will be increased and produce a decreased transmission rate. In a less congested situation, w_r will gradually decreased to the original values. The client regularly sends the feedback to the server. The server can collect the information about buffer occupancy, packet loss ratio, playback schedule, and playback rate. Then it can calculate the transmission rate.

The congestion control of our rate control scheme can be used to alleviate the burst packet loss in the congestion. To improve the quality of the reconstructed frames, when some particular packet is lost, the following transmission policies such as the retransmission of the I-frame packet, the selective drop of the B-frame packet, and adaptive playback rate adjustment are applied. However, after the first two policies are applied, there will still be some packet losses, which will be handled by the error concealment of the H.26L decoder.

I-frame packet retransmission: I-frames have the most contribution to the decoded video frames. The lost of an I-frame packet will severely degrade the quality of the decoded video frames. Since UDP does not guarantee the reliable delivery of the packets, a retransmission algorithm is adopted. Each time a loss of an I-frame is detected, an acknowledgement is sent back to the source to request for the retransmission of the lost I-frame packet.

Selective B-frame packet drop: During severe network congestion, it is recommended to reduce the transmission rate for congestion control. To meet the deadline of the playback schedule of each GOP, the B-frame packets are randomly dropped to decrease the requirement of the bandwidth.

Adaptive playback rate: When the network is too congested and the client buffer is underflowed because of the bandwidth limitation, a playback rate adjustment is adopted where the video is decoded at a lower frame rate, which is the half of the normal playback rate. When the congestion is alleviated and the client can buffer enough packets for the next GOP, it is switched back to the normal playback

3. Simulation results

3.1. Simulation environments

In this paper, we aim at evaluating the performance of our framework under a certain coder instead of examining the performance of the coder. The simulations are conducted using NS-2. A typical single bottleneck is used with the typical dumb-bell configuration. In this commonly used network topology, congestion only occurs in the link connecting two routers. The bandwidth of the link is the bottleneck bandwidth. The background traffic consists of TCP-based connections. Simulations are run under different bottleneck bandwidth with 5 flows to test the performance of our framework under different network congestion scenarios and packet loss ratios.

Table 1. Simulation Parameters

Parameter	**Value**
Average Bit Rate	1.15 Mbps
Buffer Capacity	500K Byte
Prefetch Size	5 GOPs
w_r range	[1, 4]
w_q	4
w_p	4
Number of Flows	5
Feedback Interval	16/30 seconds

We use the standard test video sequences (Tempete [6]) in the simulations. This is a high-bit rate video and the video is CIF resolution (352 x 288 pixels) at 30 fps. The video sequences are in the YUV 4:2:0 format and is encoded using H.26L encoder. The GOP consists of 16 frames in the order of IBBPBBPBBPBBPBBP. 256 frames are encoded into 16 GOPs, and the average bit rate is 1.15Mbps. 5 GOPs are prefetched before the playback begins, and so the initial playback delay is approximately 2.5 seconds. The feedback interval is 16/30 seconds in our simulation. At the end of the playback of a GOP, the feedback information is sent back to the source. Table 1 gives the parameters used in the simulation.

3.2. Comparisons of PSNR

To evaluate the performance of our approach, we compare our proposed framework (denoted as Approach A) with the integrated end-end buffer management and congestion control approach proposed in [1] (denoted as Approach B) in terms of PSNR under the same network environment. PSNR is defined as the Signal to Noise Ratio in dB's.

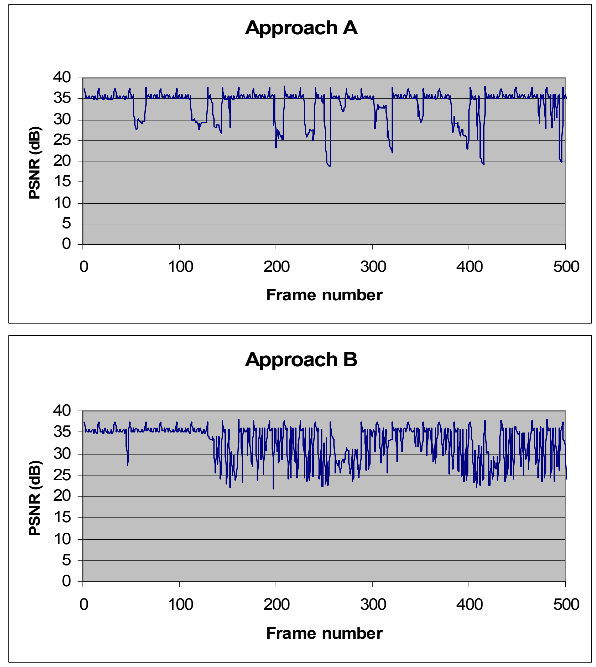


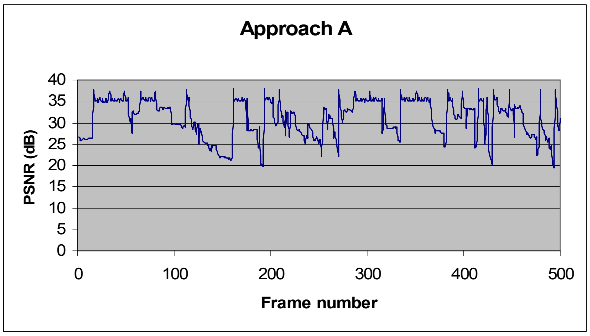
Figure 1. PSNR comparison of video sequences under the bottleneck bandwidth of 5Mbps.

Comparisons are made under different network congestions with different bottleneck bandwidths (i.e., 6 Mbps, 5 Mbps, and 4.5 Mbps). The comparison of PSNR is displayed in Figure 1 with the bottleneck bandwidth of 5Mbps. It contains 500 frames, which lasts about 16 seconds. It can be seen from the figure, under the same network conditions, our approach can get better PSNR values than those of Approach B. Another scenario with a larger packet loss ratio is compared in Figure 2 with the bandwidth of 4.5 Mbps.

Table 2. Comparison of PSNR in Different Scenarios

Bottleneck Bandwidth	PSNR (dB) Approach A	PSNR (dB) Approach B
6Mbps	35.04	34.21
5 Mbps	33.53	32.33
4.5 Mbps	31.01	29.33

Table 2 gives the average PSNR of the video frames under different scenarios. Compared with Approach B, the average PSNR gain is more than 1 dB. We have a larger PSNR gain when the network is more congested. Since Approach B with its buffer management has a 6 dB's gain over the approach without buffer management, we can claim that our approach has an obvious performance improvement. To illustrate how the PSNR changes in the video sequence, we give a more detailed comparison in Figure 3, which displays the PSNR from frame 300 to 400. Besides the average PSNR value, we also need to examine the variability of the displayed frames, especially the neighboring ones. Figure 3 shows that the PSNR changes more smoothly and less frequently in Approach A. It gives a better presentation effect to the users. Hence, another advantage of our proposed framework is that the presentation quality of the decoded frames is more smoothly.



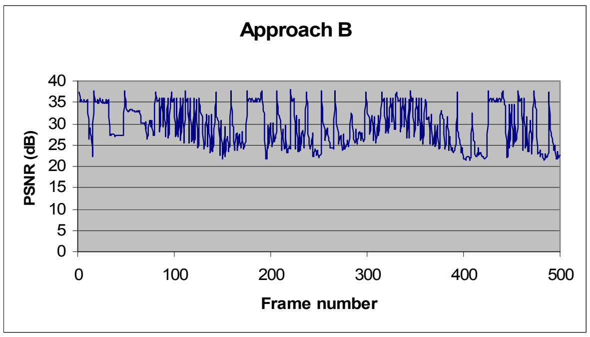
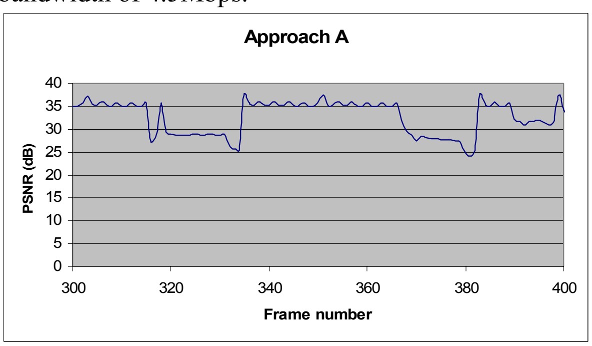


Figure 2. PSNR comparison under the bottleneck bandwidth of 4.5Mbps.



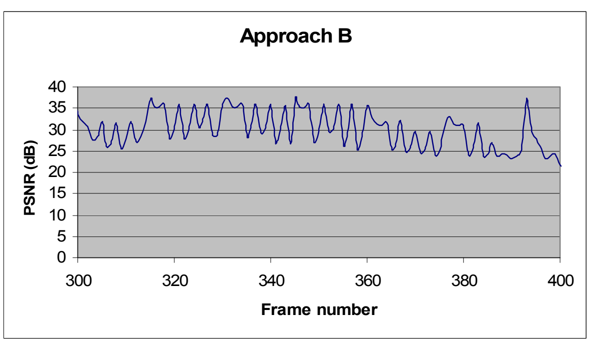


Figure 3. A more detailed view (Frames 300 – 400) for the PSNR comparison in Figure 1.

4. Conclusions

In this paper, an end-to-end video transmission framework is proposed. An optimal transmission rate is obtained to maximize the utilization of the client buffer with a minimal bandwidth allocation. The retransmission of the I-frames, selective drop of the B-frames, and adaptive playback rate adjustment are incorporated in our framework to improve the video quality under the constraints of limited bandwidth and playback deadline. When buffer underflow occurs because of severe network congestion, the selective dropping and playback schedule adjustment can effectively alleviate the degradation of the displayed video frames. Our proposed framework can achieve larger PSNR improvement when the network is more congested, and the quality of the decoded frames changes more smoothly, which is more favorable to the users.

5. Acknowledgement

For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260. For Shu-Ching Chen, this research was supported in part by NSF EIA-0220562, NSF HRD-0317692, and the office of the Provost/FIU Foundation.

6. References

[1] I. V. Bajic, O. Tickoo, A. S. Kalyanaraman, and J. W. Woods, "Integrated End-to-End Buffer Management and Congestion Control for Scalable Video Communications," *IEEE ICIP'03*, Barcelona, Spain, September 2003.

[2] Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, "Joint Model number 1, revision 1 (JM-1R1)," JVT-A003rl, January 2002.

[3] D. Loguinov and H. Radha. "On Retransmissions Schemes for Real-time Streaming in the Internet," *Proceedings of Infocom*, pp. 1310 – 1319, Anchorage, AL, May 2001.

[4] M.-L. Shyu, S.-C. Chen, and H. Luo, "Optimal Bandwidth Allocation Scheme with Delay Awareness in Multimedia Transmission," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2002)*, pp. 537-540, August 26-29, 2002, Lausanne, Switzerland.

[5] M.-L. Shyu, S.-C. Chen, and H. Luo, "End-to-End Congestion Controll via Optimal Bandwidth Allocation for Multimedia Streams," *Proceedings of the 15th International Conference on Computer Applications in Industry and Engineering*, pp. 57-60, November 7-9, 2002, San Diego, California USA.

[6] http://kbs.cs.tu-berlin.de/~stewe/vceg/sequences.htm (available on December 15, 2003).

[7] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting Real-time Video over the Internet: Challenges and Approaches," *Proceedings of the IEEE*, vol. 88, no. 12, Dec. 2000.