A MULTI-BUFFER SCHEDULING SCHEME FOR VIDEO STREAMING

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

In this paper, we propose a multi-buffer scheduling scheme for streaming video systems. A transmission rate is obtained via a rate control algorithm, which optimally utilizes the network bandwidth and client buffer resources. The server side maintains multiple buffers for packets of different importance levels. It schedules the transmission of each packet based on the source buffer size and playback deadline to reduce the end-to-end distortion. The performance of proposed scheme is evaluated in terms of peak signal-to-noise ratio (PSNR) in the simulations, and the simulation results demonstrate the improvement of the average PSNR values compared with two other scheduling schemes.

1. INTRODUCTION

Streaming video has the bandwidth requirement in order to provide a good quality of service. When the transmission rate is limited by congestion control and cannot satisfy the bandwidth requirement of the original presentation, adaptive quality streaming should be provided. Among many approaches for adaptive quality streaming, selective dropping at the source side can provide a smooth degraded video quality under the rate constraint. There have been several selective dropping schemes proposed [1][2][3][6][7]. In [2], it addresses the problem of streaming packetized media in a rate distortion optimized way and gives a rigorous analysis of it. The scheduling algorithm decides which packets will be transmitted under the rate constraint while minimize the end-to-end distortion. [3] models the streaming system as a queuing system. An optimal substream is selected based on the decoding failure probability of the frame and the effective network bandwidth. A probability dropping mechanism is proposed in [1] to calculate the dropping probability for each layer. [6] presents a streaming framework centered around the concept of priority drop. It combines the scalable compression and adaptive streaming to provide a graceful degradation of the quality. Most of these approaches work in a rate-distortion optimized way which satisfies a rate constraint. The rate constraint is obtained based on the estimation of the available network bandwidth.

Compressed video consists of packets with different levels of importance and the packets have different impacts on the presentation quality of the decoded videos. Treating all of the packets with equal importance usually results in severe quality degradation during packet losses in heavy congestion. A rate-distortion optimized streaming works on the allocation of bandwidth resources between different packets and in a way that minimizes the reconstruction distortion of the presentation at the clients. Group-of-pictures (GOPs) of H.26L [4] video codec consists of I, P, and B frames. The packets of different frames have dependency, thus the encoder generates packets of different priorities and importance levels for decoding. I, P, and B frames are of different importance levels. Within an importance level, the packets appear earlier in the frame have higher priorities.

In this paper, we propose a packet scheduling scheme to provide adaptive quality for video streaming. We presented a rate control algorithm in [5] for real-time best-effort streaming, which allocates a minimal bandwidth to satisfy the playback requirement. Packet scheduling addresses the problem of how to provide a better and smoother video quality under a limited rate. The contribution of our work is that the proposed approach tries to transmit the more important packets subject to the rate constraint obtained from the rate control algorithm via a multi-buffer scheme at the source. The packet scheduling works in conjunction with rate control, but in a different scale. The rate control algorithm itself works in a larger time scale, allocating the bandwidth to each GOP; while in this study, the scheduling works within each GOP, allocating resources between packets inside a GOP.

So packet scheduling can provide a refined resource allocation to provide an adaptive quality.

The paper is organized as follows. Section 2 gives the proposed multi-buffer scheduling scheme. In Section 3, we present the simulation results and the comparison with different schemes. Conclusions are given in Section 4.

2. MULLTI-BUFFER SCHEDULING SCHEME

Packet scheduling decides which packets to send and at what rate. If the packet arrives after the deadline, it will be discarded, which will lead to a degradation of the presentation quality and a waste of network bandwidth. It should send the packet of a higher importance, and thus maximize the presentation quality of the decoded video. The proposed multi-buffer scheduling scheme is composed of two components: rate control [5] and packet scheduling. The rate control component adopted our previously proposed rate control algorithm which decides a suitable transmission rate by considering the network congestion, playback requirement, and client buffer occupancy. Subject to this rate constraint, the packet scheduling component decides which packets to send in order to achieve a better presentation quality at the client. Some packets must be selectively dropped and not transmitted when the original playback requirement cannot be satisfied by the network bandwidth.

First, we briefly review our previously proposed optimized rate scheme. Let Q_r be the allocated buffer size for each client. At time interval k, R_k is the number of packets transmitted from the server, P_k is the number of packets arriving at the client buffer, and L_k is the number of packets used for playback. 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.

$$Q_{k+1} = Q_k + P_k - L_k . \quad (1)$$

To make the optimal utilization of the client buffer and network bandwidth, while satisfying the playback schedule, we try to minimize the following J_k function,

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

where w_p, w_q, and w_r are the weighting coefficients. Because of the network delays, the change of transmission rate R_k will result in a change of the packets in the client buffer Q_{k+d0}, where $d_0 \geq 1$ indicates the delay. The transmission rate is adjusted periodically based on the feedback information provided by the clients. Some modification during the calculation is that the scaled playback rate is used here instead of the original playback rate.

Our rate control algorithm allocates the bandwidth for each GOP while trying to obtain an optimal utilization of the network resources. For each GOP, it needs to schedule which packets to be sent out in a distortion-optimized way. It can also be viewed as a refined resource allocation scheme, which allocates the bandwidth among packets within each GOP, while the rate control is decided among different GOPs.

The source buffer is used to hold those packets that will be sent out during the next time interval. The source buffer actually consists of multiple buffers, each of such buffer works like a FIFO queue with different importance levels or prority. The importance levels and the concept of layers are interchangeable. The definition of the level here can be very generic. For example, we can have 3 queues for I, P, and B packets separately, with the queue for I packets has the highest priority. In this manner, we have 3 importance levels. It can also work in a more refined way. For a specific GOP encoded as IPBBPBBPBBPBBPBB, we have 1 + 2*5 = 11 levels. Hence, for a coarse quality control, the level of 3 can be considered. On the other hand, for a more refined quality control, 11 levels can be used. For layered video, different layers can directly be used as importance levels.

For a packet, the delay it may experience before it is decoded and played at the client is denoted as *delay(t)*.

$$delay(t) = \frac{B(t)}{R_c(t)} + s\_rtt + k(t) * \mathrm{F} , \quad (3)$$

where $B(t)$ is the total source buffer occupancy, $R_c(t)$ is the transmission rate obtained from the rate control algorithm, s_rtt is the current total delay from the server to the clients, $k(t)$ is the current number of GOP in the client buffer that waits to be decoded, and F is the playback duration for one GOP.

Define $B(t)$ to be the total buffer occupancy at the source side, since the source buffer occupancy is composed of multiple buffers for different levels.

$$B(t) = \sum_{i=0}^{M} B_i(t) , \quad (4)$$

where $B_i(t)$ is the separate buffer occupancy for layer i, and M is the total number of importance levels.

We need to estimate the current single trip time,

$$s\_rtt = \alpha * s\_rtt + (1-\alpha) s\_rtt_r \quad (5)$$

where $0 < \alpha < 1$, s_rtt_r is the most recent s_rtt obtained from the feedback report. The number of GOPs in the client buffer is also available from the feedback. Since we use the original playback schedule, which is fixed, it is easy to calculate whether the packet can meet the deadline or not. If not, this packet and the subsequent packets in the same buffer will be discarded. This scheduling works from the lowest level buffer to the higher level buffer.

3. SIMULATION RESULTS AND ANALYSIS

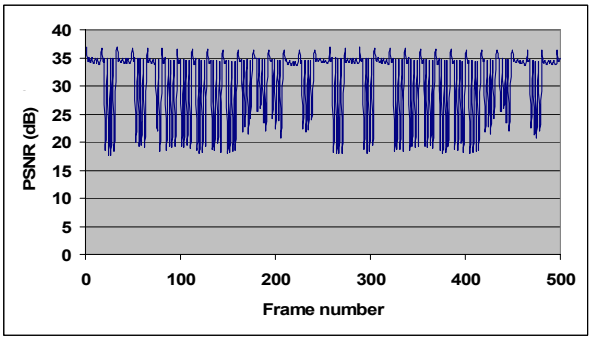
The simulations are conducted on the standard video test sequences, Mobile & Calendar, coded using H.26L. These video sequences are in the format of 4:2:0 with 352 x 288 pixels per frame and 30 frames per second. The average

bit rate of this video clip is 1.58 Mbps. A total of 256 frames are encoded into 16 GOPs. Each GOP consists of 16 frames in the order of IPBBPBBPBBPBBPBB. The 16 GOPs are repeatedly sent from the server. At the client side, the video sequence is played at a fixed schedule at 30 frames per second. The video sequences are transmitted over the simulated environment in NS2. A single bottleneck is used as the network configuration where congestion only occurs in the link connecting two routers, thus the link bandwidth is the bottleneck bandwidth. The background traffic is generated by ftp connections to produce the competing TCP flows.

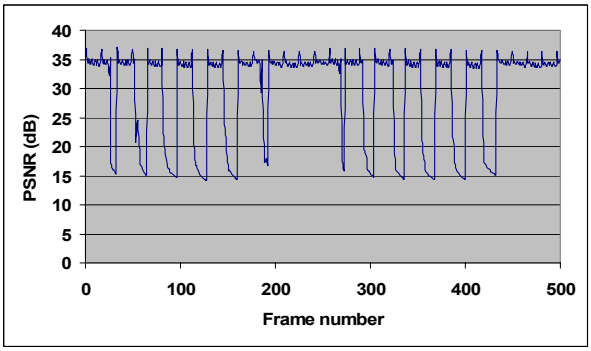
To illustrate the advantages of our approach, we compare it with two other different selective drop mechanisms. In the first approach, the rate is decided based on our previous rate control scheme. The packets are dropped at the source using a conventional scheme which depends solely on the playback deadline. The source drops the packets of the remainder when a packet is found to be able to miss the playback deadline. It is denoted as the "deadline based" approach. The second approach, the packets of each layer is dropped randomly with some probability under available network bandwidth. The probability is corresponding to the packet size of each layer. It is denoted as the "probability drop" approach. In all of the three approaches, the clients start the playback after it has buffered 6 GOP packets, which is approximately 3 seconds waiting for prefetch.

In the first simulation, we examine the situation when the bottleneck bandwidth is 10 Mbps. With 8 multimedia flows with an average bit rate of 1.58 Mbps and the background TCP competing for the 10 Mbps, the source dropping is unavoidable. The PSNR values of one client for each approach under the bottleneck bandwidth of 10 Mbps are displayed in Figure 1. From the figure, we can see how the PSNR changes in each GOP under different approaches. Generally speaking, our proposed approach has a better performance over the other two schemes. In the "deadline based" approach, at the beginning of some GOP, a high PSNR can be obtained, but often followed by some drastic degradation of the PSNR to even 15 dB. This is because some P frame packets may be dropped because they can not meet the deadline. In the "probability drop" scheme, followed by a high PSNR for the first several frames in a GOP, sometimes there are continuous low PSNR frames. In this scheme, P and B frames are randomly dropped with some probability, and so the dropping of a P frame is possible. It is also possible some B frame packets that appear earlier with a higher priority in the GOP are dropped, while B frame packets appear later with a lower priority are kept. That is why a larger distortion of the quality is resulted from this scheme.

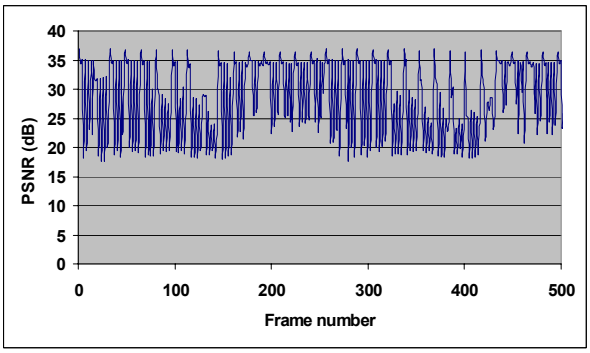
To give a better understanding of how different packet dropping schemes will affect the presentation quality, a detailed PSNR view for Figure 1 is given in Figure 2. The PSNR values of 3 GOPs (GOP 7 – GOP 9) are plotted. In the 48 frames, frames 1, 17, and 33 are the leading frames for the 3 GOP. Since all I frames are sent, the leading frame will have a high PSNR. The exception is when an I packet is lost during the transmission because of the network congestion. The dropping of the PSNR of the leading frame of the third GOP in probability dropping is resulted from this kind of I frame packet loss in the network. Compared with the proposed approach, the "probability drop" approach has a lower PSNR in the subsequent frames in each GOP, or with the peak value decreasing. The continuous low values of the second GOP in the "deadline based" approach are resulted from some earlier P frame packets dropping, which lead to a severe degradation of the whole GOP.



(a) Proposed Approach

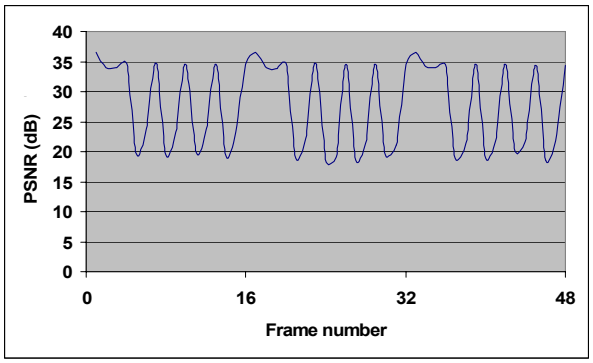


(b) Deadline Based

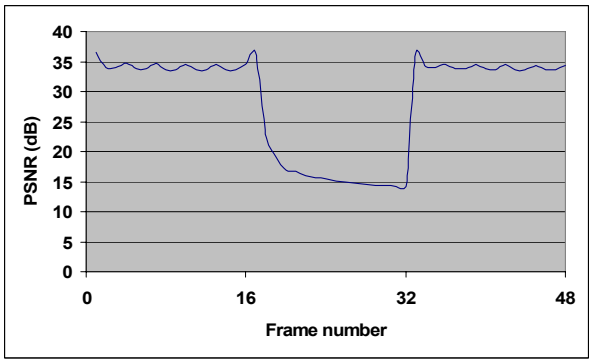


(c) Probability Drop

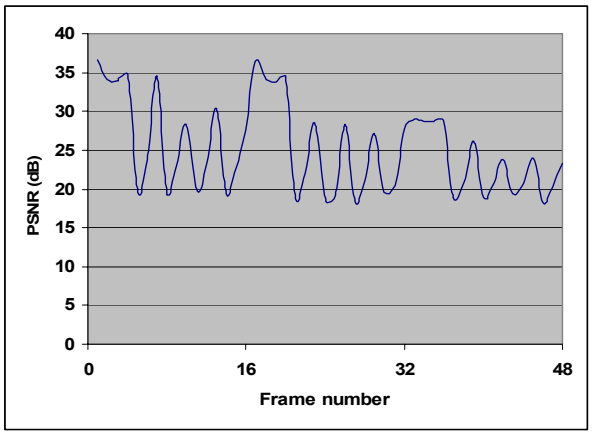
Figure 1. PSNR comparison of frames under bottleneck bandwidth of 10 Mbps with 8 flows.



(a) Proposed Approach



(b) Deadline Based



(c) Probability Drop

Figure 2. A more detailed PSNR view (Frames of GOP 7 to GOP 9) for Figure 1.

The simulations are also run under different bottleneck bandwidth to illustrate the performance of our approach under different selective source dropping ratios. The resulting PSNR values are averaged over all 8 flows to give a fair evaluation of the performance. The average PSNR values under different bottleneck bandwidth are displayed in Table 1. From this table, it can be easily seen that our proposed scheme outperforms the other two schemes under all bandwidth limitations. The advantage of the proposed scheme is obvious under different congested bottleneck bandwidths in terms of PSNR. The "deadline based" approach performs better than the "probability drop" approach since it adopts our rate control algorithm, which considers the playback requirement, network congestion, and buffer occupancy together. The "probability drop" approach has the lowest PSNR values, because it does not consider the interdependences of the packets at the same level. Under the same network situation, if more important packets are dropped, it will have a larger distortion of the quality.

Table 1. Average PSNR values (dB) for 3 approaches under different bottleneck bandwidth

Bottleneck Bandwidth	Proposed Approach	Deadline Based	Probability Drop
12 Mbps	33.20	32.88	31.52
11 Mbps	31.74	31.33	30.29
10 Mbps	30.04	29.30	28.17

4. CONCLUSIONS

In this paper, we present a multi-buffer packet scheduling scheme for video streaming. This scheme schedules the transmission of the packets based on an optimal rate control algorithm. Multiple buffers for different importance levels are applied at the source, via which the scheduling scheme differentiates packets with different priorities. In the simulations, we compare it with two other approaches, and the simulation results have shown that the proposed multi-buffer scheduling scheme outperforms the other two approaches and can improve the PSNR of the transmitted video.

5. 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] P.A. Chou and Z. Miao, "Rate-distortion Optimized Streaming of Packetized Media," Technical Report, MSR-TR-2001-35, Microsoft Research, February 2001.

[3] S.H. Kang and A. Zakhor, "Effective Bandwidth based Scheduling for Streaming Multimedia," *IEEE ICIP'03*, Barcelona, Spain, September 2003.

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

[5] H. Luo, M.-L. Shyu, and S.-C. Chen, "An End-to-End Video Transmission Framework with Efficient Bandwidth Utilization," *IEEE International Conference on Multimedia and Expo (ICME 2004)*, Taipei, Taiwan, R.O.C., June 27-30, 2004.

[6] C. Krasic, J. Walpole, and W. Feng, "Quality-Adaptive Media Streaming by Priority Drop," *NOSSDAV 2003*, June 2003.

[7] P. Cuetos, and K. Ross, "Optimal Streaming of Layered Video: Joint Scheduling and Error Concealment," *ACM Multimedia*, pp. 55-64, 2003.