Smooth Playout Control for Video Streaming over Error-Prone Channels

Yi-Hsuan Yang, Meng-Ting Lu, and Homer H. Chen
Graduate Institute of Communication Engineering, National Taiwan University
b91901109@ntu.edu.tw, b88901124@ntu.edu.tw, homer@cc.ee.ntu.edu.tw

Abstract

The quality of media streaming over best-effort networks suffers from network delays and packet losses. The latter is more profound for wireless video. To enhance the QoS of streaming services, adaptive media playout (AMP) has been developed to adjust the playout interval. With AMP, the risk of delay and buffer underflow is reduced. However, the smoothness of playback is not guaranteed. In this paper, we propose a novel AMP control that enables smooth playout and meanwhile maintains reliable visual quality. Our AMP control adjusts the playout interval based on an estimation of channel quality, so it is more adaptive than conventional AMP controls that are based on buffer fullness. Experimental results are provided to justify our approach. Even at 20% packet loss rate, the proposed AMP control is still able to provide smooth and reliable playback.

1. Introduction

With the streaming technique, a video sequence is sent in the form of a continuous stream and is played as it arrives. Therefore, a viewer need not wait for the file to be completely downloaded before seeing the video. Real-time video services and internet conferences can be realized by using this technique.

However, under the prevailing best-effort network structure, the quality of streaming services suffers severely from network delays and packet losses. Generally, media data are buffered at the client to reduce delay jitter. However, at high packet loss rate, playback interruptions may occur due to buffer underflow, resulting in visual quality degradation.

Delay reduction techniques for audio playout have been studied before [1], [2]. These techniques concentrate on the minimization of playout latency, which is critical for audio streaming. Comparatively, variation in the video playout interval is subjectively less annoying than playout interruptions and long delays [3], despite variations up to 25% are noticeable. Adaptive media playout (AMP) is such a technique that utilizes this property to maintain the QoS of media streaming services.

Conventional AMP controls adjust the playout interval to compensate the fluctuation in the receiving buffer. When the buffer fullness is low, the playout interval is decreased to avoid buffer underflow. However, a slower playout interval increases the viewing latency. On the contrary, when the buffer fullness is high, playout interval is increased to reduce delays. So the key to AMP lies in the trade-off between buffer underflow and playout delay. A study of the relationship between channel parameters, buffer performance tolerance, and quality of service is reported in [4].

Playout interval adjustment is a critical issue of AMP. The adjustment should be adequate, or the client would run into the risk of buffer underflow. Besides, the associated variation in playout interval should be smooth lest viewing discomfort would occur. The work described in [5] takes smooth playout into consideration as well, but the issue associated with high packet loss rate is not addressed.

The rest of the paper is organized as follows. In Section 2, we describe the related work. In Section 3, the channel model and streaming system in our work are specified. In Section 4, the proposed AMP control is described. Finally, experimental results and conclusions are given in Sections 5 and 6, respectively.

2. Related Work

As opposed to the approaches described in [2] and [6], where delay jitter is the primary concern, we place our focus on the influence of packet loss for the following two reasons. First, as stated in [3], with buffering delays on the order of seconds, the effect of delay jitter, typically on the order of tens of milliseconds, is relatively minor compared to that of the packet loss. Second, wireless applications such as mobile ad-hoc network (MANET) are becoming more popular. The packet loss rate for such applications is high [7], so an AMP control that targets error-prone channels is needed.

The playout policy described in [3] is characterized by the following equation:

$$\mu(n)=\begin{cases}1/(s\cdot t_F) & n<N\\ 1/t_F & n=N\\ 1/(f\cdot t_F) & n>N\end{cases}, \qquad (1)$$

where $\mu(n)$ is the time interval between the current packet and the previous one to be played out, t_F is the default playout rate, n is the receiving buffer fullness, and N is a threshold value. The parameter s (with value greater than 1) is used to represent a decrease in the playout interval and f (with value smaller than 1) to represent an increase in the playout interval. Based on this policy, a Markov chain analysis is developed to exam the tradeoff between buffer underflow and latency. The values of s, f, and N, which control the performance of this policy, are empirically determined and fixed. A drawback of this approach is that fixed parameters can not fit all network conditions. Moreover, it lacks a control of smooth playout. When the buffer fullness is near the threshold value, $\mu(n)$ may fluctuate severely.

The model used in [3] is a two-state burst-error Markov chain without considering delay jitters. A similar model is adopted in our system described next.

3. System Model

As shown in Figure 1, the system model consists of a streaming server, an error-prone channel, and a streaming client. The server sends out video packets at a fixed interval R (R =33 msec/frame in our system). Video packets are passed to the client through the channel with packet loss rate a. Lost packets are not retransmitted. The client stores the received packets in the buffer and uses AMP to control the playout.

4. Proposed AMP Control

Our AMP control scheme can be divided into two parts: channel condition estimation (CE) and playout interval adjustment (SA). CE estimates the mean packet loss rate and SA makes the playout interval smoother. Figure 2 gives the block diagram of the AMP control.

4.1. Channel Condition Estimation (CE)

Let the mean frame loss rate be a, $0\le a\le 1$. Because the lost frames will never arrive at the client side and because R=33, the mean receiving interval is

$$33(1-a)+66a(1-a)+99a^2(1-a)+\ldots=\frac{33}{1-a}. \qquad (2)$$

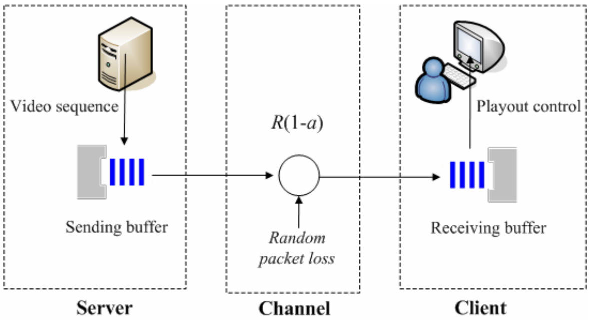

Figure 1. System model. The server sends out video packets every *R* msec, the packets are transmitted through a lossy channel with loss rate α, and the client uses AMP to control video playout.

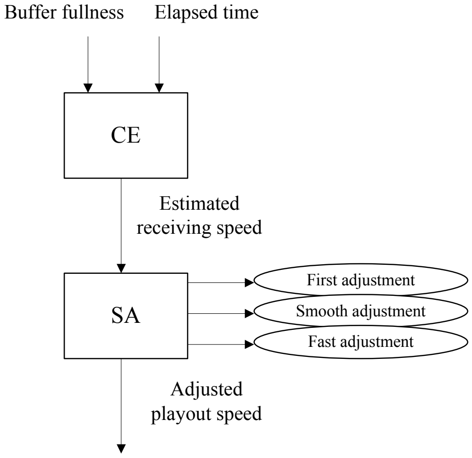

Figure 2. Block diagram of the Proposed AMP.

This indicates that the channel condition is reflected on the mean receiving interval. By estimating the receiving interval and setting the playout interval accordingly, we can tightly incorporate the current channel condition in the calculation of the playout interval. As the channel condition varies, the playout interval changes accordingly. This way, the buffer fullness is nicely controlled because the receiving interval and the playout interval are kept in harmony.

We estimate the receiving interval according to the buffer fullness and the elapsed time. After pre-rolling a number of frames in the receiving buffer, we record the present time and start the playout at the default playout interval p_0= 33 ms/frame. When no packet loss is experienced, the playout is very smooth, and the buffer fullness is almost constant. When packet loss occurs, the buffer fullness will start to decrease. When the decrement reaches x, which is a predefined value, the corresponding elapsed time y is recorded. The relation between y and the number of played frames during this decrement is:

$$y = p \times k, \quad (3)$$

where p is the playout interval, and k (k–x) is the number of played (received) frames during the decrement. As shown in Figure 3, the relation between p and the receiving interval r is governed by

$$p \times k - r \times (k - x) < r. \quad (4)$$

Since p and k are known, r can be estimated by choosing the minimum value r that satisfies the above relationship. The reason that we choose to use the minimum r is to reduce the viewing latency.

$$\hat{r} = \min(r)\text{: } r \text{ satisfies Eq. (4).} \quad (5)$$

The estimated receiving interval $\hat{r}$ is then used in the SA to adjust the playout interval. After recording the present time, we wait for another buffer increment or decrement of x to trigger another adjustment. The value of x is set to five in our experiment for its good performance.

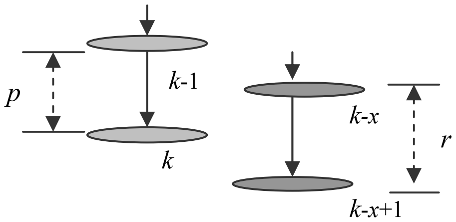

Figure 3. Receive interval estimation.

4.2. Playout Interval Adjustment (SA)

In this section, we show how to make the playout interval come close to the receiving interval in a gradual way such that the playout interval variation is kept smooth. Three policies of playout interval adjustment are designed: first adjustment to make them come close at the very beginning, smooth adjustment to fine tune, and fast adjustment to cope with sudden changes in network condition.

4.2.1. First Adjustment. At the very beginning we need to make the receiving interval and the playout interval close. Because the first estimation of the receiving interval may be inaccurate due to burst packet losses, we make the adjustment slighter: we adjust the playout interval p_1 to *mean* ($\hat{r}_1, p_0$), where $\hat{r}_1$ is the first estimated receiving interval.

4.2.2. Smooth Adjustment. After first adjustment, the fluctuation in buffer fullness is expected to become more moderate. We keep monitoring the buffer fullness. When *decrement*-of-five is detected, we increase our present playout interval by one. Comparatively, when *increment*-of-five is detected, no adjustment is made to alter the viewing latency. But if consecutive increment-of-fives are detected, the playout interval is reduced by one. These fine tunes are smooth and are capable of keeping the two intervals in harmony under slight channel condition fluctuations.

Note that we make no adjustment when single increment-of-five occurs because buffer overflow rarely happens to big buffers and the resulting buffer fullness increase can help lower buffer underflow risk.

4.2.3. Fast Adjustment. To prevent buffer underflow, we follow the $p_k = mean(r_k, p_{k-1})$ policy when the buffer fullness is too low. In cases where the packet loss rate drops suddenly or where severe burst-loss happens, this policy enables more rapid response.

5. Experimental Results

The parameters of interest in our experiments are the preroll delay, capability of buffer underflow control, and smoothness of the playout process. Preroll delay is the time lapse before the receiving buffer stores enough packets to start the playout. The more packets the buffer stores, the less likely it will undergo underflow. We fix the preroll delay to be 0.5 second (15 frames), and examine the buffer control and playout smoothness of several AMP control policies.

We use packetized bridge-close_cif.yuv [8] as test sequence. It contains 2,000 frames. We evaluate the buffer control by examining whether one AMP control can prevent buffer underflow at a given mean packet loss rate. In addition, we evaluate playout smoothness by the long-term and short-term standard deviations of playout interval (LSTD, and SSTD) as proposed in [5]. We set the calculation window size to be 1 second, and examine the peak SSTD instead of mean SSTD.

Our AMP control is compared with the one in [3], using different values of s (1.15, 1.25, and 1.33) and setting $f=s^{-1}$ and N =15 in Eq. (1). The experimental results are shown in Tables 1 and 2, with packet loss rate being 10% and 20%, respectively. Theoretical mean receiving intervals with different loss rate are computed using Eq. (2) to assist analysis, see Table 3.

When the packet loss rate is 10%, all AMP controls are able to prevent buffer underflow. Note the mean playout intervals in these controls are all adjusted near the theoretical value 36.67 frames. Smaller s value in Eq. (1) results in smoother playout, and our control gives the smoothest result.

When the packet loss rate is 20%, [3] with s=1.15 and 1.25 are fail to control buffer fullness because their slowest playout interval are not slow enough to

Table 1. Performances under 10% mean packet loss rate.

AMP	slowest playout interval	buffer underflow	mean playout interval	peak SSTD	LSTD
no control	33 ms/frame	after 120 frames	33.00 ms/frame	0	0
AMP in [3], s =1.15	38 ms/frame	no underflow	36.69 ms/frame	3.7953	2.7764
AMP in [3], s =1.25	41 ms/frame	no underflow	37.24 ms/frame	6.2668	5.4335
AMP in [3], s =1.33	44 ms/frame	no underflow	36.77 ms/frame	8.9617	8.4881
our control	44 ms/frame	no underflow	37.26 ms/frame	2.5321	1.4313

Table 2. Performances under 20% mean loss rate.

AMP	buffer underflow	mean playout interval	peak SSTD	LSTD
no control	after 67 frames	33.00 ms/frame	0	0
[3], s =1.15	after 264 frames	37.82 ms/frame	1.5374	1.0481
[3], s =1.25	after 834 frames	40.39 ms/frame	4.8315	2.6572
[3], s =1.33	no underflow	41.20 ms/frame	8.0540	6.0182
our control	no underflow	42.16 ms/frame	2.7218	1.2232

Table 3. Mean receive interval, computed using Eq. (2).

loss rate	mean receiving interval
10%	36.67 ms/frame
15%	38.82 ms/frame
20%	41.25 ms/frame

---

match the theoretical receiving interval. Again, our control provides smooth playout along with good buffer fullness control.

Besides, we should point out that parameters such as *s* and *f* are not needed in our AMP control. Therefore, our AMP control is not only smoother but also more adaptive than previous approaches.

In the last experiment, we change the mean packet loss rate according to the conditions specified in Table 5, and apply our AMP control. The resulting playout interval with respect to played frames is shown in Figure 4. The playout interval is adjusted close to theoretical receiving interval in response to the variation of packet loss rate.

6. Conclusions

In this paper, we have presented a novel AMP control with a smooth frame-rate adjustment scheme that provides more robust and smoother playback than previous approaches. This AMP control detects the channel condition automatically and keeps the playout interval closed to the received interval, thus it is more adaptive than conventional buffer fullness based controls. Experimental results show that the proposed AMP control algorithm performs well under the kind of high packet loss rates experienced in wireless applications.

References

[1] M. Roccetti et al, “Design and experimental evaluation of an adaptive playout delay control mechanism for packetized audio for use over the internet,” Multimedia Tools and Applications, May 2001, vol. 14, no. 1.

[2] S. B. Moon, J. Kurose, D. Towsley, “Packet Audio Playout Delay Adjustment: Performance bounds and algorithms”, in ACM Multimedia Systems, 1998.

[3] M. Kalman, E. Steinbach, and B. Girod, “Adaptive media playout for low-delay streaming over error-prone channels,” IEEE Transaction on CSVT, Jun. 2004, vol. 14, no. 6, pp. 841-851,.

[4] H. C. Chuang et al, “On the buffer dynamics of scalable video streaming over wireless network,” IEEE VTC, Los Angels, CA, Sept. 2004.

[5] H. C. Chuang et al, “A Novel Adaptive Video Playout Control for Video Streaming over Mobile Cellular Environment,” IEEE ISCAS, May 2005.

[6] D. L. Stone and K. Jeffay, “Queue monitoring: A delay jitter management policy,” NOSSDAV, Nov. 1993.

[7] K.Sundaresan et al, “ATP: A reliable Transport Protocol for Ad-Hoc Networks,” MobiHoc, 2003, pp. 64-75.

[8] http://trace.eas.asu.edu/yuv/index.html

Table 5. Packet loss rates.

frame number	packet loss rate
1 ~ 600	10%
601 ~ 1200	20%
1201~ 2000	10%

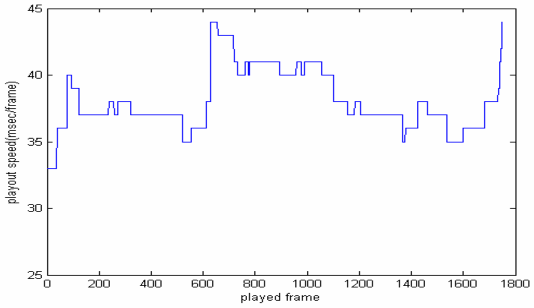

Figure 4. Variation of playout intervals as the loss rate varies as in Table 5.