Smooth Control of Adaptive Media Playout for Video Streaming

Ya-Fan Su, Yi-Hsuan Yang, Meng-Ting Lu, and Homer H. Chen, Fellow, IEEE

Abstract—Client-side data buffering is a common technique to deal with media playout interruptions of streaming video caused by network jitters and packet losses of best-effort networks. However, stronger playout interruption protection inevitably amounts to larger data buffering and results in more memory requirements and longer playout delay. Adaptive media playout (AMP), also a client-side technique, can reduce the buffer requirement and avoid buffer outage but at the expense of visual quality degradation because of the fluctuation of playout speed. In this paper, we propose a novel AMP scheme to keep the video playout as smooth as possible while adapting to the channel condition. The triggering of the playout control is based on buffer variation rather than buffer fullness. Experimental results show that our AMP scheme surpasses conventional schemes in unfriendly network conditions. Unlike previous schemes that are tuned for a specific range of packet loss and network instability, the proposed AMP scheme maintains consistent performance across a wide range of network conditions.

Index Terms—Adaptive media playout, error-prone channels, media buffering, video streaming.

I. INTRODUCTION

The popularity of applications such as video conference and video on demand is growing rapidly due to the advance of media streaming technology. By streaming, a video sequence is sent in the form of continual stream that can be played as it arrives, so the viewer can watch the video content instantly without downloading the media first. However, network delays and packet losses due to the inconsistency of network quality severely affect the QoS of media streaming. Client-side data buffering technique is a technique commonly adopted to address the issue. Although this technique is able to combat the impact of network jitter and playout interruptions, it introduces additional memory requirements [15] and playout delay [13]-[14]. Therefore, significant research efforts have been devoted to issues about buffering less data [26], [28] or introducing less playout delay [28]-[30] without sacrificing the QoS.

Adaptive media playout (AMP) is a technique applied at the client side for reducing data buffering while preventing buffer outage. It adjusts the playout interval to compensate for the fluctuation of network conditions. This is accomplished by increasing the playout interval when the buffer level is low, and vice versa. The premise of AMP is that the playout interval variation is more tolerable and less annoying to human perception than playout interruption and long delay [1], [9].

We are primarily concerned with AMP for video streaming. Issues related to AMP for audio streaming have been addressed elsewhere [2]-[3], [7]-[8], [16], [32] and are not considered here.

The AMP process must be as smooth as possible to minimize the degradation of visual quality caused by the variation of playout interval. Most previous AMP mechanisms, however, trigger the adjustment of playout interval according to the buffer fullness [4]-[6], [10], [19]-[20], [27], [30], [32]. As a result, the trigger can be easily pulled if the buffer fullness threshold is set too high, resulting in unnecessary playout interval adjustment even when the buffer is far from outage. This can easily degrade the visual quality and cause buffer fluctuation. Conversely, when the threshold is set too low, the playout control has a shorter reaction time and hence buffer outage is more likely to occur before the playout control is able to compensate for the estimation error of receiving rate. Normally, the threshold is dynamically determined according to the predicted channel quality [4], [5], [10], [19], high threshold for poor network conditions and vice versa. Unfortunately, it is difficult to reliably predict the channel quality [24], especially for wireless communication networks.

Unlike the conventional buffer-level-based approach, the quality-oriented AMP scheme proposed by Park and Kim [23] predicts the buffer occupancy and triggers the playout adjustment when the playout pause or skip is about to occur. Although it reduces the temporal distortion due to packet delay and jitter, the playout speed fluctuation caused by playout adjustment itself is not considered. Li et al. [31] proposed a scheme for content-aware AMP across the MAC layer and the video application layer. It reduces the perceptible effect of playout speed variation by considering the scene characteristics of a video sequence. However, the high computational complexity of this scheme makes it unsuitable for real-time implementation.

In this paper, we propose a novel AMP scheme to circumvent the difficulties described above and make the

Manuscript July 26, 2008. This work was supported in part by grants from the National Science Council of Taiwan under contracts NSC 96-2219-E-002-003 and NSC 95-2219-E-002-012.

Y.-F. Su, Y.-H. Yang, and M.-T. Lu are with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan.

H. H. Chen is with the Department of Electrical Engineering, Graduate Institute of Communication Engineering, and Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan (phone: +886-2-33663549; e-mail: homer@cc.ee.ntu.edu.tw).
streaming video playout as smooth as possible while avoiding buffer outage. The scheme monitors the receiving buffer and triggers the playout control when there is large variation in buffer fullness. We take the buffer variation rather than the buffer fullness level as the criterion for triggering the playout control because it is the buffer variation that reflects the deviation of the playout rate from the expected receiving rate. Although the buffer level reflects the potential likelihood of buffer outage, it is by no means an indication of the appropriateness of playout rate. We only use it to control the adjustment speed.

The rest of the paper is organized as follows. Section II describes the channel model and the overall streaming system. Section III describes the proposed AMP control in details. Section IV presents the experimental results and performance comparison, and Section V concludes the paper.

II. SYSTEM MODEL

As shown in Fig. 1, our system model consists of a streaming server, an error-prone channel, and a streaming client. We assume that the video data are always available at the server and the sending buffer is never empty.

The server sends out video packets at a fixed frame rate \( R \) (30 frame/sec in our system) to the client through a channel with packet loss rate \( \alpha(S_i) \), which is a function of the channel condition represented by the network state \( S_i \) and its value is between 0 and 1. The client stores the received packets in the buffer and uses AMP to adjust the playout rate. (The playout rate and the playout interval are used interchangeably throughout this paper.) When a packet is lost, the client checks the identifier of the lost packet, such as the sequence number. Then the client sends a retransmission request to the server if the playout time of the lost packet is not over and not too close to the current time (that is, \( t_i + RTT < t_p \), where \( t_i \) is the time the client detects the packet loss, \( t_p \) is the playout time of the lost packet, and \( RTT \) is the round trip time). The retransmission request suffers a loss rate of \( \beta(S_i) \). In our model, the server would not retransmit a packet unless it receives the retransmission request from the client.

A complete network model should also consider the state of the on-the-fly packets in addition to channel condition. But including the packet retransmission scheme in the system inevitably complicates the network modeling. For simplification, the packet retransmission failure caused by retransmission request loss or retransmitted packet loss is considered equivalent to packet loss and modeled as an increment of packet loss rate. The resulting total packet loss rate, which also ranges from 0 to 1, can be approximated by

\[
\gamma(S_i) \equiv \alpha(S_i) + \beta(S_i) \beta(S_i).
\]

The derivation of (1) is provided in Appendix I. The effect of packet retransmission due to error-prone channel is also considered by Kalman et al. [6], where a simplification similar to ours is made by modeling the retransmission of lost packets as a reduction of throughput.

We adopt a Markov model [18] for the system described above, with each state \( S_i \) representing a specific channel condition. More details of the Markov model are given in section IV.

III. PROPOSED ADAPTIVE MEDIA PLAYOUT

Fig. 2 shows the flow chart of our AMP scheme. Before starting to play the video, the client pre-buffers the video data.

![Figure 1. System Model. The server sends out video packets \( R \) frames/sec, the packets are transmitted through an error-prone channel modeled by a Markov model, and the client uses AMP to control video playout.](image1)

![Figure 2. Flow chart of the proposed AMP control scheme.](image2)

![Figure 3. The reference buffer level is updated each time when a PA order is issued.](image3)
until the buffer is half full to minimize the playout interruption. The system keeps monitoring the fullness and variation of the receiving buffer. Whenever there is a need to adjust the playout rate, the system issues a playout adjustment (PA) order. Then the playout interval is adjusted accordingly.

Specifically, if the receiving rate and the playout rate of a video sequence are roughly the same so that the receiving buffer stays within a predefined threshold \( r \) with respect to the reference buffer level, no PA order is issued. In this case, the system keeps the playout rate unchanged. On the other hand, if the receiving buffer deviates from the reference buffer level by a margin greater than or equal to \( r \), a PA order is issued to trigger the playout adjustment.

As mentioned earlier, our AMP scheme triggers the playout adjustment based on the buffer variation. This is effected by the use of dynamic reference buffer level in the decision process, and the difference between the buffer level and the reference buffer level is defined as the buffer variation. The reference buffer level is initialized to half of the buffer size at the moment when the client starts to play the video. As a PA order is issued, the reference buffer level is updated and set to the current buffer level. Fig. 3 illustrates the decision of the dynamic reference buffer level at \( t_1 \), \( t_2 \), and \( t_3 \). Note that a playout adjustment is triggered at \( t_3 \) even though there is no buffer outage concern at that time instant. This is the main distinction between our scheme and others. The buffer variation can be either positive (e.g., the one measured at \( t_3 \)) or negative (e.g., the one at \( t_1 \)). Without loss of generality, we only discuss the negative case in the rest of this paper.

When a PA order is issued, two values must be determined before the AMP scheme starts to adjust the playout interval. The first one is the target playout rate, and the second one is the speed of the playout adjustment (that is, how aggressively the playout rate is to be adjusted to the new rate). The target playout rate is determined by the receiving rate re-estimation (RRR) component of the system, and playout adjustment speed is determined by the expected change determination (ECD) component through a control parameter called expected change.

Based on the information coming from RRR and ECD, the AMP scheme computes the transition period and starts adjusting the playout interval. At the end of this period, the playout interval reaches its target value, and the system returns to normal video playout mode.

During the transition period of a playout adjustment, another playout adjustment order may be issued before the current transition period is over if the receiving rate estimated by RRR does not match with the actual value or if the channel condition changes. In either case, the AMP scheme terminates the current playout adjustment and starts executing the new one immediately. It follows the same procedure to determine the new transition period and adjust the playout interval accordingly.

Our scheme deals with unexpected network conditions this way without introducing any additional mechanism. In contrast, those AMP schemes based on the reference buffer level can easily suffer a buffer outage when unexpected network conditions occur.

In the following, we discuss the details of our scheme with respect to the estimation of receiving rate, the buffer outage avoidance, and the smoothness of playout adjustment.

A. Receiving Rate Re-Estimation

Without packet loss, the mean receiving interval is equal to the sending interval \( I \). It becomes greater than the sending interval when packet loss occurs. For simplicity, we assume each packet carries one video frame. Let \( k \) denote the number of consecutive lost packets before a successfully transmitted packet and \( p(k) \) the corresponding probability. Denote the receiving interval by \( I \). Then the mean receiving interval can be computed by

\[
E[I] = \sum_{k=0}^{\infty} E[I|k] p(k) \\
= \sum_{k=0}^{\infty} ((k+1)I_s) \times \gamma(S) (1-\gamma(S)) \]

\[
= \frac{I_s}{1-\gamma(S)} ,
\]

(2)

where \( E[\cdot] \) is the expectation operator and \( \gamma(S) \) the packet loss rate described by (1). The derivation of (2) is provided in Appendix II.

In practice, however, we cannot determine \( E[I|k] \) from (2) because \( \gamma(S) \) is not known. Fortunately, we can determine its upper and lower bounds and come up with an approximation for it.

Denote the current buffer variation by \( c \) if \( c \geq r \) and the time interval between the last PA order and the current PA order by \( s \). Suppose the client has played \( z \) frames during the time interval \( s \). Then the number of received frames during the interval is \( z+c \). As shown in Fig. 4, the relation between the time interval \( s \) and the mean receiving interval \( E[I] \) must satisfy

\[
(z+c-1)E[I] < s < (z+c+1)E[I] .
\]

(3)

Given \( s \), \( z \), and \( c \), we calculate the approximation \( \hat{E}[I] \) of the mean receiving interval by

\[
\hat{E}[I] = \frac{s}{z+c} .
\]

(4)
Note that we do not use the accumulative number of packets and the total elapse time to estimate the receiving interval because it is time variant. An estimation based on the most recent behavior of the network is preferred.

The target playout interval $I'$ is obtained by

$$I' = \hat{E}(I).$$

(5)

B. Expected Change Determination

To avoid abruptly changing the playout interval to the target playout interval, a smooth playout adjustment is desired. However, slowing down the adjustment speed increases the time it takes for completing the playout adjustment and hence creates the chance of buffer outage as well. Therefore, there is a need to maintain a good trade-off between the playout quality and the risk of buffer outage.

In this work, an expected change parameter is used to control the adjustment speed. It represents the expected amount of buffer to be changed, which is the difference between the expected buffer level at the end of the transition period and the current buffer level. When the current buffer level is far from outage, the system increases the magnitude of the expected change, resulting in a lower adjustment speed. On the contrary, when there is a risk of buffer outage, the system decreases the magnitude of the expected change. The expected change $C$ is determined by

$$C = \begin{cases} (M - \tau) - L, & L \geq M + \tau, \\ (L - \tau) - L, & L \leq M - \tau, \\ (M - 2\tau) - L, & \text{otherwise}, \end{cases}$$

(6)

where $L$ denotes the current buffer level and $M$ the middle buffer level. The value within each pair of parentheses in (6) is the expected buffer level at the end of the transition period.

We illustrate the three conditions in (6) with Fig. 5. When the buffer level is above the middle buffer level by a margin greater than or equal to $\tau$ (Fig. 5.a), the risk of buffer outage is low. Therefore, the expected buffer level is set to $M - \tau$, which ensures that $|C|$ is at least $2\tau$ and implies that the buffer level would not deviate too much from $M$ at the end of the playout adjustment. We prefer the buffer level staying around $M$, since it is the level having the least risk of buffer outage.

When the buffer level is below the middle buffer level by a margin greater than or equal to $\tau$ (Fig. 5.b), the risk of buffer outage is higher. Note that the expected change $|C|$ must be less than $\tau$ in such case, or another PA order would be issued during the current playout adjustment interval and causes the expected change to be re-determined. In other words, the expected buffer level would continually decreases until the buffer underflow occurs if the expected buffer level is set to a value less than $L - \tau$. Therefore, we set the expected buffer level to $L - \tau$, which corresponds to a maximal $|C|$ that can prevent the continual decrement in the expected buffer level.

When the buffer level is equal to the middle buffer level (Fig. 5.c), the expected change is heuristically set to $-2\tau$.

C. Transition Period Determination

In this step, the target playout interval $I'$ and the expected change $C$ are used to determine the transition period of the playout adjustment.

Assume for now that the transition period is given, the playout interval is adjusted from the value $I_0$ to $I'$ according to the transition function

$$I_t = I_0 + \frac{I' - I_0}{T}(t - t_0), \quad t_0 \leq t \leq t_0 + T$$

(7)

where $t$ denotes the time variable, $t_0$ the current time, $I_t$ the playout interval at time $t$, and $T$ the duration of the transition period. The playout interval $I_0$ at $t_0$ is determined by

$$I_0 = \begin{cases} I_0 + \Delta I, & C \leq 0, \\ I_0 - \Delta I, & C > 0, \end{cases}$$

(8)

where $I_0$ is the playout interval just before $t_0$ and $\Delta I$ (1 millisecond in our system) a fixed time interval. The playout interval starts from $I_0$ at $t_0$ and ends at $I'$ at the end of the transition period, $t_0 + T$. The playout interval is adjusted linearly, as it minimizes the maximum of playout interval variation during a given transition period.

Suppose the receiving interval is equal to the target playout interval $I'$ in (5), the buffer level change $\overline{C}$ during the
To evaluate the playout smoothness, we must precisely determine the distribution of packet losses.

\[ \sum_{ij} b_{ij} N_{ij} \gamma(S_i) = \sum_{ij} b_{ij} N_{ij} \gamma(S_j), \quad i, j \in [1, 2, \ldots, N] \text{ and } i < j, \]  

where \( N \) is the total number of states. Note that the multi-state Markov model is equivalent to a Gilbert-Elliot model when \( N \) equals two.

It has been reported that playout speed variation up to 25\% is unnoticeable [6]. According to (2), this corresponds to \( \gamma(S_i) = 0.2 \). Therefore, we limit the maximum packet loss rate \( \gamma(S_0) \) to 0.2 in our experiments.

The transition probability distribution of the multi-state Markov model is described by a transition matrix \( \mathbf{B} \), with each entry \( b_{ij} \) of the matrix being the probability of transitioning from state \( S_i \) to state \( S_j \). Also \( \sum_j b_{ij} = 1 \) for each row of \( \mathbf{B} \).

The transition matrix \( \mathbf{B} \) represents the stability of the network. We model the network by setting the transition matrix as

\[ b_{ij} = \begin{cases} \lambda, & i = j, \\ (1-\lambda)\left(N-1\right), & \text{otherwise,} \end{cases} \]

where \( \lambda \) is a network stability parameter, \( 0 \leq \lambda \leq 1 \). When \( b_{ij} = 1 \), the network state does not transfer from one to another, meaning the network is absolutely stable. On the contrary, when \( b_{ij} = 0 \), the network is completely unstable.

### IV. EXPERIMENTS

In this section, we first describe the network channel model used to simulate the network conditions and the metrics used for visual quality measurement of video playout. Then an experiment is conducted in Section IV-C to test the sensitivity of the parameter \( \tau \) required in our proposed AMP scheme. In Section IV-D we discuss how the parameter \( \tau \) can be determined for different buffer sizes. Finally, two experiments are conducted in Section IV-E to compare different triggering criteria for playout control.

#### A. Markov Network Model

Our AMP scheme aims at providing smooth playout in error-prone network environments such as the wireless channels. Typically, the Gilbert-Elliot model, which is a two-state (good, bad) discrete Markov model, is used to model the burst errors of a wireless channel [17], [22]. With this model, one can generate the trace of a network and use it to simulate different network conditions. However, instead of applying the Gilbert-Elliot model, we adopt a more general model called the multi-state Markov model [21] because it can model more precisely the distribution of packet losses. Besides, like the Gilbert-Elliot model, it has the ability to simulate burst errors. We use it to impose the transition of channel conditions in the experiments.

Each state \( S_i \) of the multi-state Markov model stands for a channel condition and has a packet loss rate \( \gamma(S_i) \). For simplicity, we label the states by the corresponding packet loss rates as follows:

\[ \gamma(S_i) < \gamma(S_j), \quad i, j \in [1, 2, \ldots, N] \text{ and } i < j, \]  

The first integral gives rise to the number of received frames and the second integral the number of played frames. We choose \( T \) such that \( \overline{C} \) is equivalent to the expected change \( C \) determined by (6). Then the duration of the transition period can be determined by substituting (6) and (7) into (9):

\[ T = C \left( \frac{1}{I^1} \frac{1}{I^2} \ln \left( \frac{I^1}{I^2} \right) \right)^{-1}. \]

### B. Playout Quality Metrics

The standard deviation of the playout interval within one second, denoted as the short-term standard deviation \( \sigma_5 \) [4], is adopted as the metric to evaluate the playout smoothness. A smaller \( \sigma_5 \) represents a smoother playout adjustment. In our experiments, we take the mean value \( \sigma \) of \( \sigma_5 \) over a video sequence as the measurement of playout quality of the sequence.

Since unsmooth playout usually occurs as a result of buffer outage, \( \sigma \) can also be considered an indication of the ability of
an AMP system to avoid buffer outage. Therefore, there is no need to use any additional metric such as mean time between buffer underflows (MTBBU) [6].

C. Sensitivity of Parameter $\tau$

In this section we show by experiment that the parameter $\tau$ required in our scheme has little relevance to the network condition and is easy to determine. For most previous AMP mechanisms that trigger the playout control when the buffer level is below a threshold, it is difficult to determine the threshold as it varies with the network condition. Our AMP scheme does not have the problem because it does not take the buffer level as the trigger criterion.

Since the packet loss rate and the network stability govern the network condition, we divide the experiment into two parts. In the first part, the network stability is fixed and the AMP scheme is tested with different packet loss rates. In the second part, the packet loss rate is fixed and the AMP scheme is tested with different degrees of network stability.

In the first part of the experiment the network stability is fixed by setting $\lambda$ to 0.5 and letting the Markov state transfer every 30 seconds. We adjust the maximum packet loss rate $\gamma(S_i)$ from 0.04 to 0.2 in 5 uniform steps and, without loss of generality, assume $\gamma(S_i)$ is proportional to $i$. Also, we set $N$ to 5 and the buffer capacity to 64 frames. The test video sequence is ten minutes long. The experiment is repeated 300 times, and the average of $\sigma_3$ is recorded.

The results are shown in Fig. 6(a), where each curve corresponds to a different $\gamma(S_i)$ value. As we can see, the higher $\gamma(S_i)$ is, the worse (higher $\sigma$) the AMP scheme performs. The curves are concave upward; therefore, neither large nor small $\tau$ should be chosen. This makes sense as a small $\tau$ causes the AMP scheme to frequently trigger the playout control and results in unnecessary playout adjustment, whereas a large $\tau$ causes slow reaction to the channel quality variation and results in buffer outage.

We can also see that the middle section of each curve (with $\tau$ ranging roughly from 5 to 8) in Fig. 6(a) is pretty flat; that is, the playout quality is insensitive to $\tau$. Thus any point in this middle section can be considered a good approximation of the minimum point of each curve. For simplicity, therefore, we may consider that the minimum points of these curves all correspond to the same $\tau$.

In the second part of the experiment, we fix $\gamma(S_i)$ at 0.15 and $N$ at 2 and adjust $\lambda$ from 0 to 0.8 in an increment of 0.2 to simulate different degrees of network stability. We let the Markov state transfer every 5 seconds. Other settings remain the same as the first part of the experiment. The results are shown in Fig. 6(b), where each curve corresponds to a different $\lambda$ value. Similar to the results shown in Fig. 6(a), $\sigma$ maintains approximately constant between $\tau = 7$ and $\tau = 9$.

Considering these two parts of the experiment together, we conclude that the proposed AMP scheme is insensitive to the packet loss rate and the network stability if we choose $\tau = 7$ or 8. The significance of this conclusion is that we can determine a fixed $\tau$ beforehand and that, as opposed to previous methods, there is no need to estimate the network conditions and adjust $\tau$ during video playout.

D. Determination of $\tau$ for different buffer sizes

We show in Section IV-C how to make the proposed AMP scheme insensitive to the network condition by choosing a proper buffer variation threshold $\tau$. There, we consider a fixed buffer size. The buffer variation threshold $\tau$ for different buffer sizes can be determined empirically in the same way.

Fig. 7 shows the results generated by the proposed AMP with different buffer sizes. Except for the buffer size, the other settings of the experiment remain the same as those of the experiment described in Section IV-C. As we can see, the optimal $\tau$ increases with the buffer size; a good choice of $\tau$ would be about 4 for buffer size = 32 frames and 12 for buffer size = 128 frames. More buffer sizes have been tested, although only three are shown in Fig. 7. Based on the experimental results, we find a linear relationship in the log domain between the optimal buffer size and the optimal buffer variation threshold $\tau_{opt}$:

$$\tau_{opt} = \begin{cases} 
4, & \text{buffer size} < 32, \\
2^{-0.8 \log_{2} (\text{buffer size})}, & 32 \leq \text{buffer size} \leq 128, \\
12, & \text{buffer size} > 128.
\end{cases}$$

Figure 7. Performance (in terms of the smoothness of playout interval) of the proposed AMP scheme under various network conditions and buffer sizes. For each buffer size, five performance curves, from top to bottom, corresponding to $\gamma(S_i) = 0.20, 0.16, 0.12, 0.08$, and 0.04 are shown.
there is large buffer variation, whereas the latter adjusts the playout interval even when the buffer is far from outage. Therefore, a trigger is desirable to determine \( \tau \) for a given buffer size.

### E. Benefit of Triggering Playout Control by Buffer Variation

An experiment is performed to compare two playout control schemes: one based on buffer variation and the other on buffer level. The main difference between them is that, as long as there is large buffer variation, the former adjusts the playout interval even when the buffer is far from outage, whereas the latter adjusts the playout interval only when the risk of buffer outage is high.

To evaluate the playout quality of the conventional AMP scheme using the buffer-level-based triggering criterion, the playout control is triggered when the buffer level is either below a low threshold \( T_L \) or above a high threshold \( T_H \). In our experiment, we set \( T_L = \tau \) and \( T_H = \text{buffer size} - \tau \).

The settings of the experiment are the same as the first part of the experiment described in Section IV-C. The ratio \( r \), which is a function of \( \tau \), is defined as

\[
r(\tau) = \sigma_p(\tau) / \sigma_c(\tau),
\]

where \( \sigma_p \) denotes the \( \sigma \) generated by the proposed scheme and \( \sigma_c \) the conventional scheme. The ratio \( r \) is used to compare the performance of the two AMP schemes by using the same control parameter \( \tau \). If \( r < 1 \), the proposed scheme performs better than the conventional scheme since it has a smaller \( \sigma \) value. The results are listed in Table I. It shows the proposed scheme outperforms the conventional scheme in most cases.

Since the value of \( \tau \) corresponding to the minimal \( \sigma \) of the two schemes may be different, it is more meaningful to compute the ratio

\[
R = \min_\tau (\sigma_p(\tau)) / \min_\tau (\sigma_c(\tau)),
\]

where \( \tau \in [3, 15] \), to compare the best performance of the two AMP schemes over a range of \( \tau \).

The results are listed in Table I as well, which shows that the proposed scheme outperforms the conventional scheme more as \( \gamma(S_N) \) increases. The value of \( \sigma \) decreases by 100% – 91.7% = 8.3% when \( \gamma(S_N) = 0.08 \) and 100% – 20.9% = 79.1% when \( \gamma(S_N) = 0.20 \). This is a desirable characteristic because the variation of playout interval is larger and easier to perceive when \( \gamma(S_N) \) is higher. In contrast, the increase of \( \sigma \) by 20% (= 120% – 100%) when \( \gamma(S_N) = 0.04 \) is negligible because \( \sigma \) is very small and the corresponding playout interval variation is hard to perceive.

Sudden change of channel condition is a challenge to AMP. Therefore, another experiment is performed to illustrate how

---

### Table I

<table>
<thead>
<tr>
<th>( \gamma(S_N) )</th>
<th>( \tau ) = 3</th>
<th>( \tau ) = 5</th>
<th>( \tau ) = 7</th>
<th>( \tau ) = 9</th>
<th>( \tau ) = 11</th>
<th>( R ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.04</td>
<td>1.391</td>
<td>1.219</td>
<td>1.158</td>
<td>1.231</td>
<td>1.300</td>
<td>120.0</td>
</tr>
<tr>
<td>0.08</td>
<td>1.628</td>
<td>1.512</td>
<td>0.472</td>
<td>0.569</td>
<td>0.475</td>
<td>91.7</td>
</tr>
<tr>
<td>0.12</td>
<td>1.984</td>
<td>0.523</td>
<td>0.374</td>
<td>0.396</td>
<td>0.422</td>
<td>82.6</td>
</tr>
<tr>
<td>0.16</td>
<td>0.271</td>
<td>0.409</td>
<td>0.205</td>
<td>0.314</td>
<td>0.495</td>
<td>35.0</td>
</tr>
<tr>
<td>0.20</td>
<td>0.147</td>
<td>0.244</td>
<td>0.164</td>
<td>0.253</td>
<td>0.689</td>
<td>20.9</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Mean receiving interval</th>
<th>Estimate of mean receiving interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Conventional scheme</td>
</tr>
<tr>
<td>0 ~ 2</td>
<td>33.33 ms/frame</td>
<td>33.33 ms/frame</td>
</tr>
<tr>
<td></td>
<td>33.33 ms/frame before PA order</td>
<td>33.33 ms/frame before 1st PA order</td>
</tr>
<tr>
<td>2 ~ 10</td>
<td>37.04 ms/frame</td>
<td>35.00 ms/frame after 1st PA order</td>
</tr>
<tr>
<td></td>
<td>37.00 ms/frame after PA order</td>
<td>38.00 ms/frame after 2nd PA order</td>
</tr>
</tbody>
</table>

---

Figure 8. Response to a sudden change of the packet loss rate. (a) The conventional AMP scheme, for which the buffer outage can easily occur if the estimate of channel condition is wrong. (b) The proposed AMP scheme, which successfully avoids buffer outage.
It is obvious that the probability of a packet transmitted from the server to the client without any retransmission at the network state $S_i$ is $1 - \alpha(S_i)$. To derive the probability of a packet that is successfully transmitted with $k$ ($k>0$) retransmission, we represent the transmission procedure by

$$F_j \rightarrow [(B_x \rightarrow F_j) \text{ or } B_j] \rightarrow \cdots \rightarrow [(B_x \rightarrow F_j) \text{ or } B_j] \rightarrow B_x \rightarrow F_j.$$  

This means that the first packet transmission is a failure, followed by $k-1$ retransmission failures due to retransmission request loss or retransmitted packet loss before the successful retransmission at the end. The probability of such transmission procedure under the network state $S_i$ is

$$\alpha(S_i)(1 - \beta(S_i))\alpha(S_i) + \beta(S_i)^{k-1}(1 - \beta(S_i))(1 - \alpha(S_i)).$$

The overall packet loss rate is

$$\gamma(S_i, N) = 1 - \sum_{i=0}^{N} P(k),$$

where $P(k)$ is the probability of a packet that is successfully transmitted with $k$ retransmission requests and $N$ the maximum number of the retransmission requests a client can ask for. Based on the same assumption as that described in [6], which says that the round-trip time is short enough to allow several retransmission attempts before packets are due for playout, the overall packet loss rate can be approximated by

$$\gamma(S_i, N) = 1 - \sum_{i=0}^{1} P(k) = \alpha(S_i)(\alpha(S_i) + \beta(S_i) - \alpha(S_i)\beta(S_i))$$

$$= \gamma(S_i).$$

Because $\alpha(S_i)$ and $\beta(S_i)$ are both smaller than 1, $P(k)$ approaches zero when $k \geq 2$ and becomes negligible.

**APPENDIX II**

**DERIVATION OF THE ANALYTICAL MEAN RECEIVING INTERVAL**

We show how the mean receiving interval $E[I]$ in (2) is derived by starting with the expression

$$E[I] = \sum_{i=0}^{\infty} ((k + 1)I_i) \times \gamma^i(S_i)(1 - \gamma(S_i)).$$

Multiplying (22) by $\gamma(S_i)$ we have

$$\gamma(S_i) \cdot E[I] = \sum_{i=1}^{\infty} ((k + 1)I_i) \times \gamma^{i+1}(S_i)(1 - \gamma(S_i))$$

$$= \sum_{i=1}^{\infty} (kI_i) \times \gamma^i(S_i)(1 - \gamma(S_i)).$$

Subtracting (23) from (22) we have

$$(1 - \gamma(S_i)) \cdot E[I] = I_i(1 - \gamma(S_i)) + \sum_{i=1}^{\infty} I_i \times \gamma^i(S_i)(1 - \gamma(S_i))$$

$$= I_i(1 - \gamma(S_i)) \left( \frac{1}{1 - \gamma(S_i)} \right).$$

Therefore, we obtain

**APPENDIX I**

**MODELING PACKET RETRANSMISSION FAILURE AS AN INCREMENT OF PACKET LOSS RATE**

Let $F$ denote a packet transmitted from the server to the client and $B$ a retransmission request transmitted from the client to the server. Also, let the letters $f$ and $s$, respectively, denote a failure transmission and a successful transmission.

differently the two AMP schemes react to a sudden change of packet loss rate. The experiment is set up such that the two schemes have the same initial buffer fullness, equal to 32, and the same initial playout interval, equal to the packet sending interval. We set the parameter $\tau$ to 8 and let the packet loss rate start from 0 and change to 0.1 at $t=2$. The buffer fullness and the playout interval are recorded right after each frame is played. The mean receiving intervals computed by (2) and estimated by the two schemes are listed in Table II.

Since the channel condition changes suddenly at $t=2$, the AMP schemes cannot immediately come up with an accurate estimate of the receiving interval. Fig. 8(a) shows the resulting buffer fullness and playout interval of the conventional scheme. As we can see, the playout control is triggered near $t=5$, and the playout interval is rapidly adjusted. However, since the mean receiving interval is higher than its estimate, the buffer underflow occurs and results in large jitters in the playout interval.

Fig. 8(b) shows the results generated by the proposed scheme. Since the initial value of the reference level is 32 and $r$ is 8, the proposed scheme issues its first PA order when the buffer fullness decreases from 32 to 24. Although the proposed scheme also underestimates the receiving interval, unlike the conventional scheme, the underestimation does not result in buffer outage in our case because another PA order is issued when the buffer fullness drops to 16. After the second PA order is issued, the proposed scheme re-estimates the mean receiving interval and stops the buffer from decreasing. This experiment shows that the proposed scheme is more robust to the estimation error of network condition than the conventional scheme is. This advantage is attributed to the ability of our scheme to adjust the playout interval before the risk of buffer outage becomes high.

**V. CONCLUSION**

In this paper, we have proposed a novel AMP scheme to alleviate the visual quality degradation problem caused by the playout speed adjustment. The scheme triggers the playout control according to the buffer variations instead of the buffer level, as it is a better indication of the appropriateness of playout rate. The buffer level is taken into consideration after the playout control is triggered. It helps the scheme determine the playout adjustment speed to maintain a good trade-off between the quality of video playout and the risk of buffer outage. Experimental results show our AMP scheme effectively avoids the buffer outage and greatly improve the playout quality, measured in terms of the short-term standard deviation, especially when the packet loss rate is high. Furthermore, a major advantage of our AMP scheme is that it is less sensitive to the network conditions. Therefore, estimation of network condition is not required.
\[ E[I] = \frac{I_i}{1 - \gamma(S)} \]  

(21)

REFERENCES


---

**Ya-Fan Su** received the B.S degree in Electrical Engineering from National Taiwan University, Taiwan, in 2007. He is currently working toward the M.S. degree in the Graduate Institute of Communication Engineering, National Taiwan University. His research interests include image composition, stereo vision, and video content analysis.

**Yi-Hsuan Yang** received the B.S degree in Electrical Engineering from National Taiwan University, Taiwan, in 2006. He is currently working toward the Ph.D. degree in the Graduate Institute of Communication Engineering, National Taiwan University. His research interests include multimedia information retrieval and analysis, human-centered computing, and affective computing.

**Meng-Ting Lu** was born in Peng-Hu, Taiwan. He received the B.S degree and the Ph.D. degree in Electrical Engineering from National Taiwan University, Taiwan, in 2003 and 2008, respectively. He is currently working as a postdoctoral fellow in the Graduate Institute of Communication Engineering, National Taiwan University. His research interests include video streaming, peer-to-peer streaming, and video coding.

**Homer H. Chen** (S’83-M’86-SM’01-F’03) received the Ph.D. degree in Electrical and Computer Engineering from University of Illinois at Urbana-Champaign, Urbana.

Since August 2003, he has been with the College of Electrical Engineering and Computer Science, National Taiwan University, Taiwan, R.O.C., where he is Irving T. Ho Chair Professor. Prior to that, he held various R&D management and engineering positions with US companies over a period of 17 years, including AT&T Bell Labs, Rockwell Science Center, iVast, and Digital Island. He was a US delegate for ISO and ITU standards committees and contributed to the development of many new interactive multimedia technologies that are now part of the MPEG-4 and JPEG-2000 standards. His professional interests lie in the broad area of multimedia signal processing and communications. Dr. Chen is an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology. He served as Associate Editor of IEEE Transactions on Image Processing from 1992 to 1994, Guest Editor of IEEE Transactions on Circuits and Systems for Video Technology in 1999, and an Associate Editorial of Pattern Recognition from 1989 to 1999.