2011 IEEE International Symposium on Multimedia

Delay-Aware Loss-Concealment Strategies for Real-Time Video Conferencing

Jingxi Xu and Benjamin W. Wah
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{*jxxu@cse.cuhk.edu.hk, bwah@cuhk.edu.hk*}

***Abstract*—One-way audiovisual quality and mouth-to-ear delay (MED) are two important quality metrics in the design of real-time video-conferencing systems, and their trade-offs have significant impact on the user-perceived quality. In this paper, we address one aspect of this larger problem by developing efficient loss-concealment schemes that optimize the one-way quality under given MED and network conditions. Our experimental results show that our approach can attain significant improvements over the *LARDo* reference scheme that does not consider MED in its optimization.**

***Keywords*-Internet, video conferencing, delay-aware transmissions, loss concealment, real time, bandwidth constraint.**

I. Introduction

With increased bandwidth and computational power, video-conferencing systems are widely adopted to satisfy the demand of interactive communication and collaboration. Their main design goal is to achieve good perceptual quality, or quality of experience perceived by users. Two objective metrics are useful for measuring perceptual quality: one-way audiovisual quality as well as delay from the generation of the signals at the sender to their playback at the receiver (*mouth-to-ear delay* or MED) [1], [2]. The former represents the quality of the multimedia content, whereas the latter is related to users' experience of interactivity [3].

The audiovisual quality of a video-conferencing system over the Internet is highly affected by the condition of the network transport. As the Internet is a best-effort IP network that does not guarantee in-order arrivals of packets, congestion in intermediate switches may incur network delays, which result in failure to play back the content in time. Long-term predictions of its traffic behavior is hard because it is non-stationary and dynamic, especially over long-haul connections [3]. However, it has a number of features that can be exploited in developing efficient schemes for transmitting real-time multimedia data.

First, it is possible to statistically estimate the network behavior in a short duration (say within a few seconds), despite the fact that its long-term behavior is non-stationary. To demonstrate this fact, we investigated 480 network traces from the PlanetLab [4] with different sources and destinations at various times collected in 2007. Our results show that the average packet loss rates and average network delays are consistent between the past few seconds and the following one second. The mean absolute difference between the predicted and the actual packet loss rate is 2.83%, whereas the ratio of the mean absolute difference between the predicted and the actual average delays is 14.3%.

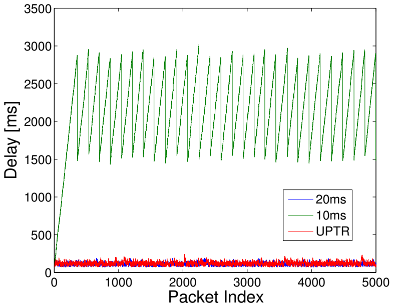


Figure 1. Delay behavior of a PlanetLab connection: a) Excessive delays experienced when packets were sent in 10 ms periods; b) Stable and short delays when packets were sent in 20 ms periods; c) Stable and short delays under UPTR (4 packets in 10 ms period and 6 packets in 26.6 ms period).

Our study further shows that instantaneous delays and loss rates will not change significantly when packets are sent at a reasonable bursty rate $PktRate^{\max}$, while maintaining a constant long-term average rate $PktRate^{\text{avg}}$. This is true because the Internet is packet-switched and can smooth *uneven packet transmission rates* (UPTR) [5]. Figure 1 illustrates that a PlanetLab link can tolerate a transmission rate of 50 packets per second (20 ms packet period), and that variations in instantaneous transmission rates do not affect the delay of packets. To overcome jitters at receivers, a playout scheduler with jitter buffers is generally used to store video and audio frames before playing them.

Last, the loss and delay behaviors in a packet-switched network are not sensitive to packet size as long as it is within the MTU (1500 bytes in wire-line networks), and the packet rate is not unreasonably high [6]. This property can be utilized to reduce the average packet rate by gathering multiple frames into a packet to within the MTU.

In general, there are trade-offs between the one-way audiovisual quality and the MED in a video-conferencing system. While MED can be increased for better one-way quality, the real-time perception may suffer. Their trade-offs for voice-over-IP (VoIP) transmissions have been studied

978-0-7695-4589-9/11 $26.00 © 2011 IEEE
DOI 10.1109/ISM.2011.14

elsewhere [7]. In this paper, we extend this earlier result and develop the coding and transmission schemes for audiovisual content in order to optimize the one-way signal quality over an Internet connection under a given MED. Our results will be useful for developing a complete video-conferencing system in the future with optimized perceptual quality.

A lot of efforts have been devoted to improving the one-way video quality over lossy networks. Source-level and channel-level loss-concealment schemes have been developed to help recover lost frames. Source-level protections are provided by codecs, whereas channel-level protections are achieved by redundant transmissions. Although the former works well in many cases, channel-level protections are needed under extreme conditions [8]. Their major drawback is that the additional time will be needed for transmitting redundant data and may increase the MED [9].

Among the various source level loss concealment schemes, *Loss Aware Rate Distortion optimization algorithms* (*LARDo*) [10] are the most popular for their good performance. However, they are not suitable for real-time applications because they are computationally expensive when estimating distortions caused by random channel behaviors.

Other source-level loss-concealment schemes, like reference picture selection (RPS), redundant slices (RS), and flexible macroblock ordering (FMO) [11], can protect data from transmission errors. However, they cannot fully recovery video frames in case of high loss rates and delay jitters, and they do not take advantage of the video content.

To reduce the negative effects caused by consecutive packet losses, delay-aware packet interleaving algorithms [12], [13] have been proposed to change the order of packet transmissions so that consecutive packet losses will not lead to consecutive video frame losses. However, they incur long delays to collect packets before playing back, which is not friendly to interactive applications with a short MED.

Kuipers, *et al.* proposed a method that uses parity FEC to recover error frames and adopted ITU-U Rec. G.1070 as the objective of the optimization [14]. Although it does not introduce extra delays for packet interleaving, it does not consider the video content in its optimization.

For improving one-way audio quality, Boutremans *et al.* proposed a delay-aware FEC algorithm for voice-only conferencing, which shows the importance of joint optimization of FEC and delays [9]. However, we have not found any delay-aware FEC algorithm in the literature that considers the joint optimization of video and audio quality.

Our main contribution in this paper is on the design of MED-aware strategies for coding and transmitting audiovisual contents, while considering the network condition in the recent past. By developing loss concealment strategies that take into account the properties of video contents, we can achieve better video quality even when compared to the computationally expensive *LARDo* algorithms. Using a short extra delay and without packet-interleaving, our method can achieve good quality under extreme network conditions. Finally, we schedule the transmission of video and audio packets together in order to reduce the risk of losing their synchronization.

The rest of this paper is organized as follows. Section II presents the previous and our proposed schemes for coding and transmitting audio/video content in video-conferencing systems. Section III presents the analysis of our loss concealment scheme, followed by its optimization in Section IV. Finally, Section V presents our experimental results.

II. Coding and Transmission Schemes

Given an MED, we like to achieve the best one-way audiovisual quality in terms of the minimum distortion. To this end, we present a loss-resilient coding scheme that utilizes the properties of the audiovisual content, as well as a packet transmission scheme that exploits the network behavior. As one-way perceptual quality is strongly affected by lip synchronization [15], our schemes try to keep accurate lip synchronization in the audio and video streams.

A. Video Coding and Transmission Schemes

1) Coding scheme: In an error-prone network, packets containing encoded video frames may be lost. Due to motion-compensations employed in most video coding algorithms, such as the H.264/AVC, the loss of a single packet may result in unpleasant artifacts in subsequent frames, leading to severe degradations in video quality.

To stop error propagations, intra-pictures or macroblocks are often inserted into a video stream [11]. Although intra-macroblocks are more space-efficient than the larger intra-pictures, they can only remove partial error drifts, since there are still inter-coded macroblocks. It is infeasible to determine in real-time whether a macroblock should be intra-coded as the process is computationally expensive [10].

In our approach, we use I-frames instead of intra-macroblocks, as I-frames can completely remove error propagations in the sequence. We assume that the very first I-frame (short for intra-picture) is always correctly received (say by retransmissions in initialization). Since the video content in video conferencing usually features a head-shoulder foreground and a static background, we code subsequent I-frames from the first I-frame using P-frames with the reference picture selection (RPS) provided in H.264/AVC [11] in order to reduce their size [5].

A reference of subsequent I-frames to the first I-frame may not work well when the background or the scene is dynamic, say when new participants join. To address this issue, we propose a referenced I-frame update scheme that allows the source to update old I-frames to the most recent I-frame acknowledged by the receiver.

To further eliminate error drifts, several refreshing frames called bridge frames (G-frames for short) are placed in a group of pictures (GOP). These refer to the same I-frame

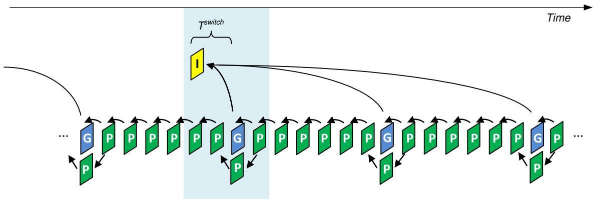


Figure 2. Our proposed coding scheme. Figure 3 shows the transmission scheme of frames in the blue block . (G: G-frames; P: P-frames; I: I-frames.)

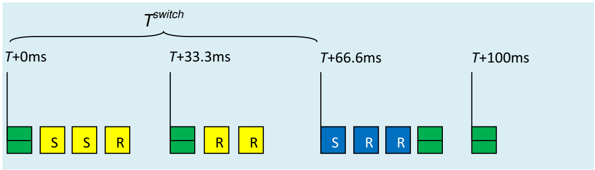


Figure 3. Our proposed transmission scheme (S: source frames; R: redundant frames in FEC; green blocks: frames protected using piggybacking). The same color in Figures 2 and 3 indicates the same type of frame.

(by RPS) and work as intra-pictures if the referenced I-frame is correctly received (see Figure 2). Although G-frames refer to a distant frame, they are just slightly larger than ordinary P-frames because video frames are all similar in a GOP.

Ordinary P-frames are coded based on previous frames as usual. When some P-frames are lost, error propagation will stop when the decoder encounters a correctly decoded G-frame. The interval for sending G-frames depends on the network condition and video complexity. If the loss rate is high or the video has complex motions, then error propagation will be serious and more G-frames will be needed. As we mainly use the G-frame period as the intra-refreshment period, we set the I-frame period (i.e., the length of a GOP) to 1 second for simplicity.

H.264/AVC provides a feature called redundant pictures, which sends additional frames in a video stream and has been proved a powerful error-resilient mechanism in error-prone networks [11]. We adopt this mechanism and always send an extra P-frame along with a G-frame in case the G-frame is not correctly decoded. This allows the following P-frame to refer to a correct P-frame or a G-frame (Figure 2) and reduces new errors in subsequent P-frames.

2) Transmission schemes:

a) *Channel-level protection.* Under high network losses or delays, source-level protection is inadequate for maintaining consistent video quality, and channel-level loss-concealment mechanisms will be needed to recover lost packets.

For real-time applications, channel-level protection cannot be based on retransmissions, since the round-trip delay for acknowledgments and retransmissions is much longer than what can be tolerated. Existing schemes are generally based on adding redundancy into a data stream. Two methods are popular in practice. In FEC [8], $n-k$ redundant packets are transmitted for every k source packets, and data can be recovered as long as k out of the n packets are correctly received. In piggybacking [6], one or more previous frames are duplicated and sent with a new frame in one packet. When a packet is lost or delayed, the information it contains can be recovered if any of the subsequent packets containing the redundant information is received in time.

By providing flexible protection degrees, FEC is useful for protecting large frames that need to be divided and sent in multiple packets. On the other hand, for smaller frames, FEC is not efficient because a small frame in a FEC block will not be able to fully utilize the payload space. Waiting for enough frames to fill up the space will lead to undue delays for all the information sent. In these cases, it is more effective to use piggybacking to gather old and new frames into a packet and send every new frame as soon as it has been encoded, while fully utilizing the space and without increasing the packet rate.

In our setup, P-frames are smaller than MTU (without using intra-macroblocks), while I- and G-frames are larger than MTU. We, therefore, adopt piggybacking to protect P-frames, and FEC for I- and G-frames (Figure 3).

To allow video frames to be played with consistent quality, the source data of every frame as well as its protection should be determined according to its type and properties. Trade-offs must be made between using the limited bandwidth for source data and for protection. Section IV presents the optimization of control parameters for transmitting source and redundant data in order to achieve the best reconstruction video quality.

In practice, the sustainable packet rate (that determines the best frame rate and size) must be dynamically selected according to the network condition and may depend on subjective preferences learned a priori. To illustrate our protection schemes proposed in this paper, we assume a CIF frame size at a constant packet rate of 50 packets/sec.

b) *Transmission strategy.* Based on UPTR observed in Section I, we send a small burst of packets at a higher rate $PktRate^{\max}$ in a short interval, while maintaining the average packet rate $PktRate^{\text{avg}}$. Figure 3 shows that packets containing P-frames are sent with packets containing I- and G-frames at a bursty rate $PktRate^{\max}$ in order to allow both I- and G-frames to be received earlier, without delaying the P-frames. We further place as much data as possible into a packet to within the MTU.

Since an I-frame is the reference frame for subsequent G-frames in a GOP, its loss will lead to the loss of subsequent G-frames. For this reason, we need to better protect I-frames with redundant information. However, the transmission of an I-frame and its redundant copy in a frame interval will exceed the prescribed bursty data rate $PktRate^{\max}$. To address this issue, we adopt our earlier method [5] to let the first G-frame in a GOP to refer to an I-frame sent T^{switch} earlier. This relaxes the time constraint for transmitting the I-frame in the current GOP, without exceeding $PktRate^{\max}$.

The consequence of relaxing the time constraint for an I-frame to be received may have some minor effect on the video quality at the receiver.

Unlike I-frames, G- and P-frames need to be displayed in real time before MED. Hence, they must have higher priority in transmission than I-frames. Between them, G-frames play the role of removing error propagation, and thus are more important than the redundant P-frames. To this end, we send a G-frame first if a frame interval has both.

The following summarizes our transmission strategy.

Algorithm 1 Transmission Strategy

```
for every GOP do
    while there are remaining packets of an I-frame do
        send a P-frame of a previous GOP;
        send packets of this I-frame in a bursty mode before
        the next P-frame is ready;
    end while
    while not end of this GOP do
        send packets of a G-frame in a bursty mode every
        G-frame interval;
        send packet of a new P-frame along with the
        previous $k-1$ frames ($k$ = piggybacking degree);
    end while
end for
```

B. Audio Coding and Transmission Schemes

There are four popular audio codecs in video conferencing: iLBC, iSAC (now part of WebRTC [16]), G729 [17], and G722.2 (AMR-WB) [18]. iLBC and iSAC are commercial products used by systems like Skype and Google Talk, whereas G729 (a narrow-band codec) and G722.2 (a wide-band codec) are open-source ITU standards. To follow the standard and to have better quality, we adopt G722.2 in our implementation.

G722.2 generates fixed size audio frames every 20 ms, with 8 sampling rates for producing audio of different qualities. We use the 15.85 kbps encoding mode in our implementation because it can produce satisfactory audio quality at reasonable bit rate (41 bytes/frame).

Under extreme network losses, the source protection in G722.2 is inadequate [19]. Similar to the protection of the smaller P-frames, we use piggybacking as a channel-level mechanism to protect them.

Separating the transmission of video and audio frames will increase the risk of losing synchronization, not to mention the increased packet rate. Hence, we packetize audio and the corresponding video frames in the same packet as far as possible and set a common MED for both. For simplicity, we place audio frames in P-frame packets. As they have different rates of generation, a packet may contain one P-frame and several audio frames. As a result, this arrangement may slightly increase the delay of audio frames.

III. Constraints of Our Proposed Scheme

In this section we present the constraints of our scheme as a function of network condition and frame type.

A. Control Parameters

As stated in Section II, we assume the length of a GOP to be 1 second. In this period, we need to determine $\#^{\text{G}}$, the number of G-frames to be inserted; $PiggyDeg^{\text{P}}$, the piggybacking degree of P-frames; and $PiggyDeg^{\text{A}}$, the piggybacking degree of audio frames. Let $(N^{\text{I}_\text{s}}, N^{\text{I}})$ (*resp.*, $(N^{\text{G}_\text{s}}, N^{\text{G}})$) be the number of source and total packets of a FEC block for an I-frame (*resp.*, a G-frame). For simplicity, we set the video frame rate F to a constant in this paper. In practice, F can be adjusted to perform congestion control according to the network bandwidth.

B. Time Constraints

a) *I-frame.* Every I-frame should be encoded T^{switch} earlier, so that when the first G-frame in a GOP needs to be decoded, the I-frame has been received and decoded. Thus,

$$T^{\text{switch}} \geq \frac{N^{\text{I}}}{\frac{PktRate^{\text{max}}}{F} - 1 - \frac{N^{\text{G}} \times \#^{\text{G}}}{F}} \times \frac{1}{F}.$$

The i^{th} I-frame packet meets the MED constraint only if

$$T_i^{\text{I}} + T_j^{\text{net}} - T^{\text{switch}} \leq MED,$$

where T_i^{I} is the buffering time before sending the i^{th} I-frame packet, and T_j^{net} is the network delay of this packet.

b) *G-frame.* It should be transmitted in a P-frame interval:

$$N^{\text{G}} + 1 \leq \frac{PktRate^{\text{max}}}{F}.$$

The i^{th} G-frame packet meets the MED constraint only if

$$\frac{i-1}{PktRate^{\text{max}}} + T_j^{\text{net}} \leq MED.$$

c) *P-frame.* The i^{th} P-frame meets the constraint when

$$\begin{cases} \frac{N^{\text{G}}}{PktRate^{\text{max}}} + (k-1) \times \frac{1}{F} & \\ \quad +T_j^{\text{net}} \leq MED & \text{in a G-frame interval} \\ (k-1) \times \frac{1}{F} + T_j^{\text{net}} \leq MED & \text{otherwise.} \end{cases}$$

where $k \leq PiggyDeg^{\text{P}}$ is the piggybacking degree of this P-Frame in the stream.

d) *Audio-frame.* We pack audio frames with P-Frames, where the k^{th}, $k \leq PiggyDeg^{\text{A}}$, copy of an audio frame meets the MED constraint when

$$k \times \frac{1}{F} + T_j^{\text{net}} \leq MED.$$

C. Bandwidth Constraints

According to UPTR, the number of packets/sec should be bounded at a reasonable value to prevent congestion delays and losses:

$$N^{\text{I}} + N^{\text{G}} \times \#^{\text{G}} + F = PktRate^{\text{avg}}.$$

By using FEC, the number of source packets should be less than the total number of packets for all I- and G-frames:

$$N^{\mathrm{I_s}} \leq N^{\mathrm{I}}, \quad N^{\mathrm{G_s}} \leq N^{\mathrm{G}}.$$

We also consider the overhead of the IP/UDP/RTP headers that consume, respectively, 20/8/12 bytes. Then the payload sizes in a packet for I- and G-frames are:

$$S^{\mathrm{I}} = S^{\mathrm{G}} = MTU - S^{\mathrm{IP}} - S^{\mathrm{UDP}} - S^{\mathrm{RTP}},$$

where S is the number of bits for transporting data.

By using piggybacking, we gather a new P-frame and copies of previous P-frames as well as audio-frames into a P-frame packet. Hence, the maximum size of a P-frame is constrained by the piggybacking degree of P-frames:

$$S^{P} = \frac{MTU - S^{\mathrm{IP}} - S^{\mathrm{UDP}} - (S^{\mathrm{Audio}} + S^{\mathrm{RTP}}) \times \#^{\mathrm{maxA}}}{PiggyDeg^{\mathrm{P}}} - S^{\mathrm{RTP}},$$

where $\#^{\mathrm{maxA}}$ is the maximum number of audio frames in a P-Frame packet, and $S^{\mathrm{Audio}} = 41$ bytes for a G722.2 frame with 15.85 kbps sampling rate. In our experiments, 3-way piggybacking has been found to be adequate for audio frames under most cases. This results in $\#^{\mathrm{maxA}} \leq 5$ when the video (*resp.* audio) frame rate is 30 fps (*resp.* 50 fps).

IV. Optimization of Audiovisual Quality

Under given MED and network conditions, we optimize the audiovisual quality by finding the best control parameters to minimize distortion.

A. Optimization of Audio Quality

With given codec setting, the distortion of the audio stream is related to the *unconcealed frame loss rate* (UCFR), or the rate of unrecovered frames after loss concealment. Past experimental results have shown this distortion to be insignificant when UCFR$\leq 2\%$ [3]. Accordingly, we set the piggybacking degree to a value that can achieve UCFR$\leq$2%, based on network conditions captured in the past 7 sec. To model the trade-offs between video and audio quality under $PktRate^{\mathrm{avg}}$, we define α to be the fraction of bandwidth allocated to audio data. In general, α depends on user preference and may have to be learned a priori. We bound the piggybacking degree by a function of α.

$$PiggyDeg^{\mathrm{A}} \times (S^{\mathrm{Audio}} + S^{\mathrm{RTP}}) \times F^{\mathrm{A}} \leq \alpha MTU \times PktRate^{\mathrm{avg}}, \quad (1)$$

where F^{A} is audio frame rate. For simplicity, we assume a fixed α that results in $PiggyDeg^{\mathrm{A}} \leq 3$. In practice, α may be reduced (*resp.* increased) if better video (*resp.* audio) quality is desired.

B. Optimization of Video Quality

For video, we use a distortion model to estimate the quality of the reconstructed video. This distortion is affected by the source distortion (or quantization distortion) and channel distortion (or distortion due to losses or delays). We first estimate these distortions and then present a method for calculating the objective of minimizing the expected reconstruction distortion.

1) Estimating the Source Distortion: Given the bits for encoding a frame, we can calculate the corresponding quantization step Q_{step} as follows:

$$S = c_1 \times \frac{\tilde{\sigma}}{Q_{\mathrm{step}}} + c_2 \times \frac{\tilde{\sigma}}{Q^2_{\mathrm{step}}}, \quad (2)$$

where S is the total size of a frame, c_1 and c_2 are two coefficients, and $\tilde{\sigma}$ is predicted by a linear model using the actual *mean absolute difference* (MAD) of the previous stored pictures [20]. Therefore, we can estimate the resulting Q_{step} from S. Further, it is well known that the source distortion can be estimated by

$$D^{\mathrm{source}} = \frac{Q^2_{\mathrm{step}}}{12} \quad (3)$$

in *mean squared error* (MSE) scale [21]. Similar to [21], we assume that the rate-distortion model is available for all I-, G-, and P-frames, but with different weights:

$$D^{\mathrm{source}} = \frac{\left(\frac{Q_{\mathrm{step}}}{\theta}\right)^2}{12}. \quad (4)$$

θ can be calculated by encoding and decoding the three types of frames and by calculating the ratios of distortions at the beginning of every GOP. Since this is too computationally expensive, we estimate them offline for a system with the same frame size and T^{switch}. This can be done by averaging the values obtained in various test videos. In our system with VGA frames and $T^{\mathrm{switch}} = 15ms$, we use $\theta^I = 0.8$, $\theta^G = 0.9$ and $\theta^P = 1$. We also consider the consistency of video quality in the sequence of frames. To avoid the quality of a frame suddenly degraded and users perceiving a significant blur, we keep the distortion of the three types of frames identical:

$$D^{\mathrm{I}} = D^{\mathrm{G}} = D^{\mathrm{P}}. \quad (5)$$

The resulting source distortion equals $\min(D^{\mathrm{I}}, D^{\mathrm{G}}, D^{\mathrm{P}})$ if these distortions are not equal.

2) Estimating the Channel Distortion: A receiver-based loss-concealment scheme may affect the reconstruction distortion. Although spatial- and frequency-domain recovery schemes can conceal losses gracefully, they add extra complexity in computation [22]. Hence, we use the simple frame-copy loss-concealment approach [22]. The result of channel losses can then be estimated by D^{diff}, the MSE of two sequential frames received. For computational efficiency, we only calculate this when an I-frame is coded and consider it the same for the other frames in the GOP. When several P-frames are lost, distortions will increase if there is no refreshing G-frame to stop the error propagation. Let $D^{\mathrm{accuerror}}$ be the accumulated error distortion.

a) When an I-frame is lost, all G-frames will be rendered useless because the reference frame is missing.

b) When a G-Frame is available, the distortion $D^{\mathrm{accuerror}}$ caused by the accumulated error will return to 0. However, when a G-frame is lost or delayed, $D^{\mathrm{accuerror}}$ will almost remain the same if a corresponding P-frame is received. (In fact, there will be a small difference between a G-frame and a redundant P-frame). Otherwise, $D^{\text{accuerror}} = D^{\text{accuerror}} + D^{\text{diff}}$.

c) With a lost P-frame, $D^{\text{accuerror}} = D^{\text{accuerror}} + D^{\text{diff}}$.

3) Deriving the Reconstruction Distortion: Displayed frames are frames reconstructed at receivers, whose distortion D_i can be estimated by

$$D_i = \begin{cases} D_i^{\text{G}} & \text{for correctly decoded G-frame} \\ D_i^{\text{P}} + D^{\text{accuerror}} & \text{if only P-frame is available} \\ D_{i-1} + D^{\text{accuerror}} & \text{if no available P- or G-frame.} \end{cases} \quad (6)$$

Note that in the case when there is no available P- or G-frame, the error is accumulated twice, leading to a quadratic penalty for a longer stretch of lost frames.

If we know the status of the I-/G-/P-frames received, we can calculate the distortion of every reconstructed frame. The overall distortion is the average distortion of all the reconstructed frames:

$$D = \frac{1}{F}\sum_{i=1}^{F} D_i. \quad (7)$$

4) Expected Reconstruction Distortion: Since the packet loss rate and delay can be estimated by statistics collected in the recent past (Section I), we use the packet loss rate and the CDF of network delays for calculating the expected reconstruction distortion. For every GOP, we first derive the loss rate of different frames and then calculate the expected reconstruction distortion for every reconstructed frame. Finally, we use their mean as the expected overall distortion of the video sequence.

a) *I-frame loss rate.* The loss of a frame may be caused by either the loss of packets or their late arrival. Thus, the loss rate of the j^{th} packet of an I-frame is

$$p_j^{\text{I}} = p + (1-p)\left(1 - CDF\left(MED + T^{\text{switch}} - T_j\right)\right), \quad (8)$$

where p is the packet loss rate, and T^j is the buffering time before sending this packet.

Since we use FEC to protect an I-frame, if we receive at least $N^{\text{I}_{\text{S}}}$ out of N^{I} packets, then this I-frame is correctly received. Therefore, the loss rate of an I-frame is

$$p^{\text{I}} = \sum_{l=0}^{N^{\text{I}_{\text{S}}}-1} Pr(\text{only } l \text{ packets are received}). \quad (9)$$

b) *G-frame loss rate.* The loss rate of the j^{th} packet of a G-frame is

$$p_j^{\text{G}} = p + (1-p)\left(1 - CDF\left(MED - \frac{j-1}{PktRate^{\text{max}}}\right)\right). \quad (10)$$

Similarly, the loss rate of a G-frame is

$$p^{\text{G}} = \sum_{l=0}^{N^{\text{G}_{\text{S}}}-1} Pr(\text{only } l \text{ packets are received}). \quad (11)$$

c) *P-frame loss rate.* The loss rate of the k^{th} copy of a P-frame is

$$p_k^{\text{P}} = p + (1-p)\left(1 - CDF\left(MED - \frac{k-1}{F}\right)\right). \quad (12)$$

For simplicity, the loss rate of P-frames transmitted along with a G-frame is also estimated by this formula. This P-frame is lost only when all its copies are lost. We have

$$p^{\text{P}} = \prod_{k=1}^{PiggyDeg^{\text{P}}} p_k^{\text{P}}. \quad (13)$$

d) *Expected reconstruction distortion.* When an I-frame is lost, all G-frames in this GOP are rendered useless. Therefore, the probability that a G-frame cannot be recovered is

$$p^{\text{G}_{\text{useless}}} = p^{\text{I}} + (1-p^{\text{I}})p^{\text{G}}. \quad (14)$$

Denote $D_j^{\text{accuerror}}$ as the expected error drift when decoding the j^{th} reconstructed frame:

$$D_j^{\text{accuerror}} = \begin{cases} p^{\text{G}_{\text{useless}}}(1-p^{\text{P}})(D_{j-1}^{\text{accuerror}} + \beta D^{\text{diff}}) & \\ +p^{\text{G}_{\text{useless}}}p^{\text{P}}(D_{j-1}^{\text{accuerror}} + D^{\text{diff}}) & \text{if in G-frame interval} \\ (1-p^{\text{P}})D_{j-1}^{\text{accuerror}} & \\ +p^{\text{P}}(D_{j-1}^{\text{accuerror}} + D^{\text{diff}}) & \text{otherwise,} \end{cases} \quad (15)$$

where $D_0^{\text{accuerror}} = 0$, which means no error drift at the beginning of a GOP. Here β is for estimating the small difference between the G-frame and the redundant P-frame. Let $MSE^{\text{G}_{\text{P}}}$ be the MSE between a decoded P-frame and G-frame, we have

$$\beta = \frac{MSE^{\text{G}_{\text{P}}}}{D^{\text{diff}}}. \quad (16)$$

β can be updated whenever both the G- and P-frames are received. However, this is inefficient because it needs extra decoding. In practice, β is calculated by averaging the values found in offline experiments with test videos. We found $\beta = 0.05$ to be reasonable in our experiments.

We can now derive the expected distortion of the j^{th} reconstructed frame:

$$D_j = \begin{cases} (1-p^{\text{G}_{\text{useless}}})D_j^{\text{G}} + p^{\text{G}_{\text{useless}}}(1-p^{\text{P}})D_j^{\text{P}} & \\ +p^{\text{G}_{\text{useless}}}p^{\text{P}}D_{j-1} + D_j^{\text{accuerror}} & \text{if in G-frame interval} \\ (1-p^{\text{P}})D_j^{\text{P}} + p^{\text{P}}D_{j-1} + D_j^{\text{accuerror}} & \text{otherwise} \end{cases} \quad (17)$$

$$\text{and } D_0 = \begin{cases} 0 & \text{for the first GOP} \\ D_{t-1}^{\text{previous}} & \text{otherwise,} \end{cases} \quad (18)$$

where $D_{t-1}^{\text{previous}}$ is the distortion of the last frame in the previous GOP.

The overall distortion D can then be calculated by taking the average of those reconstructed frames defined in (7).

To have the optimal video quality, we want to minimize D within the time and bandwidth constraints. Since the control parameters in our scheme are discrete, we can calculate (7) for every feasible parameter set and find the one with the minimum distortion. This optimal set will be used as the control parameter set for the next GOP.

Table I
PLANETLAB TRACES COLLECTED IN 2007.

ID	Avg Delay (ms)	StdDev Delay (ms)	Loss Rate (%)	AVGPktRate (pkts/sec)	MAXPktRate (pkts/sec)
1	214	5	43	50	100
2	80	4	14	25	50
3	107	138	0	50	100
4	119	16	7	50	100

Specifically, for $(\#^{\mathrm{G}}, \mathit{PiggyDeg}^{\mathrm{P}}, N^{\mathrm{Is}}, N^{\mathrm{I}}, N^{\mathrm{Gs}}, N^{\mathrm{G}})$, the maximum available size of I- and G-frames are constrained by the size of P-frames in (5) to keep a consistent source distortion. Therefore, $\mathit{PiggyDeg}^{\mathrm{P}}$, N^{Is}, N^{Gs} are bounded by $\mathit{PiggyDeg}^{\mathrm{P}_{\max}}$, $c_3\mathit{PiggyDeg}^{\mathrm{P}_{\max}}$, $c_4\mathit{PiggyDeg}^{\mathrm{P}_{\max}}$ respectively, where c_3 and c_4 are constants. N^{I} and N^{G} are both bounded by $\mathit{PktRate}^{\mathrm{avg}} - F$, because P-frames use F packets/sec. Finally, $\#^{\mathrm{G}}$ can be determined when other parameters are fixed. Thus, the computational complexity is

$$O(\mathit{PiggyDeg}^{\mathrm{P}_{\max}\,3}(\mathit{PktRate}^{\mathrm{avg}} - F)^2 F).$$

The computation can be done in real time, since we only need one such search for each GOP every second.

V. EXPERIMENTAL RESULTS

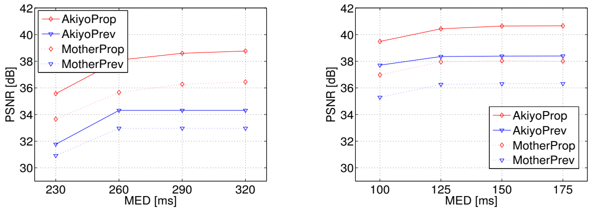
We have conducted experiments to evaluate the performance of our scheme. Our first (*resp.* second) experiment demonstrates that it has better video (*resp.* audiovisual) quality than the reference *LARDo* scheme over different MEDs. The implementation of *LARDo* is in JM with 30 decoders, and we adopt a GOP structure of IPPPP. Although its complexity precludes it for real-time applications, its significant improvement in video quality make it a widely adopted reference. To ensure that the reference scheme is an ideal implementation not affected by uncertain network behavior, we use future network statistics to calculate the estimated $\mathit{LossRate}$ for *LARDo*. Other parameters of the referenced codec were kept the same as ours. To ensure lip synchronization, we packetize audio frames in video frame packets and send every audio frame as soon as they are generated, while assuming a video frame is always ready to be sent together.

Our proposed video coding scheme is implemented using the Joint Model (JM) version 16.2 [23] of H.264/AVC, and our audio coding scheme using the G722.2 Fixed-point C-code [24]. The network statistics collected in the past 7 seconds with 500-ms delay is used to predict the control parameters in the next second. Audio frames are coded in the G722.2 15.85 kbps mode and use merely source-level protection.

We have developed a network simulator that drops or delays packets according to PlanetLab network traces. Table I presents four representative traces under various network conditions (Traces 1 & 3 representing rather extreme conditions). Figure 6 shows the latency histogram of Trace 3. In our experiments, we measure video quality by PSNR and audio quality by PESQ [25].

(a) Akiyo (b) Moth.-D'ter (c) Foreman (d) Outdoor Girl

Figure 4. Test sequences used: (a) Akiyo with 1500 CIF frames at 15/30 fps; (b) Mother and Daughter with 1500 CIF frames at 15/30 fps; (c) Foreman with 300 CIF frames at 30fps (d) Outdoor Girl with 779 CIF frames at 30 fps.



(a) Trace 1 (b) Trace 2

Figure 5. A comparison of the video quality in PSNR between our proposed and the reference schemes.

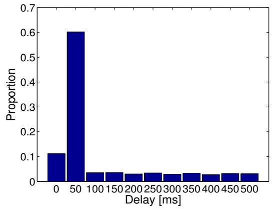


Figure 6. Latency histogram of Trace 3.

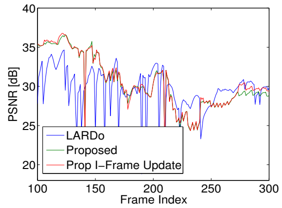


Figure 7. PSNR of the reference scheme and our proposed schemes using Foreman.

Table II
COMPARISON OF ERROR-FRAME RATIOS OF AKIYO IN TRACE 1

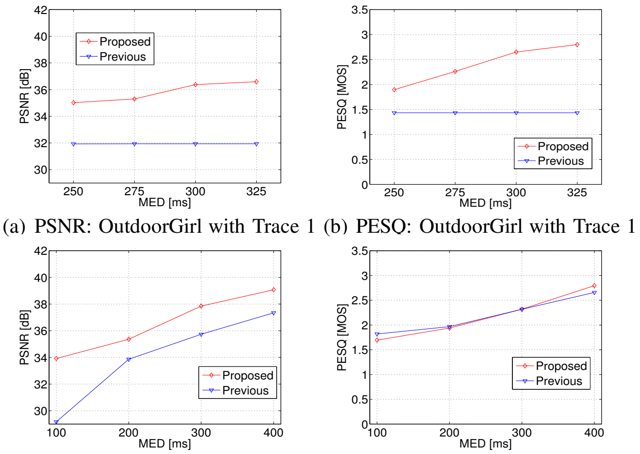
MED	230	260	290	320
Referenced	79%	66%	66%	66%
Proposed	50%	24%	17%	14%

A. Performance of Video Coding and Transmission Schemes

Figures 4a and 4b show the two standard CIF videos used in our experiments: *Akiyo* and *Mother and Daughter*, both representing typical slow-moving scenarios with a static background. We test them under Trace 1 (with high losses) and Trace 2 (with medium losses). To fit the available packet rate, we choose the frame rate to be 30 fps for Trace 1 and 15 fps for Trace 2. To overcome possible bias of the network trace at different times, we paste multiple copies of each test video into one long test sequence with 1500 frames.

Figure 5 shows that our proposed scheme outperforms the reference scheme in all test cases with various MEDs. The improvement in terms of PSNR can be as much as 2 dB. These improvement is mainly due to the proposed source- and channel-level error-concealment schemes which largely reduce the number of error frames (see Table II).

To illustrate that our proposed scheme can work well under dynamic scene changes, we compare the performance of the reference scheme, the proposed scheme with/without I-frame update using the the standard 30-fps CIF test video *Foreman* and Trace 4. (*Foreman* has fast scene changes



(a) PSNR: OutdoorGirl with Trace 1 (b) PESQ: OutdoorGirl with Trace 1

(c) PSNR: OutdoorGirl with Trace 3 (d) PESQ: OutdoorGirl with Trace 3

Figure 8. A comparison of video (in PSNR) and audio (in PESQ) quality between our proposed and the reference schemes

from frames 160 to 240.) Figure 7 shows that our proposed schemes have much better video quality (in PSNR) before the scene change, and they can recover quickly after a loss. After the scene has changed, the PSNRs of all three schemes rise again. However, the proposed scheme without I-frame update cannot reach the same quality as the reference due to its reference to a distant I-frame. With I-frame updates that replace an I-frame by a correctly decoded frame every second with 1-sec acknowledgment latency, the video quality can recover from frame 260 and performs well after then.

B. Audiovisual Quality under Various MEDs

Figure 4d shows *Outdoor Girl*, a test sequence with both audio and video contents and more motions than *Akiyo* and *Mother and Daughter*. We tested it using Traces 1 and 3 under lossy and high-jitter conditions.

Figure 8 shows that our proposed scheme provides better video quality and better or equal audio quality than the reference scheme. Despite the fact that our scheme allocates more resources to audio (and thus less resources to video), it still outperforms the reference scheme in video quality. This demonstrates the advantage of our video coding scheme. The audio quality of our proposed scheme in Trace 3 is similar to that of the reference due to the high delay (Figure 6) that causes a large number of late audio packets. In this case, redundant packets do not help because they also miss the playout deadline. For this reason, our proposed scheme reduces the redundancy degree of audio packets so that more resources are given to the transmission of video data.

VI. Conclusions

In this paper, we have studied the optimization of one-way audiovisual quality under given MED and network conditions. We have designed effective schemes for coding and transmitting audiovisual content for a real-time video-conferencing system over the Internet. Our experimental results under various network conditions show the advantages of our scheme over a scheme not aware of MED in its coding and transmission strategies. Our future work will focus on the adaptive adjustments of MED and frame rates and the comparison with commercial products.

References

[1] ITU, "P.920: Interactive test methods for audiovisual communications," 2000.

[2] T. Hayashi, K. Yamagishi, T. Tominaga, and A. Takahashi, "Multimedia quality integration function for videophone services," in *Global Telecommunications Conf. (GLOBECOM)*. IEEE, 2007, pp. 2735–2739.

[3] B. Sat and B. Wah, "Playout scheduling and loss-concealments in voip for optimizing conversational voice communication quality," in *Proc. of the 15th Int'l Conf. on Multimedia*. ACM, 2007, pp. 137–146.

[4] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an overlay testbed for broad-coverage services," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3–12, 2003.

[5] J. Lu and B. Wah, "Scheduling transmissions of real-time video coded frames in video conferencing applications over the Internet," in *IEEE Int'l Conf. on Multimedia and Expo*. IEEE, 2010, pp. 1695–1700.

[6] B. Sat and B. Wah, "Analysis and evaluation of the Skype and Google-Talk VoIP systems," in *IEEE Int'l Conf. on Multimedia and Expo*. IEEE, 2006, pp. 2153–2156.

[7] ——, "Statistical scheduling of offline comparative subjective evaluations for real-time multimedia," *IEEE Trans. on Multimedia*, vol. 11, no. 6, pp. 1114–1130, 2009.

[8] P. Frossard, "FEC performance in multimedia streaming," *Communications Letters, IEEE*, vol. 5, no. 3, pp. 122–124, 2001.

[9] C. Boutremans and J. Le Boudec, "Adaptive joint playout buffer and FEC adjustment for Internet telephony," in *Twenty-Second Annual Joint Conf. of the IEEE Computer and Communications (INFOCOM)*, vol. 1. IEEE, 2003, pp. 652–662.

[10] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H. 264/AVC in wireless environments," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.

[11] S. Wenger, "H. 264/AVC over IP," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.

[12] Y. Liang, J. Apostolopoulos, and B. Girod, "Model-based delay-distortion optimization for video streaming using packet interleaving," in *Conf. Record of the 26th Asilomar Conf. on Signals, Systems and Computers*, vol. 2. IEEE, 2002, pp. 1315–1319.

[13] R. Razavi, M. Fleury, M. Ghanbari, and M. Sadeghzadeh, "Delay-aware interleaving and forward-error correction for video over wireless: a bluetooth case study," in *Proc. of the 3rd ACM Workshop on Wireless Multimedia Networking and Performance Modeling*. ACM, 2007, pp. 46–53.

[14] B. Kuipers, R. Vaz, and M. Nunes, "Video quality protection for real time video streams over wireless networks," *Telecommunication Systems*, pp. 1–12, 2011.

[15] ITU, "P.931: Subjective audiovisual quality assessment methods for multimedia applications," 1998.

[16] Google. WebRTC. [Online]. Available: https://sites.google.com/site/webrtc/home

[17] ITU, "G.729: Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear-prediction (CS-ACELP)," 1996. [Online]. Available: http://www.itu.int/rec/T-REC-G.729/en

[18] ——, "G.722.2: Wideband coding of speech at around 16 kbit/s using adaptive multi-rate wideband (AMR-WB)," 2003. [Online]. Available: http://www.itu.int/rec/T-REC-G.722.2/en

[19] B. Wah and B. Sat, "The design of VoIP systems with high perceptual conversational quality," *Ubiquitous Multimedia Computing*, vol. 5, p. 41, 2009.

[20] K. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods," *JVT of ISO/IEC MPEG and ITU-T VCEG, JVT- K*, 2004.

[21] N. Kamaci, Y. Altunbasak, and R. Mersereau, "Frame bit allocation for the h. 264/AVC video coder via cauchy-density-based rate and distortion models," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 994–1006, 2005.

[22] B. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet," in *Multimedia Software Engineering, 2000. Proc. Int'l Symposium on*. IEEE, 2000, pp. 17–24.

[23] J. Model. H.264/AVC reference software. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[24] ITU. ITU-T G.722.2 Annex C: Fixed-point C-code. [Online]. Available: http://www.itu.int/rec/T-REC-G.722.2/en

[25] J. Beerends, A. Hekstra, A. Rix, and M. Hollier, "Perceptual evaluation of speech quality (PESQ) the new itu standard for end-to-end speech quality assessment. part ii: Psychoacoustic model," *J. of the Audio Engineering Society*, vol. 50, no. 10, pp. 765–778, 2002.