Error Control Techniques for Interactive Low-bit Rate Video Transmission over the Internet

Injong Rhee
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534
E-mail: rhee@csc.ncsu.edu

Abstract

A new retransmission-based error control technique is presented that does not incur any additional latency in frame playout times, and hence are suitable for interactive applications. It takes advantage of the motion prediction loop employed in most motion compensation-based codecs. By correcting errors in a reference frame caused by earlier packet loss, it prevents error propagation. The technique rearranges the temporal dependency of frames so that a displayed frame is referenced for the decoding of its succeeding dependent frames much later than its display time. Thus, the delay in repairing lost packets can be effectively masked out. The developed technique is combined with layered video coding to maintain consistently good video quality even under heavy packet loss. Through the results of extensive Internet experiments, the paper shows that layered coding can be very effective when combined with the retransmission-based error control technique for low-bit rate transmission over best effort networks where no network-level mechanism exists for protecting high priority data from packet loss.

1 Introduction

Transmitting interactive video over packet switching networks is time-constrained. If a packet containing a video frame is not delivered before its scheduled playout time, and the frame cannot be displayed and accordingly, the overall quality of video degrades. The difficulty of meeting this time constraint lies mainly in frequent occurrences of packet loss which are commonly caused by network congestion. The recovery of lost packets before their display times is not always feasible because of the delay associated with detecting and repairing the lost packets.

The effect of packet loss is also far-reaching. Packet loss degrades not only the quality of the frames contained in lost packets, but also the quality of their subsequent frames. This is because of motion compensation employed by video codecs. Motion-compensated video codecs remove temporal redundancy in a video stream by encoding only pixel value difference (prediction error) between an image to be encoded (inter-frame) and a previously transmitted image (reference frame). Packet loss can introduce an error in a reference frame, which can be propagated to its subsequent frames and get amplified as more packets are lost.

Error propagation can be controlled by more frequently adding intra frames (which are coded temporally independently). However, the ratio of the compression efficiency of an intra-frame over an inter-frame is as large as 3 to 6 times. Increasing the frequency of intra-frames could increase the bandwidth requirement too much for video transmission over a bandwidth-constraint network. Nonetheless, the severe degradation of image quality due to error propagation has forced several popular video conferencing tools, such as `nv`[7], `vic`[13] and `CU-SeeMe`[6], to adopt an even more drastic approach. Using a technique called *conditional replenishment*, these tools filter out the blocks that have not changed much from the previous frame and intra-code the remaining blocks. Since all the coded blocks are temporally independent, packet loss affects only those frames contained in lost packets. However, this enhanced error resilience comes at the cost of low compression efficiency. Additional compression can always be obtained if temporal redundancy is removed from each coded block (i.e., by coding only their prediction error).

The goal is to develop an error recovery scheme that solves the error propagation problem without losing much compression efficiency. *Retransmission-based* and *forward error correction* schemes are the two major error control schemes found in the literature.

Retransmission-based error recovery (REC) can provide good error resilience without incurring much bandwidth overhead because packets are retransmitted only when some indications exist that they are lost. However, retransmission always involves additional transmission delay, and thus has been widely known ineffective for interactive real-time video applications such as video conferencing. Many researchers proposed use of extended control or playout times to allow retransmitted packets to arrive in time for display [21, 18, 12, 23]. This implies that the playout time of a frame is delayed by at least three one-way trip times after its initial transmission (two for packet transmissions and one for a retransmission request). Under the current Internet environment, this delay would be intolerable for interactive video applications, and can severely reduce the interactivity of the application.

Retransmission, however, can be still a very effective technique to improve error resilience in interactive real-time video conferencing. In this paper, we study a REC scheme, called *periodic temporal dependency distance* (PTDD), that can be used for interactive video applications. In particular, the schemes do not require any artificial extension of control time and play-out delays, and thus are suitable for interactive applications. In the scheme, frames are simply displayed at their normal playout times without any delay, as they are decoded. Thus, if a packet arrives after the playout time of its frame, the frame will be displayed with errors. However, this "late" packet can be used to remove error propagation. Rather than discarding the late packet, PTDD uses it to restore its frame although the frame is displayed. Because the frame is used as a reference frame for its succeeding frames, restoring the reference frame stops error propagation.

To allow enough time for retransmitted packets to arrive before their frames are referenced for the reconstruction of their dependent frames, the PTDD scheme extends the *temporal dependency distance* (TDD) of frames. The TDD of a frame is defined to be the minimum number of frame intervals (or inter-frame delay) between that frame and another frame on which it is temporally dependent. In the PTDD scheme, every p-th frame (which we call *periodic frame*) has an extended TDD while the other frames have TDD 1. Note that the extension of TDD does not affect the playout times of frames since the playout times of frames are determined solely by inter-frame delays and the initial control time which is used to reduce delay jitter. In the PTDD scheme, the TDD of periodic frames is determined by the estimated delay between the sender and the receiver.

In PTDD, non-periodic frames with TDD 1 are not protected by retransmission. If packet loss occurs for a non-periodic frame, the frame will be displayed with errors which can be propagated to its subsequent frames until the next periodic frame is received. The receiving video quality will periodically fluctuate. To remedy this, we apply a layered coding technique, called *quality assurance layering* (QAL) [22, 15, 8], for non-periodic frames which divides video signals into essential and enhancement signals. Essential signals are protected by a simple forward error correction (FEC) technique. Since the amount of essential signals is often much smaller than that of the entire signals, with only a little FEC redundant information, we can maintain relatively good video quality.

QAL is different from the technique adopted in `vic` [14] in that QAL uses motion compensation. In QAL, frames temporally depend only on the essential signals of their reference frames. This dependency ensures that the distortion of the enhancement signals of a frame does not affect the quality of its succeeding frames. Since essential signals are protected by FEC, this dependency will significantly reduce error propagation. However, QAL has its own limitations. First, since frames are motion-compensated only to the essential signals of their reference frames, the temporal redundancy present in the enhancement signals are not exploited at all, resulting in low compression efficiency. Second, under heavy packet loss, even essential signals can be lost, causing error propagation.

These limitations, however, can be greatly reduced when QAL is combined with PTDD. In this paper, we show that a (motion-compensated) layered coding technique such as QAL can be very effective for transmission over a lossy packet-switched network even when no network-level mechanism to protect a prioritized stream exists. When used with PTDD, the layering technique yields reasonably good compression efficiency since periodic frames temporally depends on both the essential and enhancement signals of their reference frames. This dependency is safe because retransmission can recover lost packets containing both signals. In addition, periodic frames reduce error propagation because their immediately succeeding non-periodic frames use them as a reference frame.

We conducted both transatlantic and transpacific Internet transmission tests from the East Coast of the US to show the utility of the REC-based techniques for transmission over a long distance network. The experimental results indicated that the proposed technique achieve good error resilience and compression efficiency.

In Section 2, we describe the related work, and in Section 3 we describe PTDD and QAL schemes. Section 4 contains a discussion of the experimental results, and Section 5 contains the conclusion.

2 Related Work

We first discuss the related work on retransmission-based techniques for video transmission over packet-switched networks and then discuss related work on layered video coding.

Retransmission-based error recovery. Dempsey et al.[5] applied retransmission for the recovery of audio packets. They showed that by adding some delay before the playout of each received audio packet, retransmission can be used to protect audio data from packet loss. Their work hinges on the earlier behavior study by Brady [3] showing that although less than 200 ms round trip delay is required for high quality voice applications, delays up to 600 ms can be tolerable by human ears.

Ramamurthy and Raychaudhuri [21] applied a similar technique to video transmission over ATM. They analyzed the performance of video transmission over an ATM network when both retransmission and error concealment are used to repair errors occurring from cell loss. They analytically showed that for a coast-to-coast

ATM connection, 33 ms to 66 ms play-out delay is sufficient to see a significant improvement in image quality.

Padopoulos and Parulkar [18] proposed an implementation of an ARQ scheme for continuous media transmission. Various techniques including selective repeat, retransmission expiration, and conditional retransmission are implemented inside a kernel. Their experiment over an ATM connection showed the effectiveness of their scheme.

Most recently, Li et al. [12] and Xu et al. [23] used retransmission in the recovery of lost packets for video multicasting. Li et al [12] proposed a novel scheme for distributing an MPEG-coded video over a best-effort network. By transmitting different frame types (I, P and B frames) of MPEG to different multicast groups, they implemented a simple layering mechanism in which a receiver can adjust frame play-out times during congestion by joining or leaving a multicast group. For instance, consider an MPEG picture pattern: IBBPBBPBBPBB. Their scheme delays the play-out times of frames for one frame interval. They call this delayed play-out time *adaptive playback point.* When a receiver leaves the B frame group, the adaptive playback point is additionally extended by three frame intervals because the next frame to be displayed after a P frame is at three frame intervals away. The scheme is shown effective for non-interactive real-time video applications.

In a video conference involving a large number of participants, different participants may have different service requirements. While some participants may require real-time interactions with other participants, others may simply want to watch or record the conference. Xu et al. [23] contended that retransmission can be effectively used for the transmission of high quality video to the receivers that do not need a real-time transfer of video data. They designed a new protocol called *structure-oriented resilient multicast* (STORM) in which senders and receivers collaborate to recover lost packets using a dynamic hierarchical tree structure.

Layered coding. Leicher [11] applied a forward error correction scheme known as *priority encoding transmission* (PET) [1] to a hierarchically encoded MPEG video. He used a temporal layering scheme in which reference frames (e.g., I-frames and P frames) are given higher priority than other temporally dependent frames (e.g., B frames). Since B frames temporally depend on P and I frames which are more reliably received, this technique can effectively suppress error propagation. However, in a low bit rate conferencing, the frequency of I and P frames must be kept very low because of their low compression efficiency. Thus, the resulting images can be very jerky as packets are being dropped to affect B frames.

Pancha and El Zarki applied a priority packetization scheme to an MPEG encoded video stream for transmission over ATM [17]. A frequency truncation technique is applied in which a fixed number of DCT coefficients of each DCT block are allocated to the HP data (cf. [16, 4, 20]). DeCleene et al. [4] also studied the performance of MPEG under various priority packetization techniques. It was shown that by utilizing priority packetization, the basic image quality can be maintained if the HP stream is guaranteed to be received. However, these techniques do not solve the error propagation problem because the essential signals of a frame still depend on both the essential and enhancement signals of its reference frame. Since the enhancement signals are less reliably transmitted, frames that depend on these signals also carry along the same error

Intra-H.261 is used in a popular Internet video conferencing tool called *vic* that encodes each frame as an intra-frame [13]. Using a conditional replenishment technique, the codec is shown to give excellent error resilience over packet loss. Another benefit of the codec is the reduced computation in encoding since it does not involve a motion prediction loop. A layering scheme for Intra-H.261 using a bit plane division technique is also discussed in [2, 14].

Quality assurance layering (QAL) was studied in [22, 15]. In this scheme, errors due to the loss of LP packets do not propagate because each frame is temporally dependent only on the essential signals of its reference frame which are assumed to be reliably received. Shimamura et al. [22] also studied the effect of various priority packetization techniques including frequency truncation.

Ghanbari [8] proposed similar layering techniques as QAL. In the techniques, each frame is first decimated to produce a low-resolution image, then the low-resolution image is coded using H.261 or a DCT-based codec and packetized as HP data. The original image is then compared with the decoded frame of its low-resolution frame to produce a difference image which is coded using a different codec and packetized as a LP data. This coding scheme will have similar error resilience as the QAL technique since the LP data depend only on the HP data. MPEG-2 adopts an approach similar to this.

Khansari et al. [10] applied QAL to video transmission over a mobile network to solve the "fading" problem commonly experienced during a mobile host hand-off period. By keeping the size of the HP stream large (about 83% of the total bandwidth), video quality even under fading can be kept relatively high. They also studied the effects of different packetizing techniques.

Priority layering techniques are also applied to still JPEG image transmission. Posnak et al. [20] used a frequency truncation layering technique that partitions DCT blocks of JPEG encoded frames into essential and enhancement layers. Han and Polyzos [9] also studied the effectiveness of layered coding through the hierarchical mode of JPEG and presented a statistical analysis showing that the overhead of the coding method can be insignificant.

3 Error Recovery Techniques

3.1 Retransmission-Based Error Control (REC)

Our schemes are based on a careful observation on how video frames are encoded in most motion compensation-based codecs. We base our discussion mostly on H.261

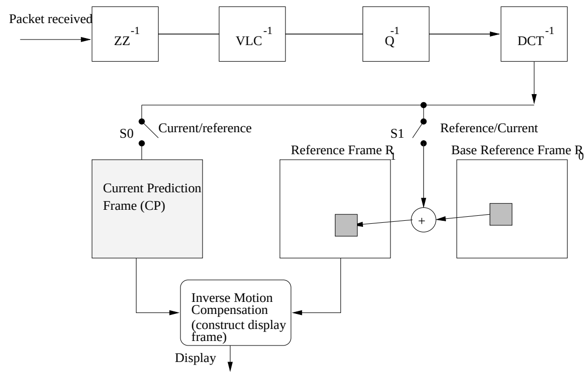

Figure 2: H.261 Decoder modified to handle the recovery of reference frames.

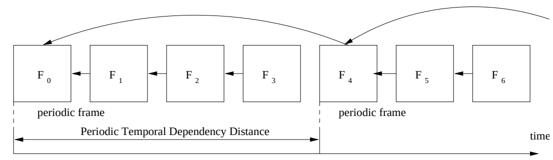

Figure 3: Periodic Temporal Dependency Distance

from which many motion compensation-based video standards such as MPEG are designed. In H.261, a video sequence consists of two types of video frames: *intra-frame* (I-frame) and *inter-frame* (P-frame). I-frame removes only spatial redundancy present in the frame. P-frame is encoded through motion estimation using another P-frame or I-frame as a reference frame (R-frame). For each image block in a P-frame, motion estimation finds a closely matching block within its R-frame, and generates the distance between the two matching blocks as a motion vector. The pixel value differences between the original P-frame and a motion-predicted image of the P-frame obtained by simply cut-and-pasting the matching image blocks from its R-frame are encoded along with the motion vectors.

Most of the previously proposed retransmission schemes work as follows. When a packet containing the encoding of a frame is lost at a receiver, the receiver detects the loss after receiving a subsequent packet of the lost packet and sends a retransmission request to the sender. After receiving the request, the sender retransmits the packet. We define the *display time* of a packet to be the time that the frame whose encoding is contained in the packet is displayed at the receiver. If the retransmitted packet arrives before its display time, the frame can be fully restored. Otherwise, it is discarded and the displayed image contains some error. All the subsequently decoded frames will carry the same error until a new I-frame is received.

Our scheme differs from others in that retransmitted packets arriving after their display times are not discarded but instead used to reduce error propagation. In motion compensation-based codecs, the correct image reconstruction of a currently displayed image depends on a successful reconstruction of its R-frames. The scheme allows that while a frames is being reconstructed, the "late" packets of an R-frame can be decoded and used for restoring the R-frame. This will stop error propagation because the next frame reconstructed would not carry over an error from the R-frame.

Figure 1 illustrates the error recovery using retransmission in a video stream containing two packets per frame. Packet 3 is lost, and the receiver receives packet 4 at time t_1 and recognizing that packet 3 is not received, sends a retransmission request (NACK) to the sender. The sender gets the NACK at time t_2 and retransmits packet 3. The retransmitted packet arrives at time t_3 which is before frame 3 is displayed. Packet 3 is now used to restore the R-frame of frame 3, so frame 3 can be decoded and displayed without an error.

Figure 2 shows a H.261 decoder modified to handle the recovery of R-frames using retransmitted packets. The only difference from the original H.261 decoder is one additional frame buffer added to handle the recovery. When a packet is received and decoded into an image block, the decoder determines whether the block belongs to the current frame being decoded or its R-frame. If it is for the current frame, then the block is stored into frame buffer CP along with its motion vector. If it is for the R-frame, the block is added with its temporally dependent block in frame buffer R_0 and stored into R_1. Note that CP contains only the prediction error and motion vectors of the current frame while R_1 contains the fully motion compensated image of the R-frame of the current frame, and R_0 contains the R-frame of R_1. We call R_0 a *base reference frame buffer*. At the next display time, the current frame is constructed using the information in CP and R_1. After displaying the current frame, R_1 is copied to R_0 and the displayed image are copied to R_1. In this scheme, as long as the packets belonging to R_1 arrive before the construction of the current frame, the packet can be used to help remove errors in the current frame. The *deadline of a packet* can be informally defined to be the arrival time of the packet at the receiver after which it is not useful for decoding any frame. Note that the decoder in Figure 2 extends the deadline of packets by one frame interval without delaying frame play-out times.

We can easily generalize the above scheme to extend packet deadlines beyond one frame interval. The periodic temporal dependency distance scheme (PTDD) generalizes the above scheme. (See Figure 3.) In the scheme, every i-th frame has an extended temporal dependency distance (TDD) i (we call this frame *a periodic frame*) while all the other inter-frames have TDD 1. The TDD of periodic frames in Figure 3 is 4. In fact, the pattern of the temporal dependency is very similar to a picture group pattern of MPEG. All the frames with TDD 4 can be regarded as P-frames while the other frames as B-frame (except the first frame). Thus, this scheme can be easily incorporated into MPEG. PTDD does not incur much computational overhead and does not require many additional frame buffers. Only two additional buffers are needed to store the R-frames of the next periodic frame.

One drawback of PTDD is that only periodic frames are protected by retransmission. An error in a non-

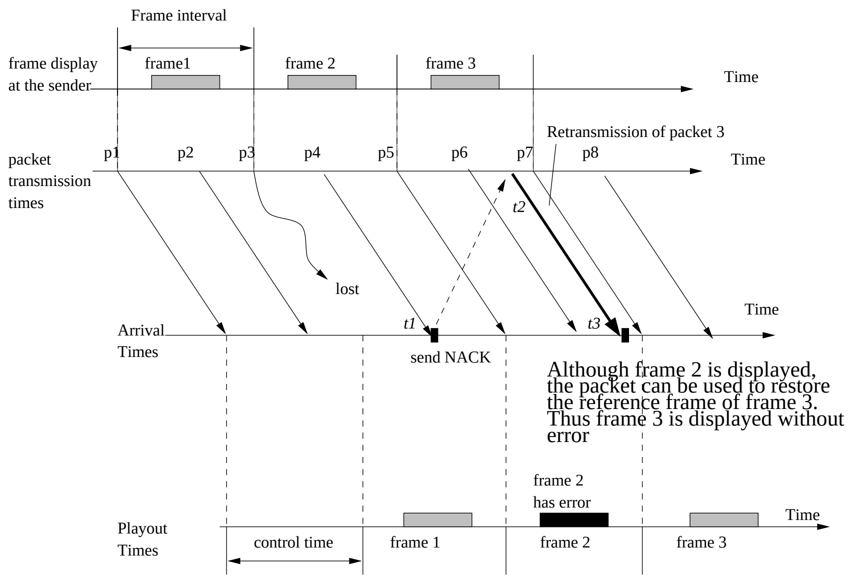

Figure 1: The recovery of frames using retransmission

periodic frame can propagate until the next periodic frame is received. In the next section, we discuss how we can overcome this drawback. In fact, there is an interesting tradeoff between the packet deadline and the extent of error propagation. A long TDD of period frames can prolong error propagation among non-periodic frames. However, it allows more time for periodic frames to be recovered. We plan to explore this tradeoff in the future.

3.2 Layered Coding with REC

The error resilience of non-periodic frames can be improved by employing a layered coding scheme that packetizes encoded frames into essential (or high priority) and enhancement (or low priority) signals. Although layered coding is originally developed for a network paradigm, such as ATM and RSVP, where a certain amount of bandwidth can be reserved, it can also be used over a best-effort network such as the Internet if the HP stream can be protected by forward error correction. The HP stream should be kept small to reduce the bandwidth required to protect the stream because the size of redundant information introduced by FEC is proportional to the size of the HP stream.

Quality assurance layering techniques guarantee a certain level of receiving video quality even though all the enhancement signals are lost. A version of quality assurance layering is shown in Figure 4. It is modified from H.261 and augmented with priority packetization and conditional replenishment. After DCT coefficients are quantized (Q), they are partitioned into HP and LP layers as described in Figure 5. For a fixed nonzero integer b less than 64, the first b coefficients are allocated to the HP layer, and the remaining coefficients are allocated to the LP layer. The subsequent inter-frame and the currently encoded frame are used to perform motion

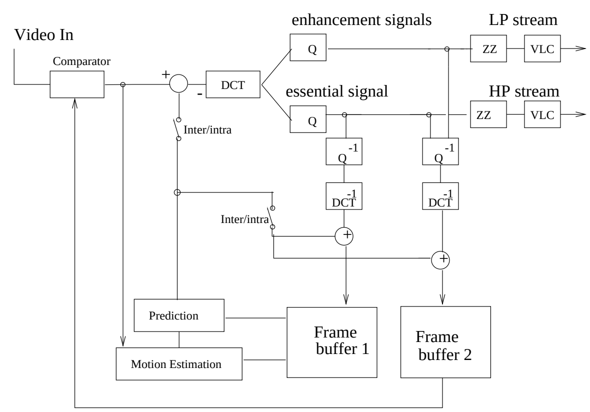

Figure 4: QAL Encoder

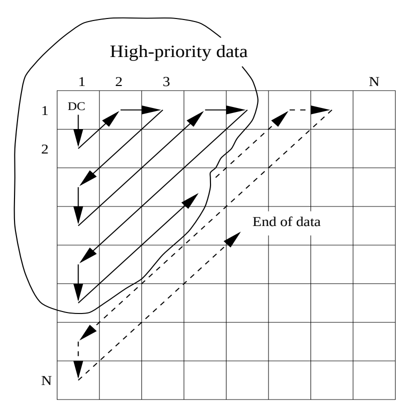

Figure 5: Partitioning of DCT Coefficients (from [16])

estimation and conditional replenishment to encode the

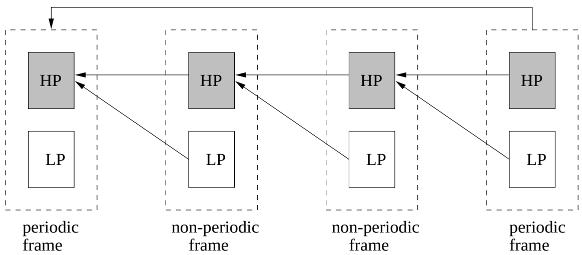

Figure 6: A Temporal Dependency Chain in Layered PTDD (PTDD +QAL)

next frame. Only the motion vectors and the HP coefficients of the current frame are used to reconstruct a predicted frame. The difference between the subsequent frame and the predicted frame is encoded. In the scheme, each frame temporally depends only on the essential signals of its reference frame. Since a frame is reconstructed from the essential signals of its reference frame, an error in the enhancement signals of the reference frame caused by packet loss does not carry over. Thus, even if all LP stream packets are discarded, a certain level of video quality can be maintained.

QAL, however, has not yet been proved to give good compression efficiency. The difficulty lies in the tradeoff between compression efficiency and the size of redundant information added by FEC. If the HP stream does not contain enough signals, there are not many temporally redundant signals between the reference frame reconstructed from the essential signals and the current frame to be encoded. On the other hand, as we add more coefficients to the HP stream, the HP stream gets larger, and so does the redundant information added by FEC to protect the HP stream. Due to low compression efficiency, QAL has been traditionally used in a situation where a large portion of bandwidth can be allocated to the HP stream (about 50% to 80% of the total bandwidth) [10]. Under the current Internet environment, this may not be reasonable.

PTDD provides a good compromise for these conflicting requirements. Since both the essential and enhancement signals of periodic frames can be restored by retransmission, we can allow periodic frames to exploit the temporal redundancy present in both of the essential and enhancement signals of their reference frames. This increases compression efficiency. Even with a small amount of bandwidth allocated to the HP stream, QAL combined with PTDD can achieve good compression efficiency. Note that non-periodic frames use only the essential signals of their reference frames including the ones immediately following periodic frames. So, when a periodic frame is displayed with some error due to packet loss in the LP stream, its dependent non-periodic frames would not be affected. Figure 6 illustrates a chain of temporal dependency in a layered PTDD scheme.

Under heavy or bursty packet loss, even the HP stream loses packets since FEC cannot provide adequate protection for the HP stream. Unfortunately, packet loss in the HP stream introduces errors in the essential signals of reference frames, causing error propagation. Thus, QAL alone cannot be very effective over a best effort network where no guarantee on the packet loss behavior can be made. However, QAL combined with PTDD effectively suppresses error propagation. The errors occurring in non-periodic frames due to loss in the HP stream do not propagate beyond the next periodic frame if the lost packets for the current periodic frame are recovered by retransmission before the reconstruction of the next periodic frame.

4 Experimental Result

The main objective of the experiment is to show that PTDD is an effective error control scheme for a real-time interactive video transmission over the Internet. We show this through video transmission experiments over transatlantic and transpacific Internet connections from Emory University, Atlanta, USA. By modifying an implementation of H.261, we implemented three variants of PTDD (HP.261, HPF.261, HPL.261), each of which differs from each other in the way that non-periodic frames are protected. HP.261 does not provide any protection for non-periodic frames; HPF.261 protects non-periodic frames by adding one Exclusive-OR parity packet (size 512 bytes) to each frame; and HPL.261 combines H.261 and QAL and it protects the HP stream by adding one Exclusive-OR parity packet (size 512 bytes) to each frame.

The performance of the three PTDD codecs is compared to that of several other codecs. H.261 is used as a base case for the comparison. H.261 is combined with a FEC technique where one Exclusive-OR parity packet (size 512 bytes) is added to each frame. We call this scheme HF.261. In addition, we implement QAL (without PTDD) based on H.261 as described in Figure 4. This implementation is called HL.261. We also implemented INTRA-H.261 which is adopted in `vic`. INTRA-H.261 is known for good error resilience over packet loss. For each frame, INTRA-H.261 intra-codes every image block changed significantly from the corresponding block in the previous frame. The remaining blocks are not coded. We also implemented INTRAL-H.261 which combines layered coding with INTRA-H.261 where the DCT coefficients of each coded block is divided into the HP and LP streams. Again, one parity packet is added to each frame to protect the HP stream. All the layered encoders code 5 DCT coefficients of each DCT block as essential signals and the remaining as enhancement signals.

4.1 Testing Methodology

A test video sequence is obtained from a typical video conferencing session where one typical "talking head" engages in a conversation with the other party. The video is sampled at 5 frames/sec rate and each frame is captured in the color CIF YUV format (352×288). This video sampling rate allows us to achieve a bit rate suitable for transatlantic and transpacific transmission without imposing too much load on the network. Considering the long distance and limited bandwidth between the testing sites, this frame rate is not unusual. The target bit rate is around 250 Kbits/sec. In addition to the controlled sample rate, we use conditional replenishment for all the tested schemes to obtain a desired bit rate. Adjusting quantization steps would be a more common way to control the bit rate. However, since INTRA-H261 uses conditional replenishment, we use the same technique uniformly for all the schemes for fairness. Finding the optimal video quality for a given bit rate is outside the scope of this paper.

About 40 second length video sequence (total 190 frames) is obtained. The video sequence is replayed several times for a five minute period for each experiment. The replay does not affect the integrity of the experiment because the first frame is always completely intra-coded (without any conditional replenishment). The 95th frame is intra-coded with conditional replenishment to remove any artifact due to the decoder drift effect. All the codecs applies a default quantization step size 8, and all the motion compensated codecs use the full exhaustive search over search window size 15 by 15. All PTDD codecs set the TDD of all the periodic frames to be 5 frame intervals. Given 5 frames/s frame rate, this TDD extends the deadlines of periodic frames up to 1 sec.

To see the compression efficiency of various codecs for the input test sequence, we measure the average peak signal-to-noise ratio (PSNR) of decoded frames over various data rates. The data rate is measured by the average number of bytes required to compress a frame (Figure 7). It is clearly shown that for a given data rate, INTRA-H.261 and INTRAL-H.261 show the worst video quality while H.261 shows the best. To achieve the same quality, INTRA-H.261 requires more bandwidth than the other motion compensation-based codecs. For instance, to obtain about 34 dB PSNR, INTRA-H.261 requires 80% (11KB/6KB) more bits per frame than H.261. It is known that more than 1 dB difference in video quality is clearly visible by human eyes. Although these measurements alone do not prove the low compression efficiency of INTRA-H.261, it clearly indicates one aspect of that.

HL.261 gives only slightly higher compression efficiency than INTRA-H.261. This is because the HP stream of HL.261 does not contain many coefficients, and only the essential signals of reference frames are used for motion compensation in HL.261. On the other hand, HPL.261 gives excellent compression efficiency. In HPL.261, although non-periodic frames are encoded in the same as in HL.261, periodic frames are motion-compensated to the full signals of its reference frame, resulting in higher compression efficiency.

Table 1 shows the chosen data rate of each codec for transmission along with the ratio of the bit rate of the HP stream over the total bit rate, and that of the redundant bit rate induced by FEC. HP.261 and HPL.261 are given a little lower data rates because the associated retransmission of lost packets may increase the actual data rate. Figure 8 shows the PSNR of each frame compressed by four different schemes under the target bit

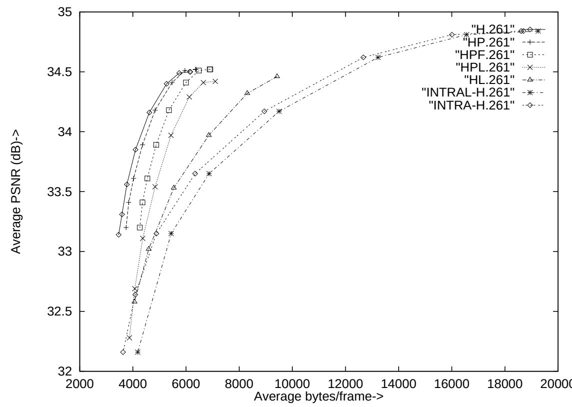

Figure 7: Compression Efficiency of Various Schemes

Compression scheme	Avg. bit rate (Kb/s)	FEC (%)	HP (%)	Avg. PSNR
H.261	240.6	0	0	34.50
HF.261	245.0	8.4	0	34.49
HP.261	232.6	0	0	34.51
HPF.261	234.2	8.6	0	34.41
HL.261	252	8	33	33.97
HPL.261	239.3	8.4	26	34.20
INTRA-H.261	247.7	0	0	33.65
INTRAL-H.261	252.7	8.1	27	33.49

Table 1: Chosen data rates for network experiments, and their average PSNR's

rate specified in Table 1. The video quality of HF.261 is similar to HP.261, and is not shown here. Note that the mean PSNR of HP.261 for the given data rates is 0.1 dB higher than that of H.261. This is due to the artifact of the encoder. In general, PTDD gives slightly lower compression efficiency than H.261 as shown in Figure 7.

The test video sequence is first compressed using each compression scheme and then packetized into approximately 512-byte packets. The packetized sequences of all the PTDD schemes are transmitted over the Internet. For each transmission test, we obtain a 5 minute trace that records the sequence numbers and arrival times of all the received packets. The transmission tests are conducted during the periods between Oct. 20

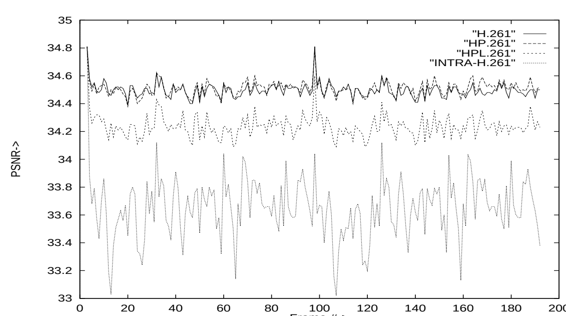

Figure 8: Video quality of encoded sequences

packet number	frame number	packet # in a frame	received Y/N	received time (ms)
0	1	0	Y	201
1	1	1	Y	450
2	1	2	Y	600
3	1	3	N	*
4	1	4	Y	980
5	2	0	Y	1100
6	2	1	Y	1305
7	3	0	N	*
8	3	1	Y	1790
9	3	2	N	*
10	3	3	Y	2100
11	4	0	N	*
12	4	1	N	*
13	5	0	Y	2710
14	5	1	Y	3000

(a)

packet number	frame number	packet # in a frame
0	1	0
1	1	1
2	1	2
3	1	3
4	1	4
5	2	0
6	3	1
7	2	2
8	2	3
9	3	0
10	3	1
11	3	2
12	3	3
13	4	0
14	4	1

(b)

packet number	frame number	packet # in a frame	received Y/N	received time (ms)
0	1	0	Y	201
1	1	1	Y	450
2	1	2	Y	600
3	1	3	N	*
4	1	4	Y	980
5	2	0	Y	1100
6	3	1	Y	1305
7	2	2	N	*
8	2	3	Y	1790
9	3	0	N	*
10	3	1	Y	2100
11	3	2	N	*
12	3	3	N	*
13	4	0	Y	2710
14	4	1	Y	3000

(c)

Figure 9: (c) is the result of mapping an actual trace (a) to a packet sequence (b).

to Oct. 27, 1997, and between Jan 30 to Feb 6, 1998.

Each packet of a frame is transmitted at a regular interval determined by the 5 frames/s frame rate and the number of packets within the frame. For example, for the frame interval of 200 ms, if one frame consists of 10 packets, a packet in the frame is transmitted at 20 ms interval. Each transmitted packet is assigned a unique sequence number. Retransmitted packets are given the same sequence numbers as their original packets.

The ARQ scheme employed during the experiment works as follows. The receiver sends one acknowledgment to the sender for each received frame. An acknowledgment contains information about the missing packets of the last periodic frame that the receiver received. After retransmitting a packet, the sender does not retransmit the same packet for about three frame intervals. It may retransmit the packet if it receives another acknowledgment after the period indicating that the packet is lost. The receiver also does not request for the retransmission of packets whose deadlines are expired. These mechanisms reduce the number of unnecessary retransmissions.

Each trace is fed to an off-line decoder to measure the signal-to-noise of the received frames. To simplify the experiment, we did not add any jitter control time for frame play-out. Each frame is considered to be displayed at the arrival of the first packet of its next frame if that packet is received. If that packet is not received, the frame is considered to be displayed at 200 ms after its previous frame's play-out time. If no packet for the frame is received, any frame displayed last will be displayed for that frame. Retransmitted packets are not used for the display of their frames, but used only to restore their corresponding reference frames.

For fair comparison, we run trace-driven simulations by mapping each of the traces T to the packetized sequence of H.261, HL.261, INTRAL-H.261 and INTRA-H.261 as follows. We first obtain a 5 minute length of a packetized sequence S for those schemes as if the sequence would have been transmitted in a real test. Each packet p in trace T is mapped to a packet q that has the same sequence number as p. If packet p is received, we record q as received and assign the receiving time of p to q. Otherwise, we record q as lost. Figure 9 illustrates this mapping. Figure 9(a) shows a sample trace of a HP.261 sequence ('$*$' indicates the packet is not received). Those packets received (indicated by Y) show received times. Figure 9 (b) shows a sample of a packetized H.261 sequence. Figure 9 (c) shows the result of mapping (a) to (b). This mapping technique provides a very accurate comparison of various transmission schemes because the sequences of all the schemes are mapped to the same traces.

This mapping technique cannot capture the dynamics of acknowledgments and retransmissions in HP.261, HPL.261 and HPF.261. This is because in the experiment, the acknowledgment is transmitted by the receiver only when a new frame is received and each sequence may have a different number of packets in a frame. Thus, acknowledgments and retransmitted packets might be received at different times for different sequences. The dynamics can only be captured through a real transmission, which is why the sequences of HP.261, HPF.261, and HPL.261 are actually transmitted. The other schemes do not have this problem because they do not involve any retransmission.

We obtained 168 traces of HP.261, 147 traces of HPF.261, and 55 traces for HPL.261 Since many traces are obtained, it is difficult to present the result of each trace independently. So we classify the traces into several loss rate groups and present only the average behavior of the traces in each group. Table 2 shows the loss groups and their corresponding loss ranges. Since high loss cases are relatively infrequent, we set a larger range for high loss rates.

Loss group	0.025	0.05	0.075	0.1	0.125	0.15
Loss range	(0, 0.025)	[0.025,0.05)	[0.05,0.075)	[0.075,0.1)	[0.1, 0.125)	[0.125, 0.15)
Loss group	0.175	0.2	0.25	0.3	0.35	0.4
Loss range	[0.15, 0.175)	[0.175,0.2)	[0.2, 0.25)	[0.25,0.30)	[0.3, 0.35)	[0.35,0.40)

Table 2: Loss Rate Groups and their loss ranges

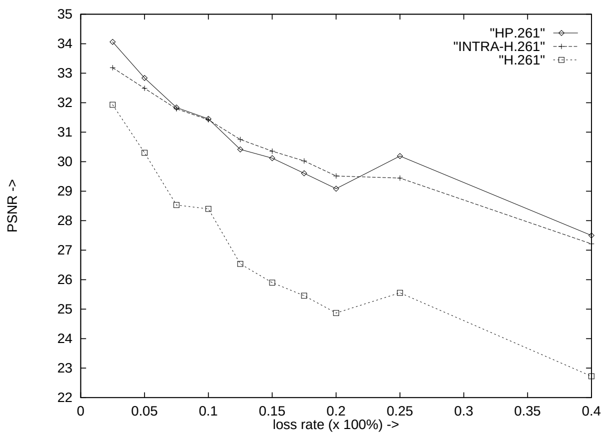

Figure 10: Mean PSNR of H.261, INTRA-H.261 and HP.261

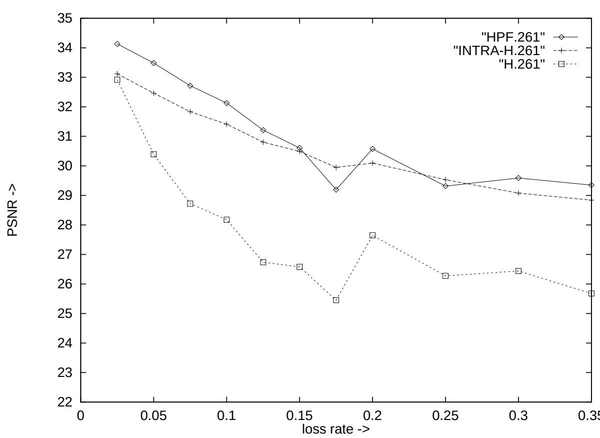

Figure 11: Mean PSNR's of H.261, INTRA-H.261, and HPF.261

4.2 Performance of PTDD (HP.261), and PTDD + FEC (HPF261)

In this section, we report the results obtained from the HP.261 and HPF.261 traces. The sequences of H.261, HF.261 and INTRA-H.261 are mapped to the traces of HP.261 and HPF.261. Figures 10 and 11 show the average PSNR's of H.261, INTRA-H.261 and HP.261 for various loss groups. Table 3 summarizes the result of H.261, HF.261, HPF.261, and INTRA-H.261.

In Figures 10 and 11, the mean PSNR of H.261 drops drastically even under small packet loss, showing the effect of error propagation. INTRA-H.261, HP.261, and HPF.261 exhibit generally good error resilience. They all show similar PSNRs for all loss groups. Between 12% and 20% packet loss, the mean PSNR of HP.261 drops below that of INTRA-H.261. This is due to the drop in the REC recovery rates.

In Figure 11, we can see that the PSNR of HPF.261 is slightly improved from that of HP.261. HPF.261 gives slightly better PSNR than INTRA-H.261 for all loss groups except two. This improvement is mainly due to FEC. However, the improvement is still marginal. Although many packets can be recovered through FEC under small packet loss, as packet loss gets severer, FEC becomes ineffective. From Table 3, under less than 10% packet loss, FEC could recover from 30% to 50% packet lost. However, as more packets are dropped, the FEC recovery rate drops below 10%.

There is a clear correlation between the round trip time and REC recovery rates. As the round trip time delays increase, the recovery rates of periodic frames also drop. When round trip times increase beyond 250ms, the recovery rates by REC are significantly reduced. This is because the increased network delay reduces the probability for retransmitted packets to be received before their deadlines. Second, in HP.261, there is no protection for non-periodic frames against packet loss, and in HPF.261, the protection by FEC was not so helpful in recovering lost packets for non-periodic frames. Thus, as more packets are lost, more non-periodic frames suffer from error propagation.

Overall, although the HP.261 and HPF.261 show better error resilience over INTRA-h.261, the improvement can be considered marginal. This is because in both schemes, little protection is provided for non-periodic frames. Packet loss in a non-periodic frame starts error propagation which is continued until the next periodic frame is received. The resulting video quality fluctuates substantially. This problem disappears when PTDD is combined with a better FEC scheme such as QAL.

4.3 Performance of PTDD + QAL + FEC (HPL.261)

In this section, we report the result obtained from the HPL.261 traces. HPL.261 combines PTDD and QAL. The HP stream generated from QAL is protected by FEC. We ran trace-driven simulations of H.261, HL.261, INTRA-H.261, and INTRAL-H.261 based on the traces of HPL.261. Table 4 summarizes the result. Figure 12 shows the average PSNR's of various schemes. We do not have traces for loss groups higher than 20%.

In Figure 12, HPL.261 shows good error resilience, showing clear separation (about 1 dB) from INTRA-H.261. 1 dB or higher PSNR difference in video quality is clearly visible by human eyes. The good performance of HPL.261 can be best explained from Figure 13. As

Loss Rate (%)	# of traces	H.261		HPF.261					INTRA-H.261		HF.261	
		PSNR (dB)	D. R. (Kb/s)	PSNR (dB)	D. R. (Kb/s)	REC(%)	FEC(%)	RTT (ms)	PSNR (dB)	D. R. (Kb/s)	PSNR (db)	D.R. (Kb/s)
2.5	24	32.91	247.0	34.16	247.9	89.7	48.88	188.5	33.12	259.0	33.62	246.4
5.0	42	30.39	246.5	33.48	249.1	96.21	42.23	197.7	32.46	258.5	31.66	246.0
7.5	32	28.72	245.3	32.71	249.6	96.64	33.77	223.5	31.83	257.3	30.07	244.8
10.0	14	28.17	245.7	32.13	250.8	84.56	24.46	239.1	31.41	257. 6	29.24	245.1
12.5	14	26.73	249.7	31.21	261.9	81.70	18.77	325.1	30.80	259. 7	27.55	249.5
15.0	9	26.58	249.5	30.60	263.3	57.94	11.78	388.1	30.4	259.0	27.26	249.9
17.5	6	25.45	250.2	29.19	263.7	50.74	9.34	437.7	29.94	259.3	25.91	250.6
25.0	5	26.27	250	29.31	261.7	41.9	5.11	359.3	29.53	260.1	2 6.69	249.8
35.0	1	25.68	239.8	29.35	245.4	24.96	6.14	243.5	28.84	251.5	26.9	239.3

Table 3: Experimental Data based on HPF.261 traces

Loss Rate (%)	# of traces	INTRA-H.261	INTRAL-H.261		H.261	HPL.261				HL.261	
		D.R. (Kb/s)	D. R. (Kb/s)	FEC %	D.R. (Kb/s)	D.R. (Kb/s)	REC %	FEC %	RTT (ms)	D.R. (Kb/s)	FEC %
2.5	22	249	243	60.7	243	238	89.9	61.3	242.6	243	63.5
5	16	247	240	59.8	240.8	239	78.0	57.8	280.4	238	57.9
7.5	7	249	242	52.4	242	240	73.2	52.1	310.5	240	54.4
10	3	249	241	35.5	240	240	57	39.1	305.6	239	41.7
12.5	2	249	243	48.5	242	245	84.4	48.7	347	244	54.5

Table 4: Experimental Data based on HPL.261 traces

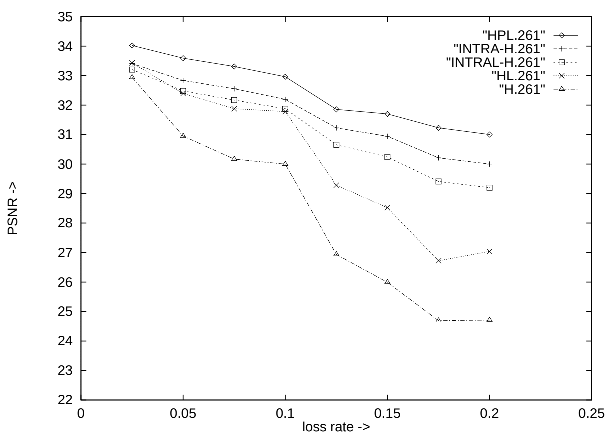

Figure 12: Mean PSNR's of H.261, HL.261, HPL.261, INTRA-H.261, and INTRAL-H.261

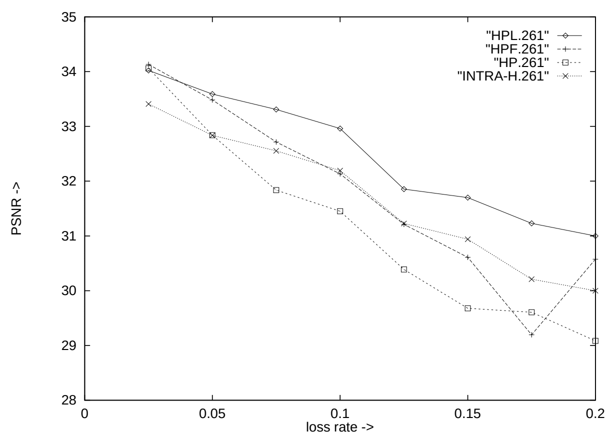

Figure 14: PSNR vs. Loss Rate Comparison of HP.261, HPF.261, and HPL.261

in HP.261, the PSNR of HPL.261 drops when there is heavy loss for a frame. However, it quickly bounces back when the loss rate of the subsequent frames quickly reduces. This is unlike HP.261 where rebound happens mostly around periodic frames. We clearly see many plateaus indicating that packet loss does not have much impact on the video quality of non-periodic frames.

HL.261 performs quite well under low packet loss. As packet loss gets substantial, The PSNR of HL.261 quickly drops. This is because beyond low loss, the FEC scheme protecting the HP stream becomes ineffective and the HP stream starts losing packets causing error propagation.

Surprisingly, INTRAL-H.261 performs slightly worse than INTRA-H.261. This can be explained in two ways. First, the HP stream contains have only 5 DCT coefficients from each coded block. These coefficients do not contain enough information to improve the receiving video quality even though they are successfully received. Second, the compression efficiency of INTRAL-H.261 is lower than that of INTRA-H.261. Thus, under no or little loss, its average PSNR is always a bit lower than that of INTRA-H.261. The PSNR difference between INTRAL-H.261 and INTRA-H.261 over various loss groups seems to remain constant because of these two reasons.

We also compare the mean PSNRs of all the PTDD schemes presented in this paper in Figure 14. We use the mean PSNR of INTRA-H.261 obtained from the traces of HPL.261 as a reference for comparison. From the figure, we can see that HPL.261 seems to perform best. At the beginning, when there is no or little loss, HPL.261 gives the worst PSNR among REC-based codecs. However, as packet loss gets larger, HP.261 and HPF.261

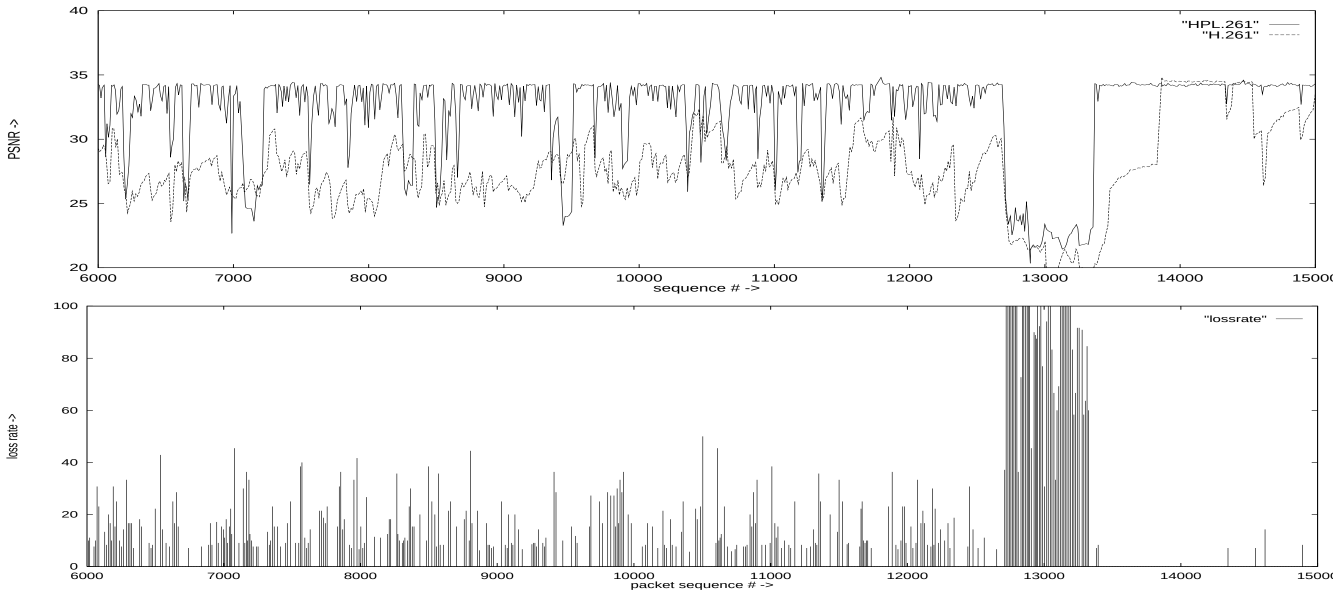

Figure 13: One trace of HPL.261 with 10% packet loss – PSNR, and Loss and Recovery Rates

quickly drops while HPL.261 sustains good video quality.

5 Discussion

Retransmission has widely been known ineffective for recovering lost packets in real-time interactive video applications. This paper challenges the conventional wisdom and presents new retransmission-based error control techniques that do not incur any additional latency in the frame play-out time, and hence are suitable for interactive applications.

One main implication of our work is that many motion compensation prediction-based codecs, such as MPEG, and H.261, are still useful for Internet interactive video transmission over a lossy network. Some of the disadvantages of the motion compensated codecs cited in the literature [14, 13] include (1) computational complexity, (2) error resilience, (3) tight coupling between the prediction state at the encoder and that at the decoder, and (4) compute-scalable decoding. In this paper, we showed that the H.261 equipped with our REC schemes achieves comparable error resilience to that of INTRA-H.261. We also believe that some of the other disadvantages can be overcome with a simple modification to the codecs. For instance, the compute-scalable decoding can also be achieved by decoding only periodic frames and shedding off the computational load for decoding non-periodic frames in PTDD.

Having said some of the disadvantages of motion-compensated codecs, we would like to emphasize one of their advantages over INTRA-H.261, which is high compression efficiency. Although INTRA-H.261 gives good error resilience, the low compression efficiency of INTRA-H.261 will make it very difficult to obtain very high quality video for a low bit rate transmission. As pointed out in [23], in a multicast group, while some receivers want real-time video, others may want to watch or record the transmitted video. These observers may want the highest quality that the video source could provide although they can tolerate a longer play-out delay. At the same time, they may have only a small amount of bandwidth allocated for the video.

Motion-compensated encoding allows much better video quality for a given data rate than intra-coding. Our scheme allows real-time receivers to view video in a comparable quality as intra-coding schemes. At the same time, it allows the same encoded video to be multicasted to other non-real-time receivers. These receivers can view the video in very high quality by adding an additional delay (i.e., control time) before the display of the first frame. On the other hand, for intra-coded video to be sent on a low bandwidth network, its quality needs to be reduced substantially. In this case, although real-time receivers may get the video in similar quality as the motion-compensation scheme with REC, non-real-time receivers will receive a poor quality video. Note that this feature is different from media scaling [14] where receivers with a higher network capacity always get a higher quality image. Here, this feature allows even the receivers with a low network capacity to get a high quality while trading interact-ability. The motion compensated codecs equipped with a REC scheme can provide this feature as they generally give better compression efficiency and lost packets can be recovered through retransmission.

References

[1] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, “Priority Encoding Transmission,” IEEE Transactions on Information Theory, vol. 42, no. 6, Nov. 1996.

[2] Amir, E., McCanne, S., and Vetterli, M. "A Layered DCT Coder for Internet Video," *the Proceedings of IEEE International Conference on Image Processing. September*, 1996. Lausanne, Switzerland.

[3] P. Brady, "Effects of Transmission Delay on Conversational Behavior on Echo-Free Telephone Circuits," *Bell System Technical Journal*, vol. 50, no. 1, pp. 115 – 134, Jan. 1993.

[4] B. DeCleene, P. Pancha, M. El Zarki, and H. Sorensen, "Comparison of Priority Partition Methods for VBR MPEG," *Proceedings of IEEE Infocom'94, Toronto,* Canada, June 1994.

[5] B. Demsey, J. Liebeherr, and A. Weaver, "On retransmission-based error control for continuous media traffic in packet-switching networks," *Computer Networks and ISDN Systems,* 1996.

[6] T. Dorcey, "CU-SeeMe desktop videoconferencing software," *ConneXions,* vol. 9, no. 3, Mar. 1995.

[7] R. Fredrick, *Network Video(nv)*, Xerox Palo Alto Research Center.

[8] M. Ghanbari, "Two layer coding of video signals for VBR networks," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 5, June 1989.

[9] K. Han and G. C. Polyzos, "Multi-Resolution Layered Coding for Real-Time Image Transmission: Architectural and Error Control Considerations," Real-Time Imaging," *Academic Press* (to appear).

[10] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit-rate video transmission over fading channels for wireless microcellular systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 1-11, February 1996.

[11] C. Leicher, "Hierarchical encoding of MPEG sequences using priority encoding transmission (PET)," ICSI technical report TR-94-058, Nov. 1994.

[12] X. Li, S. Paul, P. Pancha, and M. Ammar, "Layered video multicast with retransmission (LVMR): evaluation of error recovery schemes," *Proceedings of the Sixth International Workshop on Network and Operating System Support for Digital Audio and Video,* St Louis, May 1997.

[13] S. McCanne and V. Jacobson, "*vic*: a flexible framework for packet video," in *Proceedings of ACM Multimedia'95*, San Francisco, CA, Nov. 1995, pp. 511–522

[14] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 6, pp. 983-1001, August 1997

[15] M. Normura, T. Fujii, N. Ohta, "Layered Packet-Loss Protection for Variable Rate Coding Using DCT," *Proceedings of International Workshop on Packet Video*, D7, Sept. 1988.

[16] N. Ohta, "Packet Video: Modeling and Signal Processing," Artech House Publishers, 1994.

[17] P. Pancha and M. El Zarki,"MPEG Coded Video and BISDN," *IEEE Communications Magazine*, Vol. 32, No. 5, May 1994.

[18] C. Papadopoulos and G. Parulkar, "Retransmission-based error control for continuous media applications," *Proceedings of the Sixth International Workshop on Network and Operating System Support for Digital Audio and Video,* pp. 5–12, 1996.

[19] Priority Encoding Transmission, Web Page: http://www.icsi.berkeley.edu/PET/icsi-pet.html.

[20] E. J. Posnak, S. P. Gallindo, A. P. Stephens and H. M. Vin. "Techniques for Resilient Transmission of JPEG Video Streams" *the Proceedings of Multimedia Computing and Networking*, San Jose, Volume 2417, Pages 243-252, February 1995.

[21] G. Ramamurthy, and D. Raychaudhuri, "Performance of packet video with combined error recovery and concealment," *Proceedings of INFOCOM'95*, pp. 753–761, April 1995.

[22] K. Shimamura, Y. Hayashi, and F. Kishino, "Variable-bit-rate Coding Capable of Compensating for Packet Loss," *Proceedings of Visual Communications and Image Processing,* Nov. 1988, pp. 991 – 998.

[23] R. Xu, C. Myers, H. Zhang, R. Yavatkar, "Resilient multicast support for continuous-media applications," *Proceedings of the Sixth International Workshop on Network and Operating System Support for Digital Audio and Video,* St Louis, May 1997.