ERROR-RESILIENT REAL-TIME VIDEO COMMUNICATIONS OVER IP-BASED LANS

Guo-Shiang Ma and Ming-Syan Chen[*]

Electrical Engineering Department
National Taiwan University
Taipei, Taiwan, ROC

Jeng-Chun Chen[†]

Philips Research East Asia[†]
Philips Coorporation
Taipei, Taiwan, ROC

ABSTRACT

Video communication over IP networks involves more issues than that over ATM networks in terms of error control. First, IP networks provide neither prioritized transmission nor guaranteed QOS for data packets. Second, video streams need to be packetized well before transmission over IP networks in order to have good error resilience characteristics. Finally, the mitigation of spatial and temporal error propagation in IP networks is very different from that in ATM networks upon the loss of IP packets. In this paper, we develop an error-resilient video communication system based on the H.323 family of ITU-T recommendations. With a low-delay backward channel for packet receipt reports, bi-directional error resilience mechanisms are employed to avoid error spreading in the temporal domain. Two operation modes, i.e., ACK and NACK, are devised for two communicating parties to eliminate error propagation after either one frame lag or one round trip delay without consuming much additional bandwidth as compared to simple INTRA updates. Our system can also adaptively choose an operation mode to maintain a satisfactory video quality at the decoder under variable channel error conditions. It is shown by experimental results that our error control scheme is suitable for video transmission over IP-based LANs at packet loss rates up to 20%.

1. INTRODUCTION

The Internet is in essence a packet-switched network with variable bandwidth, variable transmission delay, and variable error characteristics. Therefore, real-time video communication in the Internet needs to take into consideration various packet loss conditions. Particularly, error resiliency in video applications via error detection, recovery, and concealment is the key to satisfactory video quality. There has been an extensive literature on error control and concealment for video communication, especially over ATM networks. However, not much focus has been placed upon error resiliency in real-time video communication over IP networks. Readers are referred to [7] for a discussion of the differences between circuit-switched and packet-based videoconferencing solutions.

There has been a lot of research on error concealment at the decoder over ATM networks [6]. However, most postprocessing techniques could fail if a single packet contains rows of macroblocks. Note that transmitting a lot of small IP packets over the Internet consumes a large amount of bandwidth in packet headers. Therefore, this trade-off makes ordinary error concealment mechanisms less effective, which in turn leads our system design to incorporate feedback-based error control. As will be seen in our performance study later, utilizing a backward channel can effectively maintain the visual quality to a satisfactory level.

The encoder can encode input video sequences according to variable channel error situations when a separate backward channel is available for the decoder to send packet loss reports. This channel can usually be easily established and maintained in a small videoconference. The most attractive feature is that error propagation can be completely removed soon at the decoder even without succeeding full INTRA-coded frames. If enough memory resources are available at the encoder, it can encode frames using some previous correctly decoded frames for reference, thus removing error propagation at the decoder. This technique has two operation modes, i.e., NACK and ACK modes. As will be seen later, the NACK mode is preferred when packet loss rate is low since it places little burden on the output bandwidth. The ACK mode works better at high packet loss rates since a single packet loss only affects one frame. Note that the ACK mode generally consumes more bandwidth since acknowledged frames rather than adjacent frames are used for reference. Thus, the codec is recommended to use a coarser quantizer if rate control is required to meet a certain bandwidth requirement.

In our developed system, we set the focus on the transport level control and the selection of reference pictures. H.261 video streams are packetized on a GOB basis in our modified RFC 2032 [2] scheme. Then, the RTP module takes care of the transmission of these video data packets. A simple temporal error concealment technique is employed at the decoder to conceal packet errors with a short processing delay. We implement bi-directional error control mechanisms into our H.323 video conferencing software to deal with packet error and error propagation. Our error resilience mechanism establishes an out-of-band data channel that serves as a backward channel for the decoder to send packet receipt reports. This connection-oriented channel is assumed to have a short delay; otherwise late backward reports would lose their value to the encoder. To suppress error propagation while maintaining coding efficiency which refers to the efficiency of compression, our system is able to switch between two operation modes in the presence of variable network conditions. We then proceed to observe the effectiveness in the maintenance of picture quality without much additional bandwidth as required by full INTRA-frame refresh.

The contribution of this paper includes the design of error-resilient mechanisms and the performance analysis of the NACK and ACK operation modes. In our prototype model, we adopted the design concept of the Reference Picture Selection (RPS) option of H.263+ [5]. We use PSNR and bit rates to evaluate the performance of both operation modes at different packet loss rates. Our experimental results have shown that the NACK mode is appropriate at low packet loss rates since it requires less memory resources and consumes less bandwidth. However, it is sensitive to packet loss and it could fail when encountering successive, bursty packet losses. The ACK mode is very insensitive to packet loss because a packet error only affects a single frame. We have observed that the visual quality can be maintained in the ACK mode even at high packet loss rates up to 20%. Clearly, if some other performance

[*] The corresponding author of this paper.

measures are of interest, our system then can evaluate the performance of both modes and make proper switches between them accordingly.

This paper is organized as follows. Details of our system model are described in Section 2. We present experimental results and performance evaluation in Section 3. Conclusions are drawn in Section 4.

2. DESIGN OF ERROR-RESILIENT VIDEO COMMUNICATION

The checksum filed in the IP header ensures the integrity of the IP packet, providing protection against bit errors. Thus, forward error correction (FEC) that consumes bandwidth is not justifiable on IP networks. IP-based LANs are characterized as having adequate bandwidth for video applications and providing a bit-error-free environment. In this study, we assume a maximum packet loss rate of 20%. To validate our error-resilient design for video communication, we develop an ITU-T H.323 [4] compliant videoconference application with a focus on highly interactive, but small video conferences. To boost error resiliency, this software application takes advantage of two-way communication mechanisms such as acknowledgments of correctly received video packets via backward channel messages.

Packet losses can severely degrade the quality of future decoded pictures at the decoder since an IP packet usually carries an integer number of MB rows. Obviously it is not enough to just apply post-processing techniques for error concealment. The ITU Recommendation H.245 [3] allows the encoder and decoder to build an out-of-band channel on which the decoder can return packet loss information. To exploit this feature, our design makes good use of a connection-oriented data channel. Upon backward channel messages sent from the decoder, the encoder can take into account the packet loss and encode either severely damaged MBs in INTRA mode or less affected MBs in INTER mode.

2.1. H.323 Video Conference System (VCS)

H.323 Video Conference System (VCS) is our video-conference software for video conferencing between two parties by courtesy of Philips Innovation Center, Taipei (PICT). In our joint research project, VCS has served as an excellent testbed because of its full H.323 functionality. There are seven major subsystems in this H.323 VCS software: Session Management, Q.931 Management, H.245 Management, Channel Management, RAS Management, RTP Management, and Socket Management.

As shown in Figure 1, the Data Channel is used as a separate backward channel for interactive error concealment. Since it is built upon TCP sockets, all backward messages can reach the encoder in order and remain intact. The Session Management is responsible for passing these messages to its forward video channel. Then received back-channel messages are all processed by the encoder before a new input video frame is to be encoded. The encoder can both free useless frame buffers on ACK messages and use the decoder's requested frame for reference on NACK messages.

2.2. Coding Control for NACK mode

If no packet loss is present during video transmission, the encoder can always successfully encode data with the expected coding efficiency. However, if the encoder does not know some packets did not reach the other end, subsequent decoded frames can suffer from a degradation in visual quality because of predictive coding. With NACK or ACK messages from the decoder, the encoder can react to them right away to eliminate error propagation at the decoder.

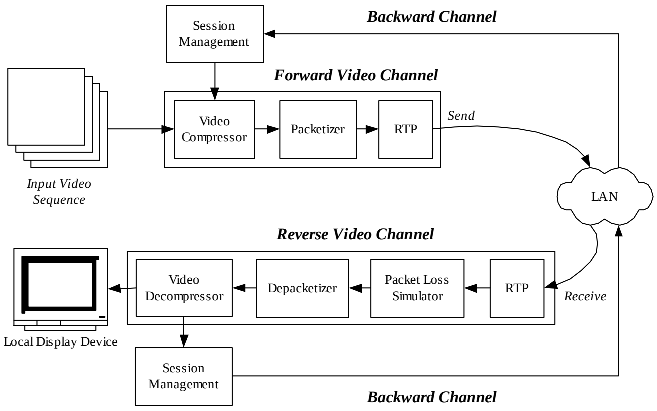

Figure 1: Video transmission model of VCS

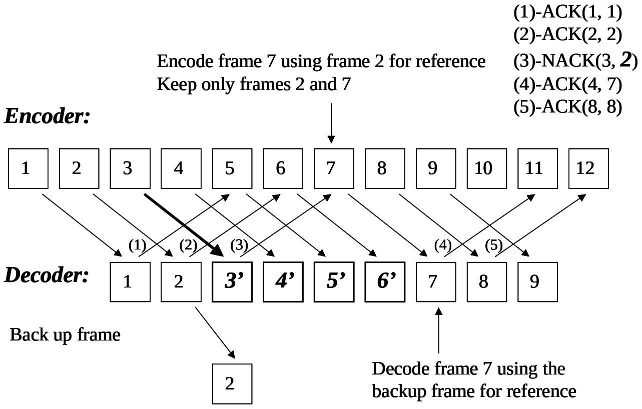

Figure 2: Illustration of operation in the NACK mode

When operating in the NACK mode without any packet loss, the decoder keeps sending backward acknowledgment messages for the encoder to maintain frame buffers. By checking the TR (Temporal Reference) field in an ACK message, the encoder can then safely release frame buffers prior to the acknowledged frame. The best coding efficiency is maintained since the time lag between the reference frame and the current one is always one frame interval. After a packet loss, the decoder sends an NACK with the RTR (Requested Temporal Reference) field set to the temporal reference of some previous correctly decoded frame kept at the decoder. Upon the receipt of this NACK message, the encoder then encodes data using the decoder's requested frame for reference. The TRP (Temporal Reference for Prediction) field in our modified RFC 2032 header is set to the RTR value in this NACK message to inform the decoder which backup frame is used for reference.

When a transmission error occurs, the decoder sends through the backward channel a negative acknowledgment message to inform the encoder. Consider the execution scenario in Figure 2 as an example. In Figure 2, assume that frame 3 cannot be correctly decoded. The decoder then backs up frame 2 into its additional frame buffer and sends backward NACK(3, 2) requesting the encoder to use frame 2 for reference. On receipt of an NACK for frame 3 before the encoding of frame 7, the encoder can therefore use frame 2 for reference and free the memory resources for frame

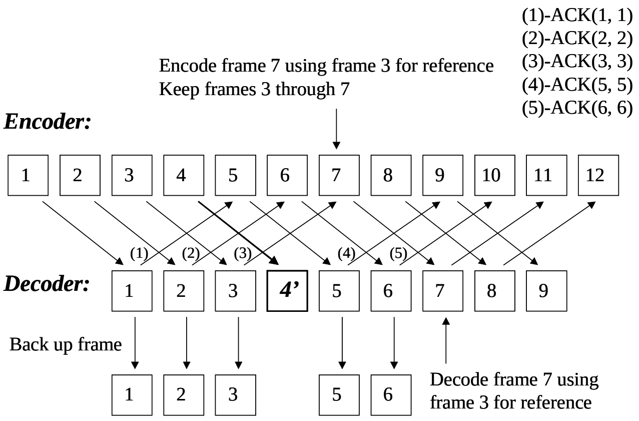

Figure 3: Illustration of operation in the ACK mode

3 through frame 6 since these frames are all corrupt at the decoder. Until the encoded data of frame 7 arrives, the decoder suffers from using inconsistent frames for reference for the period of one round trip delay. Note that the decoder requires only one additional frame buffer in NACK operation mode. If the round trip delay is short, which is assumed to be one of the error characteristics of LANs, the impact of packet losses at the decoder can be removed after an acceptable period of time. The advantage of this mechanism over simple INTRA updates lies in the increased coding efficiency.

2.3. Coding Control for ACK mode

When operated in the ACK mode, the decoder sends ACK messages to acknowledge all correctly decoded frames and the encoder uses for reference only the requested frame indicated in the back-channel message. The coding performance is lower even when no transmission errors occur since the time lag between the reference frame and the current frame is more than one round trip delay. However, error propagation can be avoided entirely since only acknowledged frames are used for reference.

Consider the execution scenario in Figure 3 as an example. The decoder acknowledges and backs up every correctly decoded frame until an error occurs to frame 4. Since frame 5 is encoded using frame 1 for reference, the decoder can still correctly decode frame 5 in reference to its backup frame 1 without any error propagated from corrupt frame 4. Note that no succeeding frames would be in reference to frame 4. The decoder can thus avoid error propagation after the error to frame 4 entirely. When the decoder receives data for frame 6, it can safely free the frame buffer for frame 1 and use frame 2 for reference. To reduce the number of additional frame buffer at the decoder by half, every two ACK messages could request the same previous reference frame. It should be noted that enough previous frames at the encoder are required to cover the maximum round trip delay of NACK and ACK messages. However, the number of additional frame buffers at the decoder can be reduced to one in the NACK mode. Storage reduction by half is possible in the ACK mode at the cost of slightly increased bit rates.

3. PERFORMANCE ANALYSIS

3.1. System Model for Experiments

To perform our experiments, we modified the public-domain PVRG-P64 Codec 1.1 [1] to incorporate error concealment and resilience mechanisms. We chose an H.261 codec over an H.263 codec since PVRG-P64 Codec is more suitable for our frame-based experiment and also the bandwidth is not our main concern because our target network environment is local area networks. Our proposed mechanism is in fact applicable to different video coding standards as long as a RTP payload-specific header format (e.g., RFC 2429 for H.263+) and a back-channel message format are available.

To simulate real-time video conferencing, our system encodes at 15 fps (frames per second) the first 800 frames of a typical videophone sequence (Mother and Daughter) in the QCIF format. This video sequence has little motion in it and the background scene does not change frequently as in a video-conferencing setting. We conducted two sets of experiments where one set is in the NACK mode and the other in the ACK mode. Each set has four different packet loss rates so as to observe the impact of packet loss. A Poisson process is used to generate packet loss patterns for different packet loss rates. The same packet loss patterns are used in both sets for fair performance comparison.

The video channel of our H.323 VCS software feeds the H.261 codec 15 input video frames every second. Each frame is first encoded and packetized into three packets on a GOB basis. Note that a modified RFC 2032 header is prefixed to each packetized data packet. The video channel then passes all packets to its RTP module for real-time transmission. Bit rates of 128 kbps are used to simulate high-bandwidth LAN connections. A constant quantizer of 8 is used for encoding the whole sequence. We compare the robustness to errors of an H.261 video coder with only error concealment to another H.261 coder that takes advantage of a feedback channel. The backward channel is error-free since it transports back-channel messages by TCP.

3.2. Experiments in the NACK and ACK Modes

The average peak signal-to-noise ratio (PSNR) is used as a distortion measure of objective quality. Note that PSNR is of less interest when a feedback channel is available since some focus should be set to the relationship between the average bit rate and the recovery time lag. To illustrate the trade-offs among compression performance and video quality, we measure frame sizes and PSNR values. The forced updating in H.261 stipulates that a macroblock should be INTRA-coded at least once for every 132 times of its transmission. The PVRG-P64 Codec simply INTRA-codes one whole frame every 133 frames. Here we present the results for frame 400 through frame 532 in Figures 4 and 5.

In Figure 4, each of the frame size curves with different loss rates (5%, 10%, and 20%) is compared with that at zero packet loss rate. In the NACK mode, each frame error can result in a sudden frame size increase of some subsequent frame. As the packet loss rate rises above 10%, the variance of frame size can be quite dramatic. This can be a burden to a constant bit rate coder. Note that as the loss rate increases, the overall bit rate increases accordingly, implying that the NACK mode is not a good option at high loss rates due to the increases in both bit rates and the number of corrupt frames at the decoder.

All four frame size curves look similar to one another in the ACK mode despite different loss rates. Since only acknowledged frames are used for prediction, packet loss has little impact on the average bit rate as long as the loss rate is not beyond 50% and the round trip delay for the back-channel is short. From Table 1, we can see that the average bit rate is in the range of 120-124 kbps. Unlike in the NACK mode, there is no obvious bit rate increase in the ACK mode as the loss rate rises. When the packet loss rate is below 5% in the NACK mode, the average frame size is slightly larger than 615 bytes and the average bit rate is below 80 kbps as shown in Table 1. In the ACK mode, however, the average frame size is 1014 bytes and the average bit rate is 120 kbps even at

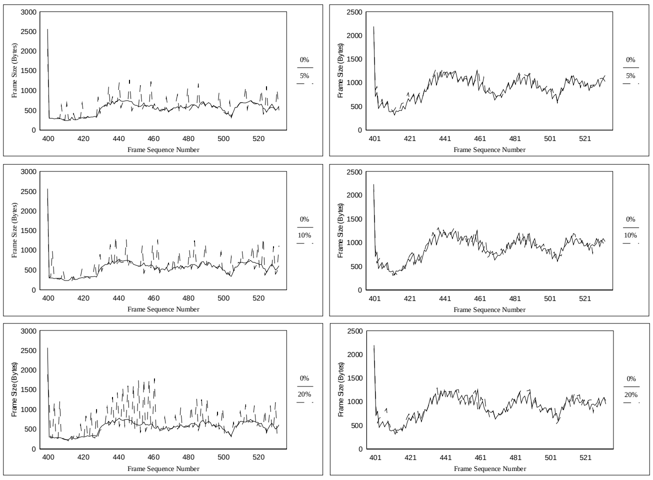

Figure 4: Encoded frame size over frame sequence number for P_B = 0.05, 0.1, 0.2. The operation mode is NACK (left) and ACK (right).

zero loss rate. In other words, the ACK mode requires 50% more bandwidth than the NACK mode does even when no packet loss is present.

From Figure 5 we can see that the NACK mode is more sensitive to packet loss in terms of PSNR values than the ACK mode. Note that PSNR values fluctuate due to the selection of reference frames. Without the capability of selecting reference frames, the curve would have gone down to a level where blocking artifacts and blurring could be very annoying. It is possible to improve the PSNR performance if sophisticated error concealment techniques are employed, which is orthogonal to the main theme of this paper. When the frame rate is high and the round trip delay is short, the time period of affected frames can be tolerable. Again, the ACK mode has a better PSNR performance over the NACK mode since the adverse impact of a packet error is confined to a single frame.

Loss Rate	0%		5%		10%		20%	
	NACK	ACK	NACK	ACK	NACK	ACK	NACK	ACK
Ave. video bit rate	73.3	120.8	78.7	121.6	81.7	122.8	87.3	124.0
Ave. Y-PSNR	34.965	35.084	34.712	35.009	34.325	34.921	33.498	34.53
No. of affected frms Total N.=665	0	0	396	101	511	209	575	453
Ave. no. of frms buf. at the encoder	5.41	5.55	4.29	5.79	3.95	6.26	3.61	6.12

Table 1 Comparison of NACK and ACK modes in terms of averaged measures over 665 frames

The average PSNR values in Table 1 are measured on 665 frames (from frame 134 through frame 798) of the test sequence. We can see that the ACK mode is not sensitive to packet loss in terms of PSNR values. The ACK mode outperforms the NACK mode primarily because only acknowledged frames are used for reference. From Table 1 we can also see that the average bit rate for the ACK mode is quite high even without packet loss. Therefore, the system we devised starts with the NACK mode and switches to the ACK mode when a threshold is met. This threshold needs to take into account the bit rate, the packet loss rate, the number of additional memory buffers, the number of affected frames at the decoder, etc. Clearly, this design issue is system-dependent, since for different network environments and execution platforms, each factor has different weighted importance.

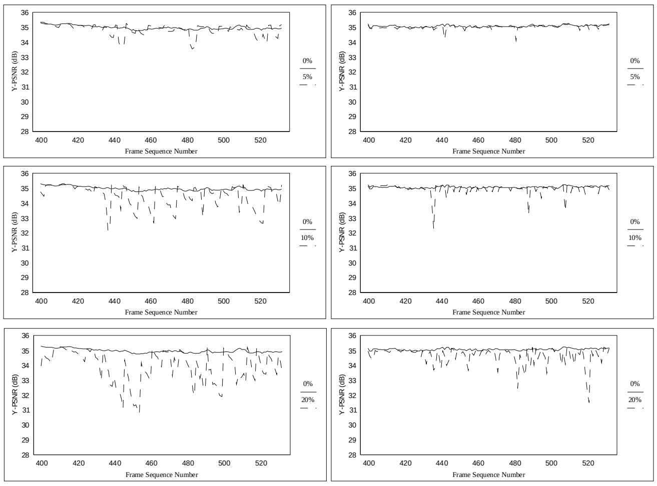

Figure 5: Y-PSNR performance over frame sequence number for P_B = 0.05, 0.1, 0.2. The operation mode is NACK (left) and ACK (right).

4. CONCLUSIONS

This paper explains the design concept of our error-resilient video communication system over IP-based LANs. We have implemented bi-directional error control mechanisms into our H.323 video conferencing software to deal with packet loss under various network conditions. We observe that our system is effective in maintaining picture quality while saving enormous amount of bandwidth as required by full INTRA-frame refresh. Our experimental results have shown that the NACK mode is appropriate at low packet loss rates due to less memory and less bandwidth required. Also, the visual quality can be maintained in our ACK mode even at high packet loss rates up to 20%.

5. REFERENCES

[1] S. U. Portable Video Research Group. PVRG-P64 Codec 1.1. 1993.

[2] T. Turletti and C. Huitema. RTP Payload Format for H.261 Video Streams. 1996.

[3] I. T. Union. ITU-T Recommendation H.245: Control Protocol for Multimedia Communication. 1996.

[4] I. T. Union. ITU-T Recommendation H.323: Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service. May 1996.

[5] I. T. Union. ITU-T Recommendation H.263 Version 2: Video Coding for Low Bitrate Communication. 1998.

[6] Y. Wang and Q.-F. Zhu. Error Control and Concealment for Video Communication: A Review. *Proceedings of the IEEE*, 86(5):974–997, May 1998.

[7] M. H. Willebeek-LeMair and Z.-Y. Shae. Videoconferencing over Packet-Based Networks. *IEEE Journal on Selected Areas in Communications*, 15(6):1101 – 1114, August 1997.