

# TCP-Friendly Internet Video With Smooth and Fast Rate Adaptation and Network-Aware Error Control

Young-Gook Kim, JongWon Kim, *Senior Member, IEEE*, and C.-C. Jay Kuo, *Fellow, IEEE*

**Abstract**—A solution for Internet video streaming with two major components is examined in this paper. First, a new rate adaptation mechanism called the smooth and fast rate adaptation mechanism (SFRAM) is developed based on the transmission control protocol (TCP) throughput equation. SFRAM not only achieves TCP-friendliness but also provides a smooth and fast rate adaptation for transmitted video. By adaptively averaging measurements such as the round-trip time and the packet loss rate over a suitable window, SFRAM mitigates fluctuation that is undesirable for video transmission. The adopted weighting scheme enables the response in a fast manner only for distinct network variations so that the overall network utilization is improved while sustaining end-to-end video quality. When integrated with active routing support such as random early detection and explicit congestion notification, SFRAM provides an even better performance. Second, an adaptive network-aware error control is proposed to alleviate error propagation due to packet loss. Based on SFRAM's capturing of distinct network variations, various source-level error control options such as reference picture selection, video redundancy coding and I-MB refresh are dynamically selected. Extensive experiments by using both the ns-2 network simulator and the real Internet environment are performed to evaluate the dynamic behavior of SFRAM.

**Index Terms**—Error control, Internet video, rate adaptation, TCP-friendliness, video streaming.

## I. INTRODUCTION

INTERNET video, encompassing streaming video playback and video conferencing, has recently emerged as one of essential applications. It is nevertheless a challenging problem to provide the required quality of service (QoS) for multimedia traffic since the best-effort Internet does not guarantee the available bandwidth. The QoS problem of Internet video has been approached via resource allocation such as Integrated Services with resource reservation protocol (RSVP) [1], [2], in which each flow attempts to reserve the resource so that the packet loss rate and the delay are bounded. Although it can provide guarantees, the associated admission control scheme is so complicated that it is still difficult and premature to deploy.

Manuscript received April 13, 2001; revised May 27, 2003. This work was supported by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, and by the Annenberg Center for Communication, University of Southern California. This paper was recommended by Associate Editor H. Gharavi.

Y.-G. Kim is with the Intervideo, Inc., Fremont, CA 94538 USA (e-mail: jameskim@intervideo.com).

J. Kim is with the Department of Information and Communications, Kwang-Ju Institute of Science and Technology (K-JIST), Gwangju, Korea (e-mail: jongwon@kjist.co.kr).

C.-C. J. Kuo is with the Integrated Media Systems Center and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: cckuo@sipi.usc.edu).

Digital Object Identifier 10.1109/TCSVT.2003.819186

Video applications over the best-effort Internet are deployed in the end-to-end sense without expecting the network's coordination, which brings out the following two issues. The first one is the amount of Internet resources (i.e., the available bandwidth) that is available to an end-to-end video application. Transmission control protocol (TCP) [3] traffic such as FTP, HTTP, and TELNET is dominant in today's Internet applications. The congestion control mechanism of TCP has contributed to the robustness of the current Internet for more than two decades [4]. However, TCP is not effective enough for video applications because its window-based congestion control does not provide appropriate rate adaptation. Unlike congestion control of TCP that emphasizes reliability, the congestion control mechanism addressed by this work poses somewhat different demands in order to support high-bandwidth, low-delay, and error-resilient video. We are particularly interested in TCP-friendly rate adaptation on top of user datagram protocol (UDP) and possibly with real-time transport protocol (RTP) as an application-layer control mechanism. The resulting scheme should not only play the same role as TCP's congestion control but also provide favorable conditions for video applications, e.g., smooth rate change.

The second issue is the handling of packet loss and fluctuating delay, which are inevitable in the shared Internet. Although a video application does not require total reliability, it is subject to temporal error propagation due to interframe/intraframe dependency [5], [6]. If a video application is not delay stringent in the sense that the end-to-end latency can be greater than round-trip time (RTT), retransmission by automatic repeat request (ARQ) coupled with a large de-jittering buffer provides the most effective remedy to packet loss. However, since real-time interactive video is delay stringent, ARQ is generally not applicable except for some limited case as given in [7]. Interactive video demands proactive error protection, which is often implemented by forward error correction (FEC) and error-resilient encoding/decoding. Under the bandwidth fairness constraint and the dynamically time-varying Internet traffic scenario, one has to exploit interactive error control schemes (with seamless integration of source and channel error control tools) while sacrificing the coding performance. Thus, a given bit budget should be distributed among source and channel error control tools properly to achieve the best video quality.

In this paper, we address the Internet video transmission problem by providing solutions to the above two issues. The novelty of this work lies in a tight feedback-based integration of TCP-friendly rate adaptation into delay-stringent Internet video. Specifically, there are two main contributions:

- 1) Developing a smooth and fast rate adaptation mechanism (SFRAM) which is TCP-friendly. By adaptively aver-

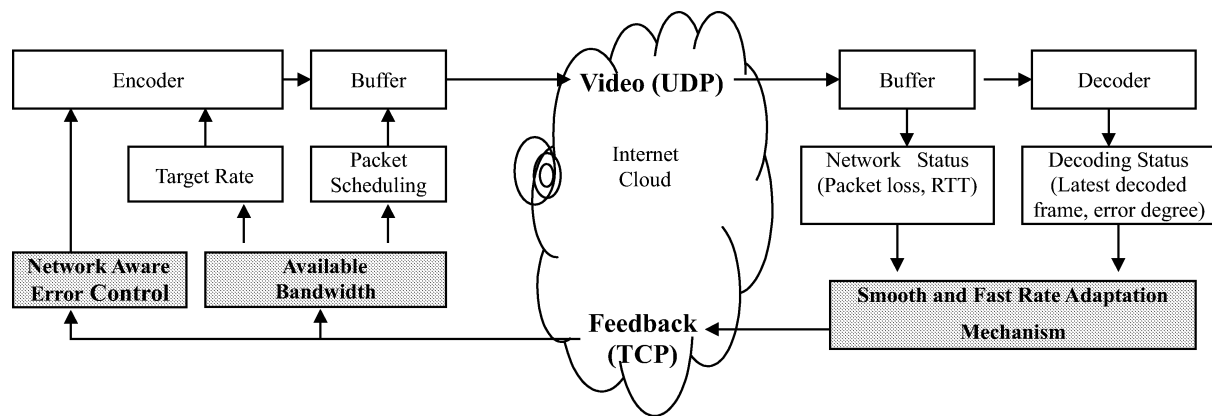


Fig. 1. Illustration of the proposed Internet video transmission.

aging measurements such as RTT and the packet loss rate within a suitable window, SFRAM mitigates unnecessary rate fluctuation that is undesirable for video transmission. The adopted weighting scheme enables the response in a fast manner with respect to distinct network variations so that the overall network utilization is improved and the end-to-end video quality is sustained. When integrated with active routing support such as random early detection (RED) and explicit congestion notification (ECN), SFRAM provides an even better performance.

- 2) Establishing network-aware error control (NAEC) based on SFRAM. SFRAM is able to probe the network status and use the information to help the video coder to select a proper error-control scheme. To be more specific, the video coder can choose the most effective source-level error-control options based on network dynamics.

The rest of the paper is organized as follows. The background information and the motivation for this work are described in Section II with a brief review of previous work. The detailed implementation of SFRAM is given in Section III with a focus on the weighting method and the determination of the sampling window size. Source-level error control options are examined in Section IV and NAEC is derived based on SFRAM. Both network simulation results with the ns-2 network simulator and the actual Internet are presented in Section VI to demonstrate the smooth and fast adaptation capability of SFRAM in various network situations. The effect of SFRAM on video quality and NAEC is also investigated. Concluding remarks are finally given in Section VII.

## II. BACKGROUND AND MOTIVATION

### A. System Overview

The functional block of the Internet video transmission system under consideration is shown in Fig. 1, which is depicted for one-way transmission for simplicity. In this framework, UDP with the proposed rate adaptation mechanism (i.e., SFRAM) is used for the data channel while another external channel using TCP for reliability is adopted for sending feedback signals. In the two-way communication case, packets may piggy back the feedback information. At the receiver end, there is a network status estimator. Basically, it records the packet loss history and RTT.

SFRAM is able to calculate the next available bandwidth (ABW) to keep network utilization high while limiting overload and improving interprotocol fairness. ABW is translated into the target rate of the video coder and used in packet scheduling to meet the bandwidth requirement at the sender. The receiver not only sends ABW feedback through SFRAM but also conveys the network status parameters such as the packet loss rate, ACK/NACK, and RTT. Together with the information from the video coder (e.g., the latest decoded frame and the error degree), the network status can be effectively utilized by SFRAM to achieve NAEC. The receiver-based rate estimation structure is in principle extensible to the multi-user (including multicast) environment since the sender does not have to process raw feedbacks from all receivers [8], [9]. Some preliminary results were reported in [10].

### B. TCP-Friendly Rate Adaptation

Existing TCP-friendly rate adaptation mechanisms [11]–[14] for congestion control can be classified into two categories. One is to mimic TCP's congestion control directly by adopting the additive-increase-multiplicative-decrease (AIMD) principle [12]. With the rule, the transmission rate is increased by sending one more packet in one RTT interval when there is no packet loss. If packet loss is detected, the transmission rate is decreased to one half of the current level. The main characteristic of this approach is the saw-tooth behavior of the resulting transmission rate. Even though the mechanism achieves TCP-friendliness, it is not suitable for video transmission due to dramatic rate fluctuation. For pre-encoded video, it demands a larger buffer to cope with rate fluctuation. This might result in severe quality fluctuation in transmitting real-time coded video. In summary, AIMD leads to a wide dynamic range of bit rates for a given application. It is desirable to have a smoother rate adaptation mechanism compared to AIMD.

To overcome the shortcomings of AIMD, an equation-based approach was proposed in [13] that uses the TCP-throughput model [15] to adjust the transmission rate. The TCP-throughput equation can be written as

$$\text{TCP throughput} = \frac{\text{MTU}}{\text{RTT} \sqrt{\frac{2p}{3}} + \text{RTO} \sqrt{\frac{27p}{8}} p(1 + 32p^2)} \quad (1)$$

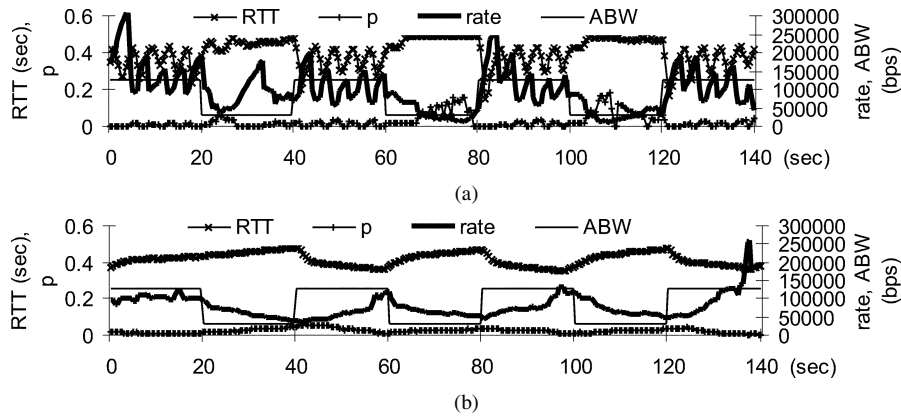


Fig. 2. Rate adaptation with different sampling window sizes with (a)  $w = 6$  and (b)  $w = 50$ .

where MTU is the maximum transmission unit, RTT is the round-trip time (RTT), RTO is the retransmission time out (RTO), and  $p$  is the estimated packet loss rate. TCP increases the number of packets by one in every RTT if there is no packet loss and decreases it by half with packet loss. Simply speaking, this AIMD behavior results in (1) with two independent parameters, RTT and  $p$ . An accurate measure of network status parameters and efficient utilization of measured data are important in practice. RTT and RTO may be measured and updated whenever the acknowledgment of a received packet arrives. For the packet loss rate, we count the packet loss only once per RTT interval regardless of the actual number of lost packets in that interval, since TCP responds only to the first packet loss in one RTT interval. Also, packet loss does not always happen in every RTT interval. Thus, it is necessary to average the packet loss rates over a suitable window of RTT intervals so that (1) predicts the average transmit allowance of TCP connection without divergence. We use the term “sampling window” denoted by  $w$  to indicate the number of RTT intervals used to average  $p$ , RTT, and RTO. It turns out that, with a proper sampling window, the TCP-equation based approach can predict a very stable and smooth rate change compared to AIMD.

To figure out a successful rate adaptation scheme, the selection of  $w$  to avoid excessive smoothing is the key. Let us take a close look at the effect of two different sampling windows as shown in Fig. 2(a) and (b). We defer the explanation of the experimental setup to Section VI. Here, just pay attention to the transient response of rate traces, which are outputs of the TCP-throughput model with two different sampling windows, i.e.,  $w = 6$  and 50, respectively. By changing the total number of flows sharing the same link, the share of each flow is alternating in every 20 s between 32 and 125 kb/s. It is shown as the piecewise-constant ABW traces in these figures. For a shorter window with  $w = 6$ , there are quite a few instances that  $p$  is zero, which causes (1) to diverge. To circumvent this problem, we enforce the additive increase (AI) constraint, which restricts the sender to send one more packet in one RTT interval. However, the packet loss rate still varies too much and, consequently, the transmission rate fluctuates significantly. This is not desirable for video delivery. The receiver will need a larger buffer to cope with this type of rate fluctuation, implying longer end-to-end delay. On the other hand, a larger value of  $w$  enables the rate change in a smooth manner sacrificing the responsive reaction to recent observations. Basically, these results

show that, by using the TCP equation intelligently, one should be able to adapt to the ever-changing Internet in an adaptive flexible manner. In this paper, we propose SFRAM that aims at a smooth yet fast response based on the TCP equation. As a solution to the conflicting demands of video applications and the best-effort network, SFRAM provides a guideline on the available bandwidth dynamically.

### C. Reaction to Packet Loss/Delay via Network Awareness

Packet loss/delay is unavoidable in the best-effort Internet. To prevent error propagation from packet loss, there are two distinct approaches, i.e., with or without feedback. With a reliable feedback channel, delay-constrained ARQ has attracted much attention for packet loss recovery in video transmission [9], [16]. Recently, H.263+ [5] has incorporated a technique called reference picture selection (RPS, also known as NewPred). RPS allows the encoder to select one of several previously decoded frames as a reference picture for motion estimation/compensation. Since motion prediction is arranged with respect to frames that are correctly received by the decoder, error propagation is effectively eliminated [17]. The major drawback of feedback-based techniques is that, when the underlying network does not support reliable feedback channels, they become less useful. Furthermore, efficiency of feedback-based error control can deteriorate due to the feedback delay.

Without a reliable feedback channel, proactive error control is needed. Source-oriented techniques such as Intra-MB (macroblock) refresh are widely used while being independent of the network feedback. Also, channel-level error control such as the use of FEC is available and usually applied in the form of unequal error protection (UEP) [18].

The dynamically varying bandwidth coupled with the TCP-friendly requirement gives a bigger challenge to video applications. If packet loss is observed, we should reduce the transmission rate while increasing the protection to cope with possible packet losses at the same time. The choice of a proper protection scheme is critical in balancing the visual quality of received video. A hybrid error control scheme was introduced in [7], where a structural grouping of video frames protected by piggy-backed FEC and ARQ was proposed. The key frame of a group was better protected by the hybrid channel error control tools to guarantee its timely recovery for the decoding of the next group. One drawback of this approach is that source error control tools were neglected. Even though the hybrid error

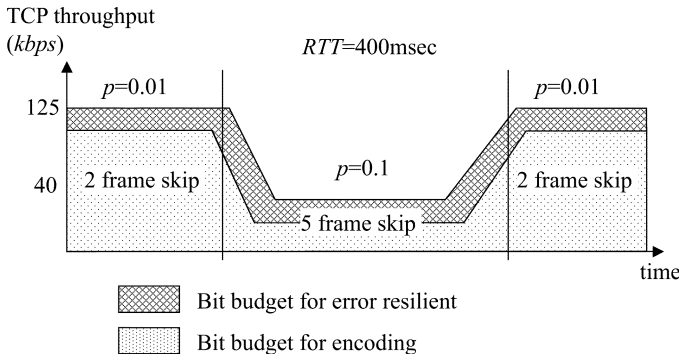


Fig. 3. An example of TCP-friendly rate adaptation and error control.

control demonstrated a good performance, it was achieved at the cost of the overhead bandwidth and the additional recovery complexity. In principle, a proper selection of source/channel error control tools based upon the current network condition should achieve better visual quality with less overhead in terms of both bandwidth and complexity. Keeping this in mind, we investigate network-aware error control together with the SFRAM feedback in this work. Fig. 3 illustrates an example of TCP-friendly rate adaptation and error control in response to abrupt network changes. This is a simplified situation of Fig. 2(b) in the beginning interval of about 60 s. It is assumed that rate adaptation is faster than that in Fig. 2(b). To achieve faster rate adaptation, the encoding frame rate is adjusted accordingly [19] as specified in H.263+ TMN10.

For packet loss/delay control, let us have qualitative characterization below. When the packet loss rate is low, reactive error control with RPS NACK is a reasonable choice. If the channel is getting worse with a higher loss rate, we may apply proactive error recovery tools, which however demand the scarce bandwidth resource. It is therefore critical to allocate the bandwidth (i.e., target rate) properly among video content and error-resilient codes. The feedback is tied with RTT. If the feedback delay and delay jitter are small, one can rely on the ACK-type solution such as RPS ACK, which tends to result in degradation of compression efficiency. Generally speaking, feedback delay tends to increase as the network becomes more congested. With longer delay, it is better to switch to proactive error controls. To conclude, the tradeoff between the overhead and error control options should be coordinated based on the network status and the application context. By employing monitored RTT, loss rates, and packet acknowledgments conveyed by SFRAM, quantitative network-aware error control will be studied in Section IV.

### III. SMOOTH AND FAST RATE ADAPTATION MECHANISM (SFRAM)

#### A. Network Parameter Measurement

To use the TCP-throughput model in (1), we have to measure RTT, calculate RTO, and count the number of lost packets. SFRAM applies TCP's rules to all three parameters while swapping the role of the sender and the receiver. At first, the receiver sends the feedback to the sender as a start signal. The sender puts the feedback number and the elapsed time after receiving the

feedback in every packet header. After receiving each packet, the receiver calculates  $RTT_{\text{sample}}$  via

$$RTT_{\text{sample}} = T_R - T_S - T_E \quad (2)$$

where  $T_R$ ,  $T_S$ , and  $T_E$  are the packet received time at the receiver, the feedback sent time at the receiver, and the elapsed time in the sender, respectively. The receiver uses the exponential filter to update  $RTT_i$  with  $RTT_{\text{sample}}$  for the  $i$ th interval. Also, the TCP algorithm for RTO is adopted to calculate  $RTO_i$ . We put the ECN capability in SFRAM so that the receiver can properly react to the marked packets with the ECN bit [20]–[22]. To calculate the packet loss rate  $p$ , the receiver counts only once regardless of the number of lost or marked packets in every RTT. The packet size (i.e., MTU) can be averaged over a sampling window. After measuring these parameters, the receiver predicts the TCP-throughput with (1) and sends it back to the sender as the next available bandwidth. Even with a large sampling window,  $p$  can be zero to result in infinity in (1). In such a case, we limit the rate increase by allowing the sender to send one more packet in one RTT interval compared to the rate in the previous RTT interval. This is the same as the additive increase principle adopted by TCP and RAP [3], [12].

#### B. Sampling Window Size

Since the goal of SFRAM is to capture the overall network variation effectively, we should average values of RTT, RTO, and  $p$  in a suitable time-scale. As explained in Section II, we use a time window of  $w \times RTT$  as the averaging period and call  $w$  the sampling window. Fig. 4(a) shows the minimum number of RTT intervals to experience at least one packet loss versus the packet loss rate. We observe that, if there is more than 1% packet loss, 10-RTT interval should be enough regardless of the loss rate and RTT. However, the packet loss occurs randomly, and it implies that a larger sampling window is needed to avoid the divergence of (1) for smooth rate prediction.

The probability of  $p = 0$  instance is empirically drawn for varying  $w$ 's in Fig. 4(b) from the ns-2 simulation. It shows that, although there are variations due to the employed network queuing and the transmit rate, it is desirable to set  $w \geq 20$  to avoid divergence. Thus, to avoid the zero-loss instance and achieve smooth rate prediction,  $w = 50$  is adopted in our approach.

#### C. Weighting Method

Even though a longer sampling window helps to achieve a smoother rate, it may not respond to network variations in a fast manner. To demonstrate this point, we show three cases in Table I with a sampling window  $w = 10$ . In the table, a larger index  $i$  means a more recent record. Table I shows the value of  $RTT_i$  and the lost or marked pattern  $(L/M)_i$  for each interval, which is erected to "1" when we encounter lost or marked packets (for the RED-ECN case). By comparing the  $(L/M)$  patterns, we see clearly the difference among the three cases. There is no distinct network variation for case 1 and the network is getting less (or more) congested in case 2 (or case 3).

Although there exist different network variations, the TCP-throughput (1) cannot discriminate them if parameters in all intervals are weighted equally. With equal weighting, the average loss rate  $p$  derived from the  $(L/M)$  pattern and  $RTT$  are the same

TABLE I  
RTT AND THE LOST/MARKED ( $L/M$ ) PACKET PATTERNS FOR THREE CASES

Sample interval ( $i$ )	1	2	3	4	5	6	7	8	9	10	
case 1	$(L/M)_i$	1	0	1	0	1	0	1	0	1	0
	$RTT_i$	0.05	0.04	0.05	0.04	0.05	0.04	0.05	0.04	0.05	0.04
case 2	$(L/M)_i$	1	1	1	1	1	0	0	0	0	0
	$RTT_i$	0.05	0.05	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.04
case 3	$(L/M)_i$	0	0	0	0	0	1	1	1	1	1
	$RTT_i$	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05
weight $W_i$ for $p$		-5	-4	-3	-2	-1	1	2	3	4	5

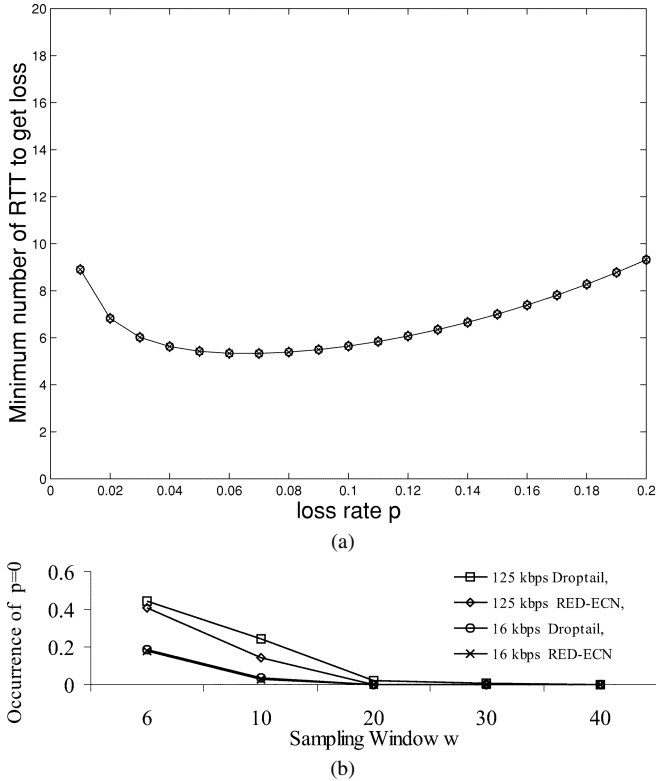


Fig. 4. (a) The minimum number of RTT intervals required to experience the packet loss. (b) The probability of the  $p = 0$  instance versus the sampling window size  $w$ .

for these three cases. Intuitively, we would like to increase the next available bandwidth for case 2 and decrease it for case 3. Thus, to enable fast adaptation to network variations, we consider the following weighting scheme.

First of all,  $RTT_i$  with  $1 \leq i \leq w$  is weighted by  $i^n$  for the averaging, i.e.,

$$RTT = \frac{\sum_{i=1}^w i^n \times RTT_i}{\sum_{i=1}^w i^n}. \quad (3)$$

The same weighting rule is also applied to RTO, i.e.,

$$RTO = \frac{\sum_{i=1}^w i^n \times RTO_i}{\sum_{i=1}^w i^n}. \quad (4)$$

Depending on the value of  $n$ , the averaging formula varies from equal weighting ( $n = 0$ ), linear weighting ( $n = 1$ ), to nonlinear exponential weighting ( $n \geq 1$ ). It plays a role in responding to recent variations of RTT and RTO while still averaging out the short-term fluctuation, which is shorter than the sampling window.

For loss rate  $p$ , we introduce the weighting factor  $W_i$  as tabulated in the bottom of Table I. The average loss rate  $p_a$  is calculated as

$$p_a = \frac{\sum_{i=1}^w (L/M)_i}{\sum_{i=1}^w \text{received}_i}. \quad (5)$$

We incorporate the weighting factor  $W_i$  to calculate the weighted loss rate as

$$p_w = \frac{\sum_{i=1}^w (L/M)_i \times W_i}{\sum_{i=\frac{w}{2}+1}^w W_i}. \quad (6)$$

Note that  $p_w$  becomes slightly positive for case 1, negative for case 2, and positive for case 3. Finally,  $p$  is determined by both  $p_a$  and  $p_w$  via

$$\begin{aligned} &\text{if } (p_w > 0), \quad p = p_a(1 + m \times p_w) \\ &\text{else,} \quad p = p_a \left( \frac{1}{1 - m \times p_w} \right). \end{aligned}$$

The above formula means that  $p$  can be increased up to  $(1 + m)$  times of  $p_a$  if it is getting more congested. It is decreased down to  $1/(1 + m)$  times of  $p_a$ , if getting less congested. The weighting method can be more effective if the RED-ECN scheme is applied since it can provide the congestion status faster to the ( $L/M$ ) pattern. By applying weighted RTT, RTO, and  $p$  to (1), we are able to achieve fast adaptation to network variations.

#### IV. NETWORK-AWARE ERROR CONTROL (NAEC)

Sophisticated compression techniques such as H.263+ and MPEG-4 exploit every possible spatial and temporal dependency for data reduction, rendering severe error propagation of packet loss. Many error resilience options have been proposed to address coding efficiency and robustness, which is called the source-level error control. Note that the tradeoff of each error resilience option is dependent upon the network status.

In this section, we investigate the selection of source-level error control options, including reference picture selection (RPS), video redundancy coding (VRC), and I-MB refresh as defined in H.263+ [5], under various networking conditions. We use a simple example to illustrate proposed ideas throughout the following discussion. In this example, the "Foreman" video clip (in 29.97 fps in QCIF) with 255 frames is coded by using TMN10 rate control at a target rate of 256 kb/s. For all cases (unless specified otherwise), the total bit rate of each scheme is kept constant for a fair performance comparison. To simulate virtual packet loss rate  $p$  with frame-unit packetization, one or two frames are dropped once every  $1/p$  or  $2/p$  frames for

isolated and pair loss patterns, respectively. Since the receiver is aware of packet loss and issues NACK only after it receives a packet of the next frame, the feedback-plus-response delay is set to  $RTT + 33$  ms. The feedback delay is set to  $RTT$  in the case of ACK. With the virtual feedback in place, the encoder can apply the error resilience options by switching the reference frame or refreshing a frame or a macroblock with intracoding. Finally, the average PSNR value for the  $Y$  component at the decoder is used to determine the performance of each error-resilience option.

### A. Reference Picture Selection (RPS)

RPS provides the most effective error-control option when a reliable and low-delay feedback channel is available. Two basic modes, ACK/NACK, are defined in [5]. For the ACK mode, the reference frame in motion-compensated coding can be selected only after its successful decoding, which sacrifices some coding efficiency due to the induced frame gap in motion compensation. In contrast, for the NACK mode, the normal frame referencing (i.e., based on the previous frame) is continued until a frame reception failure is notified. After notification, referencing is frozen until it is resynchronized. Thus, under the NACK mode, the decoder may pause and/or conceal errors till it gets resynchronized. If the application is not delay-stringent, a larger buffer at the decoder can absorb the transitional quality loss at the cost of latency. If it is delay-stringent, one may have to use the latest uncorrupted frame as the reference till resynchronization.

Both RTT and the packet loss rate play an important role in the efficiency of RPS modes. To quantify the effect of RTT and the packet loss rate, we examine the quality degradation for the test example mentioned earlier. For the ACK mode, a buffer size that can store up to 15 frames is allocated so that it can sustain a smooth play operation up to half a second at the maximum (under the rate of 30 fps). In the NACK mode, an isolated loss pattern is applied with different loss rates, and the loss is recovered by inserting intracoded frames, which is effective at the decoder after feedback-plus-response delay. As shown in Fig. 5(a), NACK is effective under a low loss rate (less than 5%) regardless of RTT. At a loss rate of around 5%, it is dependent on RTT. If the frame distance is smaller than 11 frames (or  $RTT = 330$  ms), ACK is better than NACK with more attractive visual quality. The major merit of ACK is persistent visual quality (even though the quality of each frame may be lower due to lower coding efficiency). For comparison, NACK experiences temporal quality disruption. Thus, we conclude that NACK is superior under 5% loss rate while ACK is a better choice around 5% loss rate, if RTT is under 330 ms. NACK does not perform well as the loss rate goes substantially higher than 5% even with a short RTT.

### B. Video Redundancy Coding (VRC)

The VRC option with multiple threads is implemented in H.263+ RPS [5]. Each thread keeps its own temporal dependency chain for packet loss isolation. For overlapping frames in threads, the decoder can choose the best thread based on the reception order and the error history of each thread. The multi-thread VRC scheme is controlled by the number  $N_{th}$  of threads and the number  $N_{fr}$  of frames per thread. At the end of

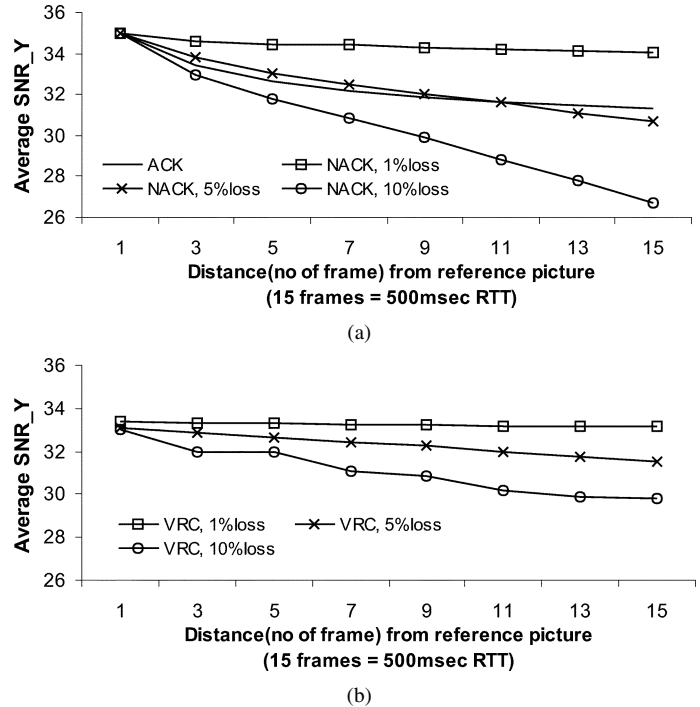


Fig. 5. The average PSNR<sub>Y</sub> value for (a) RPS ACK and NACK modes (b) VRC with  $N_{th} = 2$  and  $N_{fr} = 3$  under the pair loss pattern.

each cycle, redundant synchronization frames are encoded. The synchronization frequency  $f_{sync}$  can be calculated by  $N_{th} \times (N_{fr} - 1) + 1$ . Due to one redundant synchronization frame, coding efficiency is reduced by  $f_{sync} / (f_{sync} + 1)$ . With larger  $N_{th}$ , temporal prediction loss will increase. The error resilience characteristic of VRC is similar to that of RPS. However, in the fixed synchronized frequency case, it is applicable without feedback and not sensitive to RTT. If a separate channel for each thread is provided and the loss is isolated, it may outperform RPS since at least one thread can be kept error-free. However, as packet loss increases, every thread can get affected simultaneously such as with the pair loss pattern and the error may propagate beyond one cycle.

To quantify the VRC performance under various network conditions and error patterns, we first examine the case of  $N_{th} = 2$  and  $N_{fr} = 3$  without feedback. We apply two different packet loss patterns, i.e., both isolated and pair. Especially for the pair pattern, we may lose the last frame of one cycle for each thread. Table II shows that VRC is very good as long as packet loss happens randomly. Even with a high packet loss rate, the average PSNR<sub>Y</sub> is still high. However, if packet loss is bursty and frequent, the VRC performance without feedback is degraded severely. Fig. 5(b) shows the feedback effect for resynchronization at various loss rates with the pair loss pattern. At a loss rate of 1%, it does not have much difference in comparison with the nonfeedback and isolated loss scenario. However, with higher loss and longer RTT, its quality becomes severely degraded. Thus, if packet loss occurs randomly, VRC is effective regardless of feedback delay.

### C. Intramacroblock Refresh (I-MB Refresh)

A simple way to mitigate error propagation is to regularly refresh macroblocks in P frames via intra coding. This is known

TABLE II  
AVERAGE PSNR<sub>Y</sub> OF VRC WITH  $N_{th} = 2$  AND  $N_{fr} = 3$  FOR VARIOUS  
PACKET LOSS RATES AND PATTERNS

loss pattern	1% loss	5% loss	10% loss
isolated	33.01	33.4	33.28
pair	31.77	30.68	26.33

as the I-MB refresh. The ratio and the location of I-MBs in P-frames should be coordinated according to the packet loss rate. This technique is widely applicable regardless of network awareness (e.g., feedback). Recent research [23], [24] has focused on the rate-distortion-optimized I-MB refreshing as part of optimized coding mode decision. However, it usually demands a longer recovery time and results in a mismatched performance especially when it is operated without feedback. When the feedback is available, it can be enhanced to a better scheme by adaptively linking the ratio of I-MBs with the observed network status.

To quantify the effect of I-MB refresh, we perform the experiment under various packet loss rates with an isolated pattern and various I-MB refresh ratios. For example, 10% I-MB refresh means that one MB is intracoded for every ten frames in the average sense. From Table III, we see that 10% I-MB refresh is sufficient in achieving a performance at a loss rate of 1%. As the loss rate goes higher, a higher ratio of I-MB is needed to provide good quality.

## V. SRFAM-DRIVEN NAEC

As discussed in the previous section, the success of NAEC relies on good estimation of the network status and the choice of proper source-level error control options accordingly. In this section, we examine the relation between error control schemes and realistic TCP background traffic obtained via SFRAM.

First, feedback-based RPS is delay sensitive, and RTT is the key factor for its successful deployment. The efficiency of the RPS ACK mode is affected by RTT, even though it is guaranteed to prevent error propagation. The RPS NACK mode, while more efficient up to a higher packet loss rate, is subject to transient quality loss and requires complicated processing. Thus, switching of RPS between ACK and NACK modes has to be integrated with packet loss monitoring under a small RTT value. Fig. 6 shows the PSNR<sub>Y</sub> behavior at the decoder for the ACK/NACK modes of RPS. For the ACK mode, there is no difference between the quality of encoded and decoded frames even for those containing lost packet. Compared with ACK, NACK experiences severe quality degradation until it gets refreshed. Thus, both RTT and loss-rate are critical to NACK while ACK only depends on RTT. For the ACK mode, not only averaged RTT should be small, but its variation should also be small so that the feedback arrives for timely referencing.

VRC with multithreading is capable of preventing error propagation depending on the ratio  $N_{th}/N_{fr}$ . The selection, which makes tradeoff between error resilience and coding efficiency, has to be coordinated according to the application context. Since real-time encoding is delay-stringent, it is not desirable to apply a mode with more than two threads. Fig. 7(a) and (b) illustrates the performance of VRC with  $N_{th} = 2$ ,  $N_{fr} = 3$  and no feedback at 3% and 8% loss rates, respectively. The behavior of

TABLE III  
AVERAGE PSNR<sub>Y</sub> OF I-MB REFRESH

	1% loss	5% loss	10% loss	20% loss
10% I-MB	33.56	32.06	30.12	27.47
20% I-MB	32.93	32.18	30.92	28.92
33% I-MB	31.35	31.39	30.78	29.48

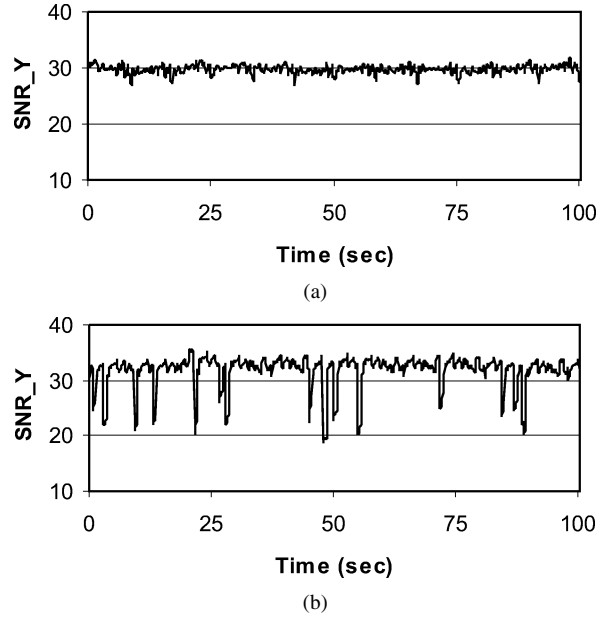


Fig. 6. The effect of RPS with (a) the ACK mode and (b) the NACK mode.

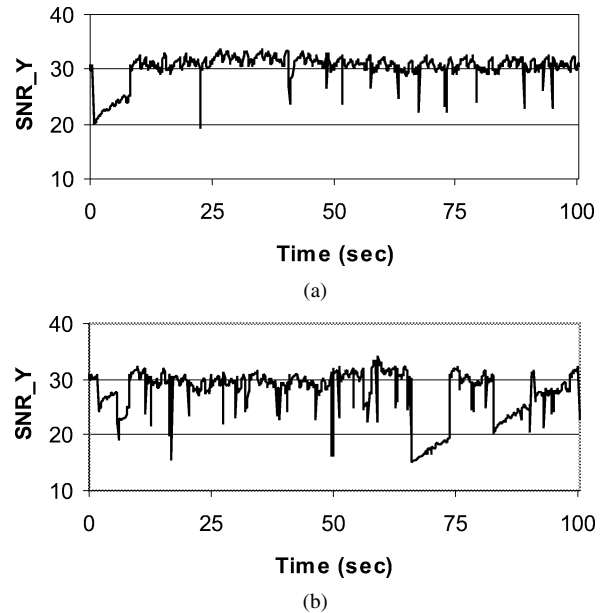


Fig. 7. The effect of VRC with  $N_{th} = 2$  and  $N_{fr} = 3$ : (a) 3% and (b) 8% packet loss rates.

VRC is similar to that of NACK in RPS. At low loss rates, it is better than NACK in the sense that the decoder can decode the uncorrupted chain in one cycle correctly while NACK experiences error propagation until refresh. However, the error may sometimes propagate beyond one cycle as shown in Fig. 7(a) (between 0~10 s). If the feedback is available at the cost of additional complexity, the performance can be improved accord-

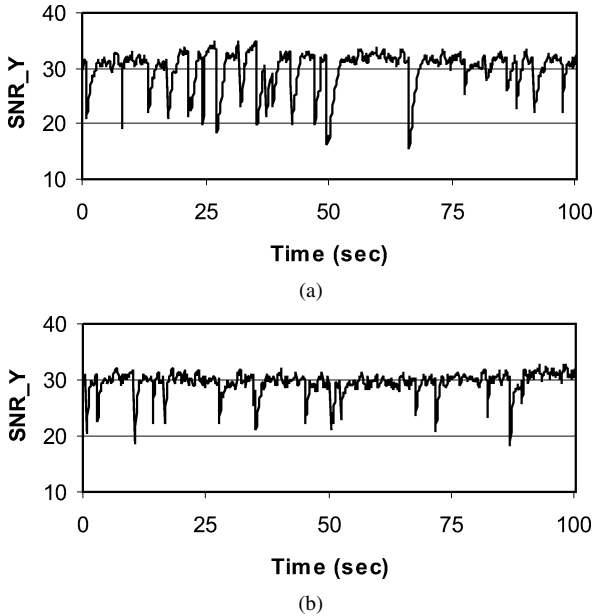


Fig. 8. The I-MB refresh effect for the case of 3% packet loss with (a) 10% and (b) 20% I-MB refresh rates.

ingly by inserting I frames intelligently. However, the probability of consecutive packet losses is low as long as  $p$  is not high. This implies that RTT does not affect the performance of VRC much for a low loss rate. Generally speaking, VRC is an effective tool even with an unstable feedback channel at a reasonable packet loss rate.

I-MB refresh recovers the error in a gradual fashion as shown in Fig. 8, where I-MB is inserted in every 10 and 5 MBs, respectively. For a channel with a low loss rate, a lower refresh frequency achieves a higher PSNR<sub>Y</sub> at the cost of longer recovery time. It is preferred that the refresh rate be adaptive with the network status. In the case of higher packet loss rates, I-MB refresh can outperform VRC without feedback.

To conclude, RPS, VRC, and feedback-free I-MB refresh have a different error control behavior based on RTT and packet loss rates. In some cases, they are complementary and can be used in a hybrid manner. Within each option group, packet loss leads to performance degradation. Our observation is summarized in Fig. 9, which depicts the overall NAEC scheme and its recommended operation ranges. The weighting method of SFRAM attempts to smooth the abrupt response to network changes. When integrated with NAEC, SFRAM can prevent excessive switching among options, since it provides RTT and the loss rate in a smooth manner. Special attention should be paid to determine the switching time. Even though SFRAM predicts the distinct change of RTT and packet loss rates, switching the error recovery mode cannot happen instantly. For example, if it has adopted the I-MB refresh, it needs to start the I frame again for RPS mode. On the contrary, it can switch at any time between the ACK and NACK modes of RPS as long as the encoder keeps the frame buffers for multiple reference pictures.

## VI. EXPERIMENTAL RESULTS

The discrete-event network simulator ns-2 [25] has been extensively utilized for the evaluation of SFRAM and its inter-

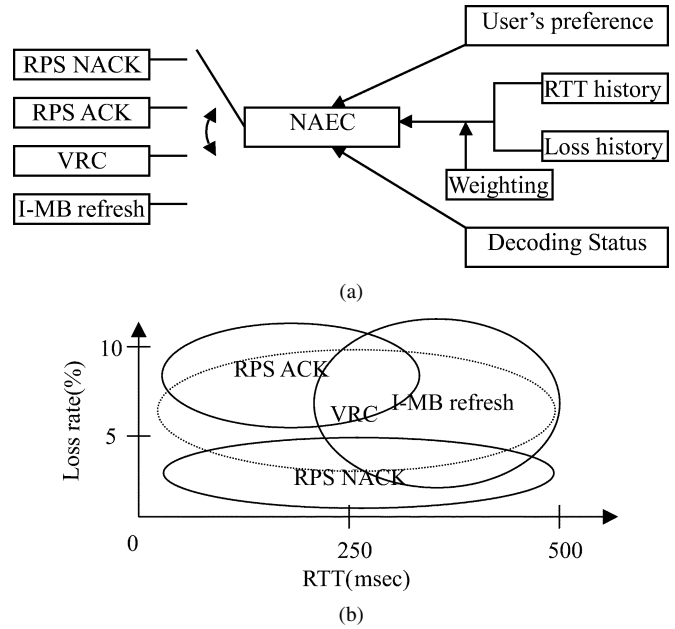


Fig. 9. NAEC: (a) overview and (b) the operational range.

action with Internet video transmission in our experiments. In Section VI-A, basic properties of SFRAM such as TCP-friendliness and smooth and fast rate adaptation are evaluated. Then, SFRAM works with the source-level error control options of H.263+ video for video quality examination. To be more specific, to evaluate NAEC controlled by SFRAM, the ns-2 simulation tool is integrated with on-line H.263+ encoder as shown in Fig. 10. It enables us to examine interaction between the network and the video codec, which is difficult to obtain with pre-encoded video or network trace evaluations.

For the network topology, we used a simple butterfly topology where all end users pass through the same bottleneck link with 2.5 Mb/s bandwidth while all other links of 10 Mb/s were congestion free. This bottleneck link was injected with 20–160 flows during simulation, resulting in from 125 down to 16 kb/s fair share (denoted by ABW) on the average. Propagation delay of the bottleneck link, denoted by  $d_p$ , was set to 50 or 100 ms, which stands for a typical value for the current Internet. For the router, we simulated the droptail or RED queuing with various queuing sizes. The sizes were defined as  $Q_{\max} = Q_k \times d_p \times \text{bandwidth}/\text{packet size}$ , where the queuing delay factor  $Q_k$  varied among 2, 4, or 8, respectively. Under each topology, delay and delay-jitter were mainly affected by the queuing delay (note the short delay of side links). The maximum delay was then calculated by  $d_{\max} = (1 + Q_k) \cdot d_p$ . Not only delay factor  $Q_k$  determines the maximum RTT, but it also affects the packet loss rate since increased queuing can absorb bursty traffic. We set  $\text{minthresh} = Q_{\max}/3$  and  $\text{maxthresh} = Q_{\max}$  for RED queuing. The ECN [21] capability was also incorporated. The marking probability was set to 0.1 for all RED queues, which means ECN bit marking once every 10 packets. For the traffic type, two types of flows, TCP and SFRAM (weighted and nonweighted), were injected side by side for comparative evaluation. For TCP flows, we used the SACK1 version implemented in the ns-2 tool and its packet size was set to 500 bytes. Note that, although TCP-SACK1 and 50/100-ms propagation delay results were provided, other TCP



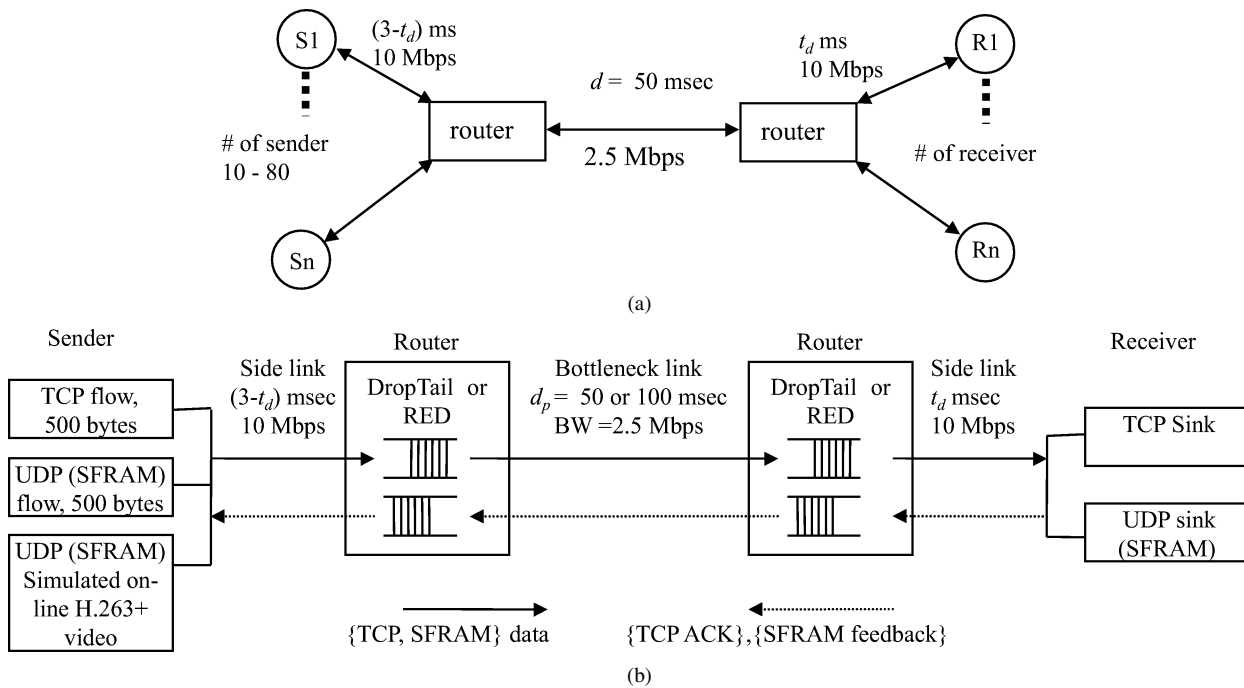


Fig. 10. Simulation setup. (a) Topology. (b) Flows.

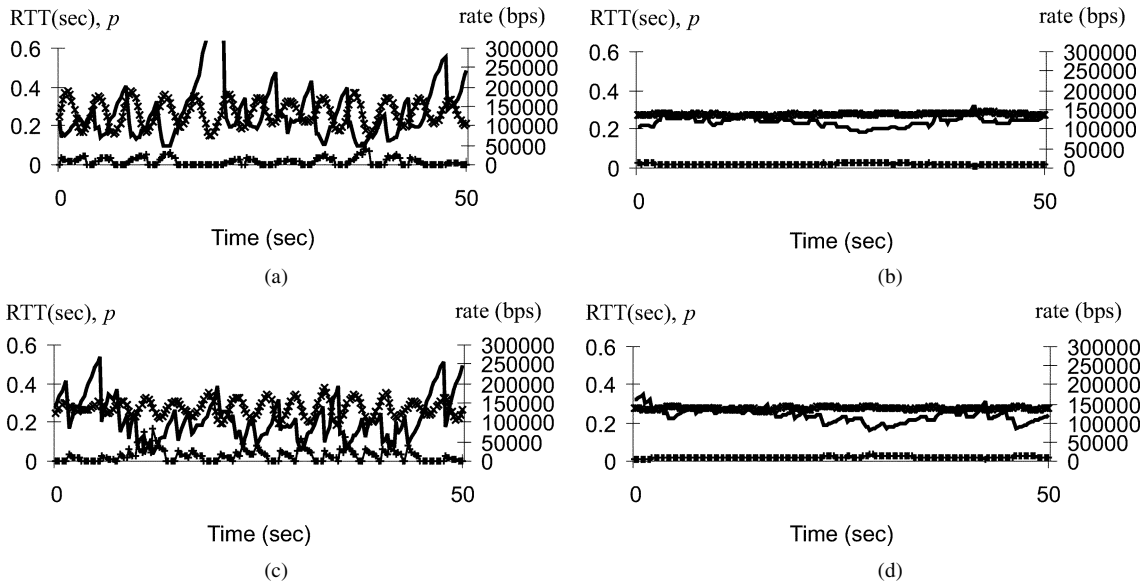


Fig. 11. Smooth rate prediction: (a)  $w = 6$ , RED-ECN, nonweighted, (b)  $w = 50$ , RED-ECN, nonweighted, (c)  $w = 6$ , RED-ECN, SFRAM, and (d)  $w = 50$ , RED-ECN, SFRAM.

implementations and delays were simulated. Nonweighted SFRAM means a simple averaging in calculating (1) while weighting is employed in a normal SFRAM setting. For SFRAM, the rate adaptation mechanism was implemented on top of UDP traffic and the packet size of UDP was set to 500 bytes (if the video encoder is not linked to the SFRAM flow). When linked to the on-line H.263+ encoder, the variable-size output of each frame was contained by a single packet as long as it did not exceed MTU.

#### A. NS-2 Simulation of SFRAM's Properties

First, the smoothing effect of SFRAM was evaluated for sampling window  $w$  set to 6 and 50, respectively. For this scenario, a fixed number (10 in the experiment) of SFRAM flows (with

and without weighting) and one TCP flow were sent over the simulated network ( $Q_k = 8$  and  $d_p = 50$  ms) sharing a bandwidth of 125 kb/s each. For SFRAM,  $n = 4$  and  $m = 1$  were used for weighting parameters. As shown in Fig. 11, a larger window leads to smoother transmission behavior and employed weighting does not hurt smoothing for this steady network. Comparing Fig. 11(a) and (b) with Fig. 11(c) and (d), SFRAM provides a comparable performance in the smoothing effect.

Next, we examined the smooth and fast adaptation capability of SFRAM to network variations. We compared SFRAM with and without weighting by simulating the pulse-train shape variation with a fair share denoted by ABW. By alternating the number of flows from 10-SFRAM/1-TCP flows to

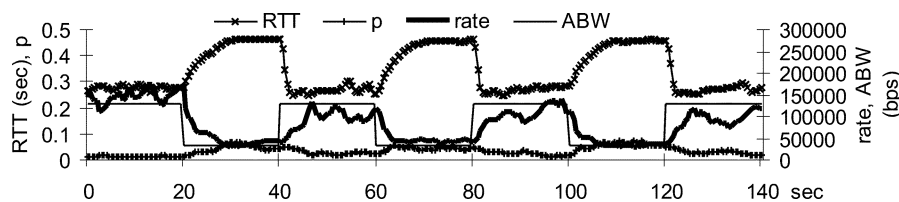


Fig. 12. The rate adaptation behavior of SFRAM (RED-ECN,  $w = 50$ ).

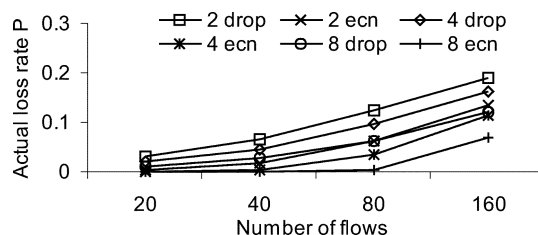
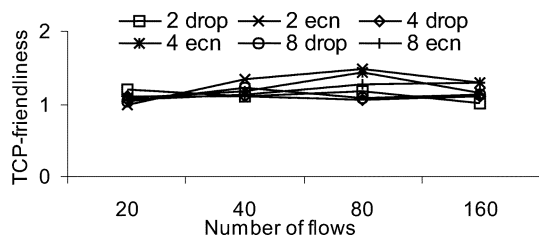


Fig. 13. TCP-friendliness of SFRAM.

10-SFRAM/70-TCP flows every 20 s, each individual flow shared from 125 to 16 kb/s under the  $Q_k = 8$  and  $d_p = 50$  ms setting. For SFRAM,  $n = 4$  and  $m = 1$  were used. To figure out a successful rate adaptation scheme, we examined the two different window sizes,  $w = 6$  and 50. As shown in Fig. 12(a), the packet loss rate varies too much and, consequently, the transmission rate fluctuates significantly. The receiver will need a larger buffer to cope with this type of rate fluctuation, implying longer end-to-end delay. On the other hand, a larger value of  $w$  enables the rate change in a smooth manner sacrificing the responsive reaction to recent observations. As we already examined in Fig. 2(b), there is some delay in adapting to network variations in the case of nonweighted and droptail queue. In comparison, SFRAM of Fig. 12 can adapt to network variations in a fast manner especially when used with RED-ECN.

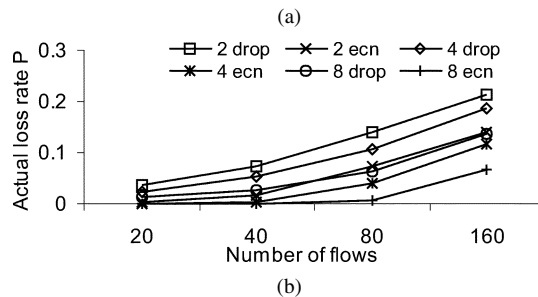
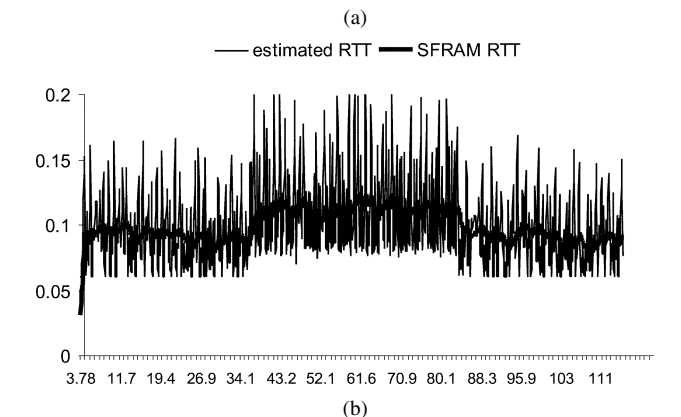
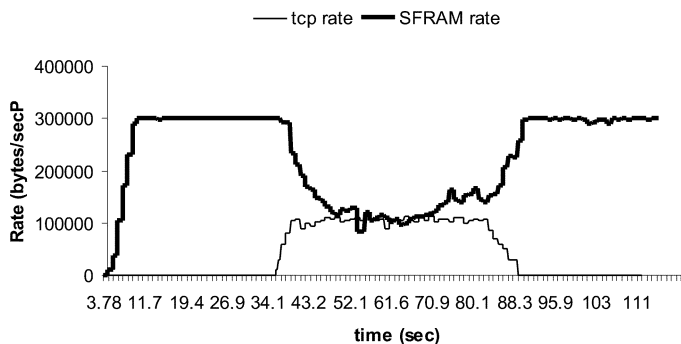


Fig. 14. The actual packet loss rates for (a) SFRAM and (b) TCP with SFRAM.

Last, TCP-friendliness was verified by sending the same number of SFRAM and TCP flows. A broad range of flow numbers and  $Q_k$  values (2, 4, and 8) were simulated with two queuing mechanisms (i.e., droptail and RED-ECN). Results for  $d_p = 50$  ms are shown in Fig. 13, since there is not much difference for other ranges between 10~100 ms. SFRAM flows can share the bandwidth with TCP flows very friendly. From measured  $p$  as shown in Fig. 14, the ECN scheme makes a significant contribution to the packet loss-rate reduction. Unlike the complicated queuing algorithm in [26], the RED-ECN scheme demands only a low complexity. Thus, one can reduce the packet loss rate effectively by implementing ECN. RED-ECN is also beneficial to establish the stable bandwidth sharing since it distributes the packet loss equally among flows.



**B. Internet Experiment of SFRAM's Properties**

We performed real Internet experiments to evaluate SFRAM's properties. A connection was made from almaak.usc.edu in Los Angeles to cross.unomaha.edu in Nebraska. At first, 1 SFRAM flow was sent where  $w$ ,  $n$ , and  $m$  were set to 50, 4, and 2, respectively. After 35 s, 30 TCP flows were sent for 5 Mbytes. Fig. 15 shows the change of the transmission rate for SFRAM and 1 TCP flows measured on 1:00 am, March 24th, 2000. As given in Fig. 15, SFRAM

Fig. 15. Internet experiment for fast adaptation: (a) rate prediction and (b) measured RTT.

achieved smoother rate change, yet responded fast to distinct network variations. We also investigated the smooth rate change when it experienced severe congestion. With the same

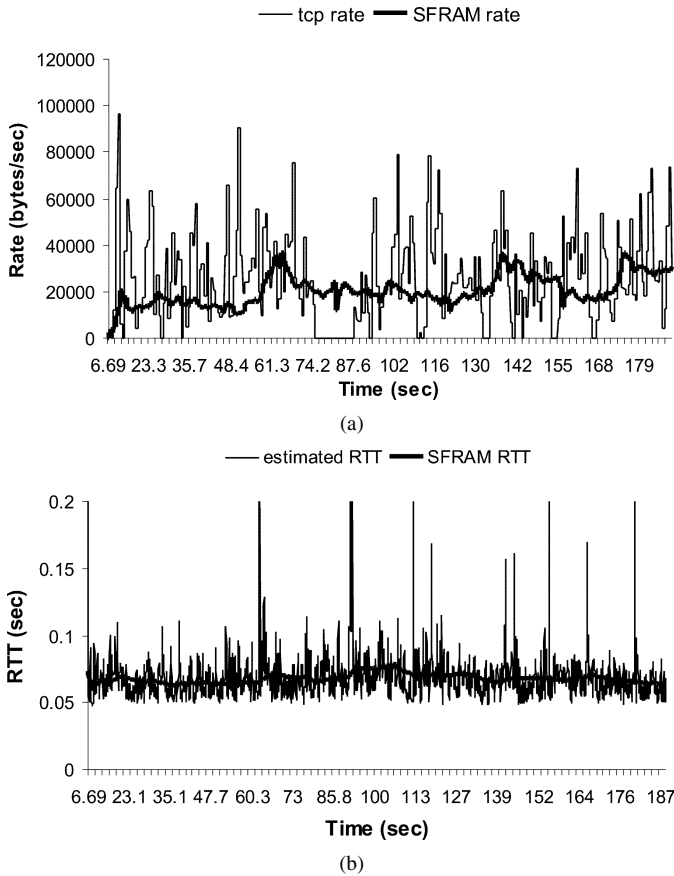


Fig. 16. Internet experiment for smooth adaptation: (a) rate prediction and (b) measured RTT.

network connection above, 1 SFRAM and 5 TCP flows were sent at 11:30 am, March 16th, 2000. Fig. 16 shows that SFRAM achieved smooth transmission compared with TCP flow while it consumed the same amount of bandwidth with TCP on the average.

The effect of the smooth rate prediction from SFRAM on the resulting video quality was also examined. We sent 20 flows of TCP and SFRAM each for 100 s where one SFRAM flow was linked to the on-line H.263+ encoder. Each flow shared 64 kb/s on the average, known as the available bandwidth (ABW). The “Foreman” video clip of the QCIF format was encoded with a target rate based on SFRAM prediction with two different window sizes,  $w = 6$  and  $w = 50$ . For H.263+, the TMN10 rate control scheme with the 2-frame skip mode was used. A video clip of 256 frames was encoded repeatedly for 100 s while the first frame of the image sequence was encoded with intraframe mode. Since the original video clip was captured with 30 fps, approximately 1000 frames were examined during 100 s. It is meaningful that the behavior of the video encoder is examined in a reproducible manner with realistic TCP backgrounds. Fig. 17 shows that the target rate (i.e., rate prediction via SFRAM) and PSNR\_Y for each window size. Note that the PSNR\_Y value was measured at the encoder side without considering the effect of packet loss since we wanted to examine the effect of SFRAM only. Even though both cases used the bandwidth fairly in average, a small window size resulted in video quality fluctuation since the target rate was so dynamic. This implies that a sensitive rate adjustment such as TCP congestion control is not de-

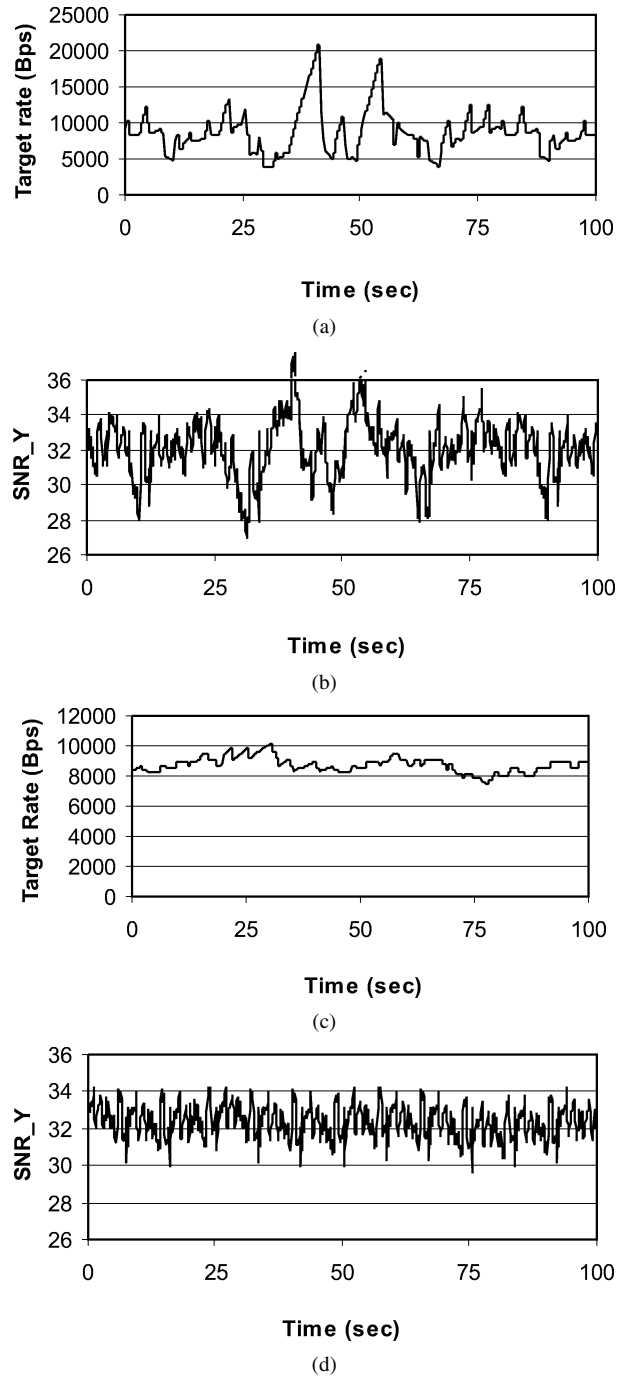


Fig. 17. Video quality adaptation: (a) the target rate with  $w = 6$ , (b) the PSNR\_Y value with  $w = 6$ , (c) the target rate with  $w = 50$ , and (d) the PSNR\_Y value with  $w = 50$ .

TABLE IV  
SELECTION RULE FOR ERROR CONTROL OPTIONS BASED ON RTT  
AND PACKET LOSS-RATE

RTT	loss-rate	optional mode
$< 300$	$> 5\%$	RPS ACK
$< 300$	$< 5\%$	RPS NACK
$> 300$	N/A	Adaptive I-MB, ratio = $3 \times p$

sirable for video transmission, especially for the delay-stringent real-time encoding scenario.

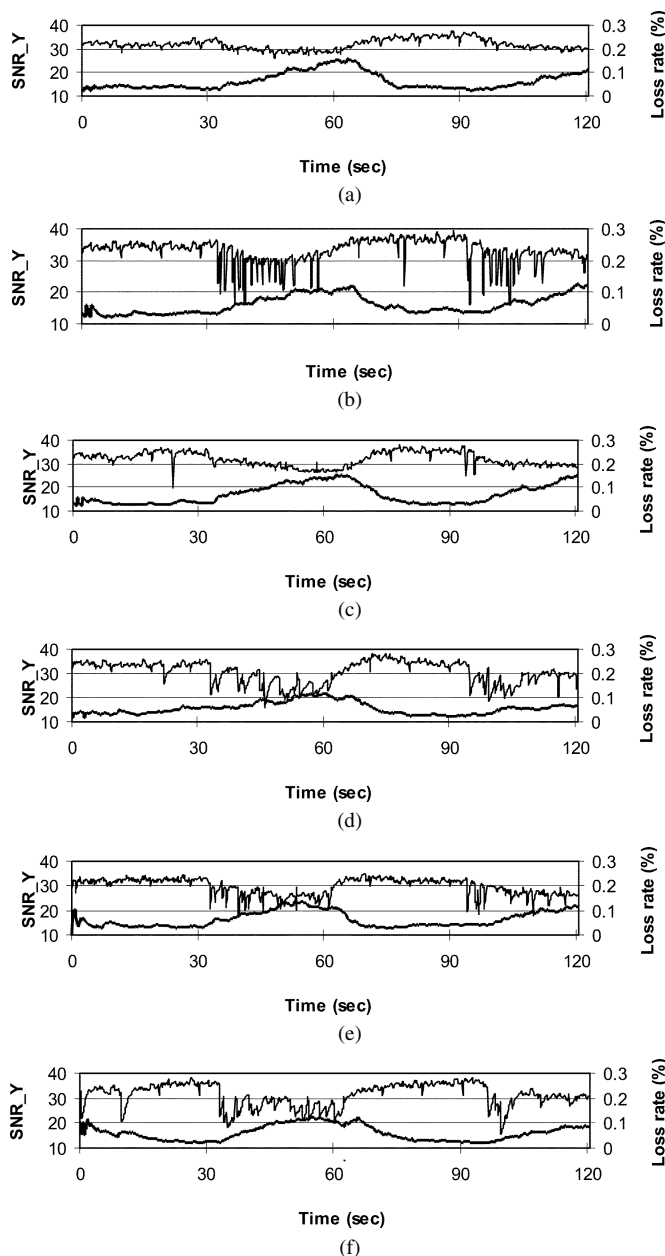


Fig. 18. Comparison of RPS: (a) RPS ACK mode only, (b) RPS NACK mode only, and (c) RPS adaptive ACK/NACK mode. Comparison of I-MB refresh: (d) 10% I-MB refresh, (e) 33% I-MB refresh, and (f) adaptive I-MB refresh.

### C. Performance of NAEC

The effect of NAEC based on the information obtained from SFRAM is studied in this section. We investigated two cases under short and long RTTs. For short RTT, we switched between ACK and NACK modes of RPS adaptively based upon the packet loss rate. The I-MB refresh method was used for longer RTT, where the ratio of I-MB was adjusted adaptively according to the packet loss rate.

First, we set  $Q_k = 2$  and consider the following cases. Let  $d_p$  alternate between 50 and 100 ms when it experienced the maximum RTT found to be 200 and 400 ms, respectively. Both 10 TCP and 10 SFRAM flows were sent for 120 s, where one of the SFRAM flows carried coded video packets. During intervals of 30~60 s and 90~120 s, 60 TCP flows were inserted to cause congestion, which mainly affected the loss rate. The

employed switching rule between source-level error control options is summarized in Table IV. Fig. 18(a)–(c) shows the result of applying RPS ACK/NACK modes adaptively for short RTT. Compared with a fixed mode, adaptive NAEC achieved much a better performance as shown in Fig. 18(c). At the low loss range, video quality of the adaptive scheme is the same as that using the NACK mode only. If the loss rate goes higher than 0.5%, the adaptive scheme is switched back to the ACK mode to prevent error propagation.

For an environment with longer RTT, adaptive I-MB refresh was used. The encoder adjusted the ratio of I-MB to be three times of the loss rate determined by SFRAM. The result is shown in Fig. 18(d)–(f). Compared to a fixed refresh ratio, adaptive refresh in Fig. 18(f) demonstrates both efficiency and resilience.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a rate-adaptive solution to Internet video streaming, called the smooth and fast rate adaptation mechanism (SFRAM) and an associated error control scheme, called network-aware error control (NAEC). Based on experiments with both the ns-2 simulation tool and the real Internet environment, SFRAM was demonstrated to be able to adjust its transmission rate in a very smooth manner by using a large sampling window. In addition, it adapts well to the network variation with a weighting method so that the packet loss rate can be significantly reduced and the overall network utilization is improved. In addition, the integration of RED-ECN into SFRAM provides an even better performance. SFRAM are very useful not only in sustaining stable video quality but also in preventing quality degradation from the packet loss.

Along this direction, an effective NAEC scheme was proposed so that SFRAM can dictate the best source-level error control option available to the encoder. Experiments based on a seamless integration of ns tool and H.263+ encoder showed that NAEC dictated by SFRAM reacts to dynamic network status in an efficient way while reducing severe degradation of video quality. The possibility of NAEC was explored with a switching rule between optional error control options in the frame-level packet scenario.

It is interesting to examine the analytical relationship between RTT, loss rate, and error control options in a more theoretical setting, especially for source/channel-level error control options combined and with fine-grained packetization in GOB or slice of H.263+. This appears to be a very challenging task. Another major issue worth further study is the scalability of the proposed algorithm in the context of multicast applications. Some preliminary results about the receiver-based multilayered multicast were discussed in [10], where a receiver uses SFRAM to estimate the desirable bandwidth for a multicast session. However, this is a very involved research topic and more work is still expected. Finally, how to make unreliable RTP and RTCP feedback be reliable would be very interesting topic since RTP/RTCP provides more efficient means of delivering multimedia contents.

## REFERENCES

- [1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," *IETF RFC 1633*, June 1994.

- [2] L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP)—version 1 functional specification," *IETF RFC 2205*, Sept. 1997.
- [3] W. R. Stevens, *TCP/IP Illustrated Volume 1—The Protocols*. Reading, MA: Addison-Wesley, 1994.
- [4] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Networking*, vol. 7, pp. 458–472, Aug. 1999.
- [5] ITU-T, Recommendation H.263 Version 2—Video Coding for Low Bitrate Communication, Jan. 1998.
- [6] Moving Picture Expert Group, MPEG-4 Video Verification Model Version 10.0, Feb. 1998.
- [7] I. Rhee and S. R. Joshi, "Error recovery for interactive video transmission over the Internet," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1033–1049, June 2000.
- [8] S. McCanne, "Scalable compression and transmission of Internet multicast video," Ph.D. dissertation, Univ. California, Berkeley, 1996.
- [9] X. Li, M. Ammar, and S. Paul, "Layered video multicast with retransmission (LVMR: evaluation of hierarchical rate control)," in *Proc. IEEE INFOCOM'98*, vol. 3, Mar. 1998, pp. 1062–1072.
- [10] Y. Kim and C. C. J. Kuo, "TCP-friendly layered video for internet multicast," *Proc. SPIE*, vol. 4671, pp. 80–91, Jan. 2002.
- [11] S. Jacobs and A. Eleftheriadis, "Streaming video using TCP flow control and dynamic rate shaping," *J. Vis. Commun. Image Representation*, vol. 9, no. 3, pp. 211–222, Sept. 1998.
- [12] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," in *Proc. IEEE INFOCOMM'99*, vol. 3, Mar. 1999, pp. 1337–1345.
- [13] J. Padhye, J. Kurose, and D. Towsley, "A TCP-friendly rate adjustment protocol for continuous media flows over best effort networks," UMass-CMPSCI Tech. Rep. 98-04, Oct. 1998.
- [14] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm: a TCP-friendly adaptation scheme," in *Proc. NOSSDAV'98*, July 1998, pp. 215–226.
- [15] J. Padhye, V. Firoiu, and D. Towsley, "Modeling TCP throughput: a simple model and its empirical validation," UMass, CMPSCI Tech. Rep. 98-008, Feb. 1998.
- [16] R. Xu, C. Myers, H. Zhang, and R. Yavatkar, "Resilient multicast support for continuous-media applications," in *Proc. NOSSDAV'97*, May 1997, pp. 183–194.
- [17] LBC, An error-resilience method based on back channel signaling and FEC, in Document LBC-96-033, 1996.
- [18] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1737–1744, Nov. 1996.
- [19] J. Kim, Y.-G. Kim, H. Song, T.-Y. Kuo, Y. J. Chung, and C.-C. J. Kuo, "TCP-friendly Internet video streaming employing variable frame-rate encoding and interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1164–1177, Oct. 2000.
- [20] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug. 1993.
- [21] K. Ramakrishnan and S. Floyd, "A proposal to add Explicit Congestion Notification (ECN) to IP," *IETF RFC 2481*, Jan. 1999.
- [22] H. Krishnan, "Analyzing explicit congestion notification (ECN) benefits for TCP," master's thesis, Univ. California, Los Angeles, 1998.
- [23] F. Leanne, F. Toutain, and C. Guillemot, "Packet loss resilient MPEG-4 compliant video coding for the Internet," *Signal Processing: Image Commun.*, vol. 15, pp. 35–56, 1999.
- [24] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error prone networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 952–965, May 1999.
- [25] Network Simulator—NS (version 2), UCB/LBNL/VINT. (1998). <http://www-mash.cs.berkeley.edu/ns> [Online]
- [26] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *Proc. IETF RFC 2475*, Dec. 1998.



**Young-Gook Kim** received the B.S. degree in physics education from Seoul National University, Seoul, Korea, in 1993 and the M.S. and Ph.D. degrees in from the Signal and Image Processing Institute of Electrical Engineering, University of Southern California, Los Angeles, in 1997 and 2002, respectively.

His research interests include transport protocol, error control technique, and quality of service for multimedia delivery over Internet systems. He has been with Intervideo, Inc., Fremont, CA, since 2001, where he is currently a Member of Technical Staff working on developing the multimedia delivery system in digital home networks.



**JongWon Kim** (S'88–M'95–SM'01) was born in SangJu, Korea on September 4, 1964. He received the B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, Korea, in 1987, 1989, and 1994, respectively, all in control and instrumentation engineering.

During 1994–1999, he was with the Department of Electronics Engineering, KongJu National University, KongJu, Korea, as an Assistant Professor. From 1997 to 2001, he was visiting the Signal & Image Processing Institute (SIPI) of Electrical Engineering-Systems Department, University of Southern California, Los Angeles, where he has served as a Research Assistant Professor since December 1998. In September 2001, he joined as an Associate Professor the Department of Information & Communications, Kwang-Ju Institute of Science & Technology (K-JIST), Gwangju, Korea. Under the slogan of "Reliable and flexible delivery system for integrated multimedia over wired/wireless IP networks," he is focusing on networked media systems and protocols including multimedia signal processing and communications. He has coauthored around 150 technical publications in international/domestic journals and conferences.

Dr. Kim is a Member of ACM, SPIE, KICS, and IEK.



**C.-C. Jay Kuo** (S'83–M'86–SM'92–F'99) received the B.S. degree from the National Taiwan University, Taipei, in 1980 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively, all in electrical engineering.

He is with the Department of Electrical Engineering-Systems and the Signal and Image Processing Institute, University of Southern California, Los Angeles, where he currently has a joint appointment as Professor of Electrical Engineering and Mathematics. His research interests are in the areas of digital signal and image processing, audio and video coding, multimedia communication technologies and delivery protocols, and embedded system design. He has guided 50 students to their Ph.D. degrees and supervised 10 postdoctoral fellows. He is a coauthor of more than 600 technical publications in international conferences and journals as well as six research monographs.

Dr. Kuo is a Member of SIAM and ACM and a Fellow of SPIE. He is Editor-in-Chief for the *Journal of Visual Communication and Image Representation*. He received the National Science Foundation Young Investigator Award and the Presidential Faculty Fellow Award in 1992 and 1993, respectively.