Video Streaming Over Wireless Packet Networks: An Occupancy-Based Rate Adaptation Perspective

Mohamed Hassan and Marwan Krunz
Department of Electrical & Computer Engineering
University of Arizona
Tucson, AZ 85721,USA
{mhassan, krunz}@ece.arizona.edu

Technical Report
TR-UA-ECE-2005-1
January, 2005

An abridged version of this paper was presented at the *IEEE Globecom 2004*, Dallas, Texas, November 2004. This work was supported in part by the National Science Foundation through grants ANI-0095626, ANI-0313234, and ANI-0325979; and in part by the Center for Low Power Electronics (CLPE) at the University of Arizona. CLPE is supported by NSF (grant #EEC-9523338), the State of Arizona, and a consortium of industrial partners. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Abstract

In wireless streaming applications, the characteristics of the source, the quality of the channel, and the occupancy of the playback buffer play major roles in the target video quality. Automatic repeat request (ARQ) and forward error correction (FEC) are commonly used to improve the reliability of the wireless link. Retransmission delay and the reduction in throughput due to the FEC overhead can lead to playback-buffer starvation. Therefore, it is desirable to reduce the bit rate of the transmitted video signal and increase error protection when the channel state is anticipated to be bad or when buffer starvation is expected. In this study, we first introduce a scalable and adaptive source-channel rate control scheme that assumes no channel variations during frame transmission. We use a probabilistic formulation to adapt the key parameters of the transmission system. Then, we propose a second rate control scheme that takes into consideration the channel variations during the transmission time of a frame and that attempts to provide gracefully degraded video quality. In both schemes, to avoid excessive quality variation, we integrate into the analysis the effect of the playback buffer occupancy which we utilize as a cushion against quality variations. The level of adaptiveness in the proposed schemes is optimized to maximize the rendered video quality while guaranteeing delay and loss bounds. This is achieved by maintaining the occupancy of the playback buffer around a predefined threshold value. Simulation and numerical investigations are carried out to study the interactions among various key parameters and verify the adequacy of the analysis.

Keywords— Source rate control, wireless channels, channel-code optimization, adaptive FEC, playback buffer control.

I. Introduction

A. Motivation

Despite recent advances in wireless technologies and video compression formats, enabling continuous video streaming over wireless channels is still fraught with challenges. On the one hand, the quality of a wireless link fluctuates significantly as result of channel fading and interference. On the other hand, video applications are not only susceptible to transmission errors, but impose tight requirements on the sustainable network throughput and delay-jitter, particularly for interactive communications. The situation is further aggravated by the contention-based nature of common wireless access techniques, which gives rise to radio interference and packet collisions. Collisions can result in packet erasures, whose impact on video quality may extend to several inter-dependent frames. For example, in MPEG sequences with inter-coded frames, artifacts due to network errors/erasures can continue for several successive frames until the next correctly received reference frame.

Several potential solutions have been used to address the situation. A traditional class of solutions uses lossless recovery

transmission through link-layer reliability (i.e., FEC and/or ARQ). Most of the techniques falling in this class are inspired by Shannon's source-channel separation theorem, which states that the design of the source and channel codes can be carried out separately to achieve the optimal performance for the overall transmission system. "Static" FEC can provide sustained throughput and bounded latency at the cost of a slight reduction in video quality (source rate is reduced to accommodate some FEC bits). However, if designed for the worst channel conditions, this approach incurs an unnecessary overhead when the channel is in a good state. Adaptive FEC (code rate varies with the channel's bit error rate) is more effective for a dynamic channel. However, adapting the FEC code rate according to the instantaneous BER in an online fashion is not straightforward and depends on the value of the propagation delay between the transmitter and the receiver relative to the time scale of channel dynamics. ARQ schemes typically incur large delays, preventing the use of such techniques in interactive applications. Hybrid ARQ schemes (e.g., [15], [8]) are believed to provide the best features of ARQ and FEC, and therefore will be used as part of our work.

Another class of solutions is based on source-rate control, often performed at the frame level (e.g., [24], [17]) or the macroblock level (e.g., [30], [22]). Such solutions rely on the scalability offered by standard compression formats. Many of the schemes in this category can be classified as *backward rate control schemes* [25], where the coding parameters for the upcoming frame are chosen based on the transmitter's buffer size, the channel data rate, and the amounts of bits used to encode previous frames. Rate control is typically done with the objectives of maximizing the average throughput and avoiding frame skipping at the transmitter.

A third class of solutions is based on joint source/channel coding (JSCC). In traditional JSCC approaches, also known as high-level JSCC, source coding (e.g., layered video coding [10] and multiple description coding (MDC) [11]) and channel coding are not fully integrated. In layered coding, the source coder produces multiple video layers and assumes that the channel coder guarantees the successful transmission of the most important layer. In MDC, the source coder assumes that the different descriptions are equally protected and are all subject to the same loss statistics. On the other hand, in a low-level or integrated JSCC, the quantizer and entropy-coder at the source coder, and FEC and the modulation scheme at the channel coder are jointly designed in an integrated manner [20], [16].

B. Related Work

Several studies have addressed the topic of video streaming over wireless links. Some schemes simply assume that video streaming takes place over error-free channels (see, for example, [12], [28] and the references therein). Others consider

erroneous channels (e.g., [3], [18], [23], [5], [21], [27], [13], [2], [1]). Very few of these schemes account for the effects of the channel code and/or playback-buffer occupancy, both of which are known to play a major role in the streaming process. The authors in [12] formulated an optimization problem that considers as design parameters the end-to-end delay, the policing constraints, and the encoder and decoder buffer sizes. Although the proposed technique is capable of finding the optimal operating points, due to its complexity, it may not be suitable for real-time computation. In [13] the authors introduced a channel-condition rate control scheme that requires a dynamic codec in which frames can be encoded at any arbitrary rate. They tried to maximize the channel utilization subject to a constraint on the playback buffer size, and suggested that the probability of buffer underflow can be minimized by equating the effective input and output rates of the playback buffer. The authors in [2] studied the rate control problem from the sender's point of view, and proposed rate control schemes that avoid the degradation in the PSNR caused by the significant reduction in the amount of bits allocated to individual frames (in case of encoder buffer fill up). A conditional retransmission and low-delay interleaving scheme was proposed in [1]. According to this scheme, the encoder buffer is used as a part of the interleaving memory. In [4] the authors introduced a stochastic rate control mechanism that uses a priori stochastic models of the source and the underlying channel. In their problem formulation, they divided the optimization process in two stages. The first stage is done offline, where a set of operating points for the allowable system states are precalculated using dynamic programming techniques. The second stage is performed online, and is used to identify the system state. The authors in [6] introduced a framework for streaming stored fine-grained scalable video over a TCP-friendly connection. They proposed a method to find an optimal transmission policy that maximizes the bandwidth efficiency subject to a constraint on the variability of the source rate. For simplicity, they considered CBR-encoded video and assumed a reliable connection where losses occur only when data arrive late at the client. In [7] the authors proposed a joint scheduling and error concealment scheme. In this scheme, the optimal scheduling policy is determined by the sender, taking into account error concealment at the receiver. The authors utilized known results on constrained Markov decision processes over a finite-horizon to obtain the optimal policy for a range of quality metrics. Channel error corrections were not addressed.

C. Main Contributions

While several schemes for transporting video over wireless channels have been suggested in the literature, these schemes are mostly aimed at optimizing the performance of the source and/or channel encoders, without accommodating the networking aspects. For instance, the primary goal of many of these studies is to optimize the effective throughput of the channel,

without taking into account the impact of the source and channel codes on the transport delay. Furthermore, such studies often ignore the dynamics of the playback buffer (i.e., its starvation and overflow), which are critical in maintaining continuous video playback. In addition, some of these schemes are computationally intensive, making them unattractive for real-time streaming. We believe that the literature on video streaming over wireless channels still requires a more comprehensive treatment, whereby channel coding, rate control, ARQ retransmissions, prioritization of video bits (and related unequal error protection), and error concealment are all performed simultaneously and adaptively with the objective of maximizing the likelihood of continuous video playback subject to varying channel conditions and frame sizes. This necessitates a well-designed rate control/scaling strategy that can improve the overall quality subject to the mentioned factors. We argue that successful video streaming hinges on accurate interaction and selection of network, channel, and source-coding parameters. We further argue that a reservoir of few correctly received frames in the playback buffer can significantly reduce the impacts of delay jitter, losses, and unpredictable variations in network resources on video quality. This reservoir is built during a preloading phase in which Q^* frames are prefetched into the buffer before playback commences. Afterwards, the objective is to maintain the playback buffer occupancy around Q^*. This can be achieved by adjusting source and channel rates so that the likelihood of frame-buffer starvation is minimized. The value of Q^*, which is referred to as the *playback buffer threshold*, can be selected depending on the average channel BER, the channel coherence time, and the target video quality.

In this paper, we propose two source-rate control schemes for streaming video over wireless channels with the objective of safeguarding the continuity of the streaming process from unpredictable channel variations. The schemes are designed to maximize the source bit rate at the encoder while preventing/reducing events of starvation at the playback buffer. They employ hybrid ARQ/FEC schemes and exploit the scalability of the video format. Our first scheme assumes no channel variations during frame transmission, making it possible to derive an expression for the probability of the total time needed to correctly deliver the upcoming frame. This assumption is relaxed in the second scheme, which is shown to provide more gracefully degraded quality and soft guarantees on frame delay. The level of adaptiveness in both schemes is optimized to maximize the rendered video quality while guaranteeing delay and loss bounds.

The rest of the paper is organized as follows. In section II we describe the video transmission model and present the proposed adaptive source/channel rate control schemes. Performance evaluation of both schemes is given in section III. Finally, section IV summarizes the results of this study and outlines our future work.

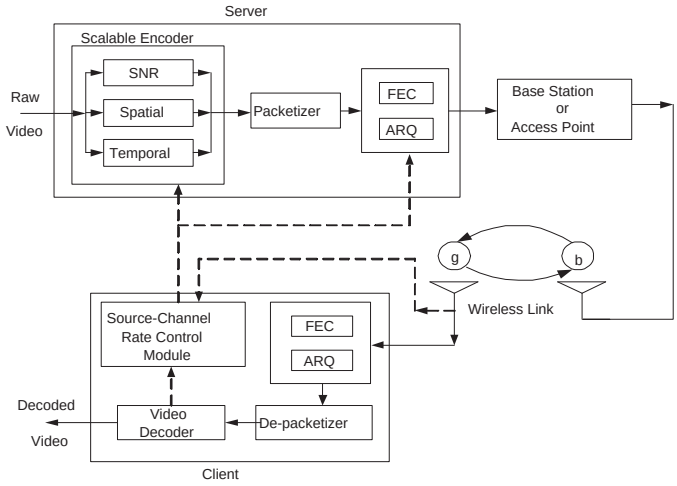


Fig. 1. Overall architecture of the video streaming system.

II. Source Rate Control Schemes

A. System Model

We consider the video streaming system shown in Figure 1. In this system, a video server delivers archived or "real-time" video to mobile clients via a base station (BS) or an access point (AP). We assume that video is encoded using a scalable VBR compressor such as MPEG-4/JVT, and can be rate controlled without any transcoding. The mobile receiver implements a *bandwidth manager*, which continuously monitors the channel state, the playback buffer occupancy, and the quality of the played back video. The receiver can use the history of the received frames to predict the size of the next video frame. Several prediction models for VBR video have been proposed (see [14] and the references therein), some of which can be used in our work. Based on above information, the receiver determines the "optimal" channel code and the required frame-size scaling, and feeds this information back to the BS and the video server. Since control packets are small, they can be adequately protected with FEC alone and/or higher SNR, ensuring that the feedback channel is almost error-free. We assume that the encoder is capable of adjusting its parameters to meet the required rate as computed by the rate control algorithm. We further assume that the wireless channel fluctuates according to a 2-state continuous-time Markov chain, where state 1 is the good state and state 0 is the bad state. For $i \in \{0, 1\}$, let p_i be the BER during state i ($p_1 \ll p_0$). The sojourn times for the good and bad states are exponentially distributed with means α and β, respectively.

B. Stabilizing the Playback Buffer

When a video frame is to be transmitted over the wireless channel, it is first segmented into one or more *link-layer* (LL) packets. Each LL packet undergoes cyclic redundancy check (CRC) followed by FEC coding. The term *decoder failure* is used to refer to errors that could not be fully corrected by the FEC decoder. These errors will be detected by the CRC decoder and will trigger a retransmission of the LL packet. This type of hybrid ARQ assumes that the CRC code is first applied to the packet followed by the FEC code. We assume a stop-and-wait ARQ policy. This assumption is justifiable when the round-trip propagation delay is much smaller than the packet transmission time, as is the case in typical wireless LAN environments. Let S be the size of a video frame. This frame is segmented into $N_p = \lceil S/K_{in} \rceil$ LL packets, where K_{in} is the number of *information bits* in each LL packet. Besides the information bits, each LL packet contains $K_{par,i}$ *parity bits*, for a total of $K_{tot,i} = K_{in} + K_{par,i}$ bits, where i is the channel state. We assume that K_{in} is fixed for all LL packets of the same frame, independent of the channel state. Determining the appropriate $K_{tot,i}$ and $K_{par,i}$ values is done while taking into account the channel characteristics, as will be explained later.

The video session starts with a preloading phase in which $Q^* + 1$ frames are prefetched into the buffer before playback commences. One reasonable choice for Q^* is to set it to a value that is slightly larger than the number of frames drained within an average fade period. The goal of the source-channel rate controller is to try to maintain the playback buffer occupancy around Q^*.

Once the preloading phase is completed, video playback can commence at a rate of f_p frames per second. Let $Q(n)$ be the number of frames in the playback buffer *right after* the display of the nth video frame, $n = 1, 2, \ldots$. Note that $Q(1) = Q^*$. The occupancy of the playback buffer evolves according to:

$$Q(n+1) = \max\left\{0, Q(n) - 1 + \frac{f_r(n)}{f_p}\right\}, \quad n = 1, 2, \ldots \tag{1}$$

where $f_r(n)$ is the average rate at which frames are *correctly* received in the interval between the playback times of the nth and $(n+1)$th frames. Under ideal conditions, $f_r(n) = f_p$, and hence $Q(n+1) = Q(n) = Q^*$. However, when the channel is in a "bad" state (i.e., going through a fading period), we are likely to have $f_r(n) < f_p$, causing the playback buffer to underflow and increasing the backlog at the *transmitter* buffer. Such underflow is compensated for by means of rate control, which allows the transmitter to drain its backlogged queue and catch up with the frame encoding process. During this compensation period, we have $f_r(n) > f_p$. Due to channel uncertainties and the predictive nature of the rate control

algorithm at the receiver, the rate controller may end up overcompensating for the fading periods, leading to $Q(n) > Q^*$.

Define T_c as the *critical time* (in seconds) within which the next frame should arrive correctly at the playback buffer, starting from the most recent playback instant. Essentially, T_c is selected such that the buffer content is kept around Q^*. The value of T_c, which is used in the subsequent determination of the source-rate and channel-code parameters, is selected as follows:

- **Case I:** If $Q(n) > Q^*$, set T_c to $(Q(n) - Q^*)/f_p$.
- **Case II:** If $Q(n) = Q^*$, set T_c to $1/f_p$.
- **Case III:** If $0 \leq Q(n) < Q^*$, set T_c to $1/(f_p(Q^* - Q(n)))$.

The rationale behind the above choice is as follows. When the channel is good, the next frame is expected to arrive after $1/f_p$ seconds. In this case, $f_r(n) = f_p$ in (1), and the buffer reaches its steady-state $Q(n+1) = Q(n) = Q^*$, as desired. In contrast, when the channel is bad, $f_r(n)$ becomes smaller than f_p (at least, temporarily), so the queue length starts to decrease away from Q^*, possibly leading to starvation. To compensate for this, the subsequent $Q^* - Q(n)$ frames need to arrive faster than usual, at an average rate of $f_p(Q^* - Q(n))$ frames per second. If that happens, the queue length will build up to Q^* in one frame period. While $f_r(n) > f_p$, the queue length may exceed Q^*. In this case, future frames need only arrive at a slow rate of $f_p/(Q(n) - Q^*)$.

Although the proposed schemes try to prevent buffer starvation, it may be impossible to completely eliminate such a possibility (e.g., the channel may undergo deep fade for an extended period of time). If starvation occurs, we resort to error concealment to maintain video playback. In this paper, we use a simple concealment approach. Incorporating more sophisticated concealment approaches will be addressed in a future work.

Proactive Concealment Mechanism: When $0 < Q(n) \leq Q^*$, we extend the playback time of the available $Q(n)$ frames by repeating the display of a selected subset of these frames. To minimize the artifacts caused by frame repetition, we enforce a minimum distance between any two repeated frames. To describe the concealment mechanism, we first define the following parameters:

Q_u: Number of uniquely accepted frames (not concealed through repetition but may be partially concealed [29]).

C_{lcf}: Sequence number of last concealed frame. Initially, C_{lcf} is set to 0. This counter keeps track of the position of the last concealed frame in the playback buffer.

D_c: Frame distance between any two concealed frames in the playback buffer. Note that D_c affects the quality of the

reproduced video. Smaller values of D_c mean closely repeated frames.

N_c: Number of so-far concealed frames.

We now summarize the sequence of events that take place when a video frame is added to the frame buffer or drained for playback:

- $Q(n)$ is incremented by one upon the acceptance of any frame in the playback buffer.
- Q_u is incremented/decremented by one when a unique frame is added/played for the first time.
- N_c is set to zero each time the algorithm is executed.
- Both $Q(n)$ and C_{lcf} are decremented by one when a unique frame or a concealed-by-repetition frame is played back.

When C_{lcf} reaches 0, it is not decremented anymore. A value of 0 means that all frames in the decoder buffer are unique.

In this study, we select $D_c \in \{1, \ldots, S_{gop}\}$, where S_{gop} is the size of one group of pictures (GOP). The algorithm in Figure 2 summarizes the steps needed to compute the subset of $Q(n)$ that will be selected for repetition (note that if $Q(n) = 0$, the last played back frame is repeated until more frames are added to the playback buffer).

Initialize: Set $C_{lcf} = 0$

if $(0 \leq Q(n) < Q^*)$ **and** first time to conceal
  Set $j = 0$
  Set $N_c = 0$
  while $(Q(n) < Q^*)$
    if $(Q_u \geq jD_c + C_{lcf} + 1)$
      Mark frame $jD_c + C_{lcf} + 1$ for concealment
      $Q(n) = Q(n) + 1$
      $j = j + 1$
    else
      Break
    end if-else
  end while
  $C_{lcf} = jD_c + N_c + 1$
else
  Set $j = 1$
  Set $N_c = 0$
  while $(Q(n) < Q^*)$
    if $(Q_u \geq jD_c + C_{lcf})$
      Mark frame $jD_c + C_{lcf} + 1$ for concealment
      $Q(n) = Q(n) + 1$
      $j = j + 1$
    else
      Break
    end if-else
  end while
  $C_{lcf} = jD_c + N_c + 1$
end if-else

Fig. 2. Algorithm to calculate the positions of the frames to be concealed.

Scheme 1 (Slowly Varying Channel)

In this scheme, we assume that the channel state does not change during the transmission of a video frame. Once T_c has been updated, the receiver uses it along with the size of the next frame (predicted or actual) and the predicted channel state i to determine the "optimal" channel-code parameters (denoted by $K^*_{tot,i}$ and $K^*_{par,i}$) for the packets of the upcoming frame. Optimality here is in the sense of maximizing the probability of delivering the next frame within T_c seconds. Formally, let $T^{(i)}_{tot}$ be the total time needed to correctly deliver the upcoming frame (including all of its LL packets) when the channel is in state i. Let $F_{tot}(x,i) \stackrel{\text{def}}{=} \Pr\{T^{(i)}_{tot} \leq x\}$, $x \geq 0$, be the CDF of $T^{(i)}_{tot}$. The goal is to find the channel-coding parameters that maximize $F_{tot}(T_c, i)$. If even with such "optimal" parameters, $F_{tot}(T_c, i)$ is still smaller than a given threshold ϵ, then the size of the frame must be reduced. So the receiver reduces the value of S (or $\hat{S}$, if the frame size is predicted), and recomputes the optimal channel-coding parameters. The value of ϵ can be selected according to the relative importance of the transmitted frame, the current channel state, and the number of frames in the playback buffer. The process of gradually scaling down the frame size and recomputing the optimal channel-code parameters continues until an appropriate frame size is found for which $F_{tot}(T_c, i) \geq \epsilon$. At this point, the scaled frame size and the optimal channel-code parameters are fed back to the video server and the BS, respectively. Let ξ be the scaling factor (ratio of the size of the scaled frame to the original size of the frame) . The video server uses the fed back information to scale down the size of the ensuing frame to $\min\{S, \xi\hat{S}\}$. We now describe the above procedure in detail.

Conditioned that the channel is in state i, $i \in \{0, 1\}$, the probability that a received LL packet contains a correctable error is given by:

$$P_{c_i} = \sum_{j=0}^{E_{max,i}} \binom{K_{tot,i}}{j} p_i^j (1-p_i)^{K_{tot,i}-j} \tag{2}$$

where $E_{max,i}$ is the maximum number of correctable errors in a LL packet when the channel is in state i. This quantity depends on $K_{tot,i}$, $K_{par,i}$, and the employed FEC scheme. For example, for Reed-Solomon code, $E_{max,i} = \lfloor K_{par,i}/2 \rfloor$.

Conditioned on channel state i, the number of retransmissions that a given LL packet undergoes (including the first transmission attempt) is a geometric random variable with mean $1/P_{c_i}$. The time between the first transmission attempt for this packet and the receipt of a positive ACK following the last (successful) retransmission attempt for the same packet is also geometric with mean of R/P_{c_i}, where R is the RTT in seconds. We approximate this time by an exponential distribution

of mean $\lambda_i^{-1} = R/P_{c_i}$, $i = 0, 1$. Let $\hat{N}_p$ be the anticipated number of LL packets in the upcoming frame, computed based on $\hat{S}$:

$$\hat{N}_p = \left\lceil \hat{S}/K_{in} \right\rceil = \left\lceil \hat{S}/(K_{tot,i} - K_{par,i}) \right\rceil. \tag{3}$$

Accordingly, $T_{tot}^{(i)}$ is gamma distributed with shape and scale parameters $\hat{N}_p$ and λ_i, respectively. Thus,

$$F_{tot}(T_c, i) = 1 - e^{-\lambda_i T_c} \sum_{k=0}^{\hat{N}_p - 1} \frac{(\lambda_i T_c)^k}{k!}. \tag{4}$$

Figure 3 shows the effects of $K_{tot,i}$ and $K_{par,i}$ on $F_{tot}(T_c, i)$ for RS code. The left-hand side of the figure depicts the delay performance as a function of $E_{max,i}$ when $K_{tot,i} = 750$ and 1000 bits, and the BER is 0.1 (bad channel state). As $E_{max,i}$ is gradually increased, $F_{tot}(T_c, i)$ increases (performance becomes better) up to some optimal point, $E^*_{max,i}$, beyond which the overhead of FEC starts to overshadow its benefit. Note that increasing $E_{max,i}$ at a fixed block size increases the chances of delivering *one* LL packet on time, but it also increases the number of LL packets per frame. The confluence of the two effects gives rise to the trend in Figure 3. The staircase behavior for large values of $E_{max,i}$ is attributed to the truncation effect of the ceil function in (3). A somewhat similar trend is observed in the right-hand side of Figure 3, where an increase in $K_{tot,i}$ (at a fixed $E_{max,i}$) improves the delay performance up to some point, after which the trend is reversed.

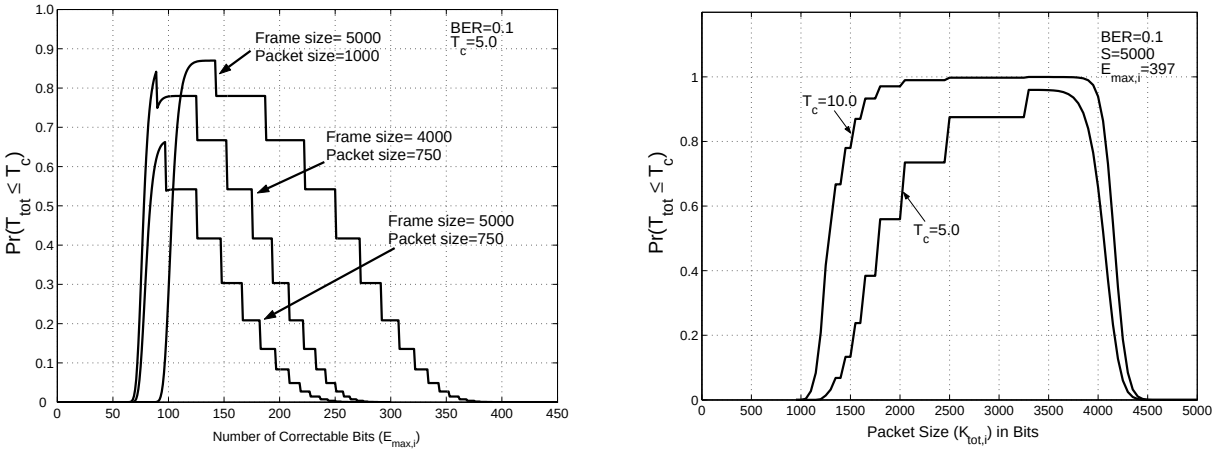


Fig. 3. Impact of channel-code parameters on frame delay (RS Code).

If the optimal ($K^*_{tot,i}$, $K^*_{par,i}$) pair results in $F_{tot}(T_c, i)$ that is still less than ϵ, we resort to rate control. A scaled frame size can be obtained subject to a minimized distortion level [19], [9]. Alternatively, based on the scaled frame size and its sensitivity (i.e., its type and effect on quality), a certain type of scaling (signal-to-noise ratio, spatial, or temporal scaling)

can be chosen. For example, the server can reduce the frame size by increasing the quantization step or by removing high frequency DCT coefficients.

Scheme 2 (Fats Varying Channel)

For the second scheme, we allow the channel state to vary during frame transmission. The receiver uses the updated T_c to determine the optimal size of the next frame. Optimality here is in the sense of maximizing the size of the frame to be transmitted within T_c seconds. Formally, let $\bar{T}_{tot,i}$ be the mean time needed to correctly deliver the next frame (including all its LL packets), assuming frame transmission starts when the channel is in state i, $i \in \{0, 1\}$. The goal is to find the number of packets N_p that can be correctly transmitted in $\bar{T}_{tot,i} = T_c$ seconds while maximizing the allowable size of the upcoming frame. This can be achieved by proper selection of the used packet size as well as the amount of FEC. The scaled frame size (computed by the receiver) and the optimal channel-code parameters are fed back to the video server and the BS, respectively.

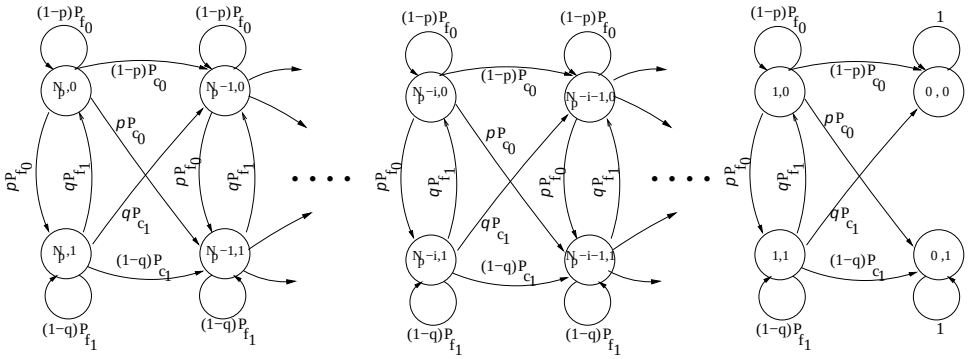


Fig. 4. Transition diagram for the Markov chain that characterizes the evolution of the frame transmission process in scheme 2.

The optimization procedure is now explained in more detail. Conditioned on the event that the channel is in state i, $i \in \{0, 1\}$, the probability that a received LL packet contains uncorrectable errors is $P_{f_i} = 1 - P_{c_i}$, where P_{c_i} is given in (2). This quantity depends on $K_{tot,i}$, $K_{par,i}$, and the employed FEC scheme. In the following, we consider a time-slotted system with channel state transitions occurring at the beginning of these time slots. In other words, we assume that the channel state does not change during one LL packet transmission but may change during the transmission of a video frame. The transmission process of a frame with N_p packets evolves according to the two-dimensional Markov chain in Figure 4. This chain is said to be in state (j, i) when there are j LL packets left to be transmitted and the channel is in state i,

$i \in \{0, 1\}$. In Figure 4, the parameters p and q are defined as:

$$p = P_{01} = P(S_{n+1} = 1 \mid S_n = 0)$$

$$q = P_{10} = P(S_{n+1} = 0 \mid S_n = 1)$$

where S_n is the channel state at the nth observation instant. Note that in the same figure, states (0,0) and (0,1) are absorbing states. Accordingly, $\bar{T}_{tot,i}$ is the mean time to absorption in the above chain. Assume that the chain starts in state $S_i \in \{(N_p, 1), (N_p, 0)\}$. Since this chain is not irreducible, a state $(j-1, i)$ can be considered as an absorbing state to any state (j, i), where $j = N_p, N_p - 1, \ldots, 1$ and $i = 0, 1$. Therefore, we can solve for $\bar{T}_{tot,i}$ by breaking down the chain in Figure 4 into N_p identical stages, where stage j represents the two states $(j, 0)$ and $(j, 1)$. The mean time $\bar{T}_{abs}^{(i)}$ spent in any of these stages represents the mean time to correctly transmit one LL packet and is given by:

$$\bar{T}_{abs,i} = N_{abs,i} \times t_p, \qquad i \in \{0, 1\} \tag{5}$$

where t_p is the time needed to know the result of one packet transmission and $N_{abs,i}$ is the mean number of retransmissions for one LL packet.

For a frame of N_p LL packets, the mean of the total time to absorption is given by:

$$\bar{T}_{tot,i} = \sum_{j=1}^{N_p} \bar{T}_{abs,i} = N_p \times \bar{T}_{abs,i}, \quad i = 0, 1. \tag{6}$$

We now compute $N_{abs,i}$. Throughout the paper, boldfaced notation is used to indicate matrices and vectors. Consider any two adjacent stages, say j and $j-1$, of the Markov chain. Together, they consist of four states, two of which are absorbing for the other two. The probability transition matrix for the Markov chain that controls these four states is given by:

$$\mathbf{\Pi} = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

where

$$\mathbf{Q} \stackrel{\text{def}}{=} \begin{pmatrix} (1-p)P_{f_0} & pP_{f_0} \\ qP_{f_0} & (1-q)P_{f_1} \end{pmatrix}$$

$$\text{and } \mathbf{R} \stackrel{\text{def}}{=} \begin{pmatrix} (1-p)P_{c_0} & pP_{c_0} \\ qP_{c_1} & (1-q)P_{c_1} \end{pmatrix}.$$

Let $\mathbf{V} \stackrel{\text{def}}{=} [v_{ij}]_{i,j}$, where v_{ij} is the number of visits to the transient state j starting from the transient state i. Then,

$$\begin{aligned} v_{ij} &= E\left[\sum_{n=0}^{\infty} I\{S_n = j | S_0 = i\}\right] \\ &= \sum_{n=0}^{\infty} E[I\{S_n = j | S_0 = i\}] \\ &= \sum_{n=0}^{\infty} P\{S_n = j | S_o = i\} = \sum_{n=0}^{\infty} [\mathbf{Q}^n]_{ij} \end{aligned} \tag{7}$$

where $I\{.\}$ is the indicator function and $[\mathbf{Q}^n]_{ij}$ is the (i,j)th element of the matrix $\mathbf{Q}^n$. Hence, we can write the fundamental matrix $\mathbf{V} = E[v_j | x_o = i]$ as $\mathbf{V} = \mathbf{I} + \mathbf{Q} + \mathbf{Q^2} + \ldots$, where $\mathbf{Q^n}$ goes to zero as n goes to infinity. Thus, $\mathbf{V} = [\mathbf{I} - \mathbf{Q}]^{-1}$. The mean time to absorption $\mathbf{N}_{abs,i} \stackrel{\text{def}}{=} [N_{abs,0}\ N_{abs,1}]$ given that the chain starts in a transient state i is given by:

$$\mathbf{N}_{abs,i} = \mathbf{e}\mathbf{V} = \mathbf{e}[\mathbf{I} - \mathbf{Q}]^{-1} \tag{8}$$

where $\mathbf{e}$ is a row vector of ones. With some manipulation, it can be shown that:

$$\begin{aligned} N_{abs,i} &= \frac{1 + pP_{f_0} + qP_{f_1} - P_{f_{(1-i)}}}{[1-(1-p)P_{f_0}][1-(1-q)P_{f_1}] - qpP_{f_0}P_{f_1}}, \\ & \quad i = 0, 1. \end{aligned} \tag{9}$$

We set $\bar{T}_{tot,i} = T_c$ and use it with $\bar{T}_{abs,i}$ to determine the number of packets in the upcoming frame:

$$N_p = \left\lceil \frac{T_c}{\bar{T}_{abs,i}} \right\rceil. \tag{10}$$

Hence, the required (scaled) size of the next frame is:

$$\xi \hat{S} = N_p \times K_{in} = \left\lceil \frac{T_c}{\bar{T}_{abs,i}} \right\rceil \times (K_{tot,i} - K_{par,i}). \tag{11}$$

The above equation is a function of the channel-code parameters $K_{tot,i}$ and $K_{par,i}$ (note that $\bar{T}_{abs,i}$ itself depends on these

two parameters). To "optimize" $\xi\hat{S}$ (i.e., minimize the amount of rate scaling performed on the frame), both $K_{tot,i}$ and $K_{par,i}$ must be properly selected, which is the topic of the next section.

1) Adaptive Computation of the Channel Code Parameters: In this section, we determine suboptimal values for $K_{tot,i}$ and $E_{max,i}$, denoted by $K^*_{tot,i}$ and $E^*_{max,i}$, which are to be used throughout one video session. For a given channel state i, the selection procedure starts by choosing an "optimal" value for $E_{max,i}$. Optimality here is in the sense of maximizing the sizes of the transmitted frames (i.e., we choose the minimum possible $E_{max,i}$ and the maximum possible $K_{tot,i}$). Figure 5 depicts the behavior of P_{c_i} as a function of $E_{max,i}$ for a fixed packet size. We select $E^*_{max,i}$ to be the point on the x-axis at which P_{c_i} starts to saturate. Formally, this is done by replacing the discrete binomial term in (2) by a continuous normal density function and the outer summation by an integral. We differentiate the resulting equation with respect to $E_{max,i}$. The resulting derivative has a bell-shaped form with a peak at $E_{max,i} = p_i K_{tot,i}$. This derivative approaches zero as $E_{max,i}$ goes to infinity. Since our goal is to choose the minimum needed $E_{max,i}$, we set the derivative to a small number $\delta \ll 1$ and solve for $E^*_{max,i}$. For state i, a reasonable approximation for $E^*_{max,i}$ is given by

$$E^*_{max,i} \approx \left\lceil p_i K_{tot,i} + r\sqrt{p_i K_{tot,i}(1-p_i)} \right\rceil \tag{12}$$

where r is the number of standard deviations from the mean of the normal density function (e.g., $r = 3$ is usually sufficient for our purposes). Note that increasing $E_{max,i}$ beyond $E^*_{max,i}$ at a fixed packet size increases the chances of delivering *one* LL packet, but it also reduces the resulting frame size and hence the reproduced quality. Figure 6 depicts P_{c_i} versus $K_{tot,i}$ at a fixed $E_{max,i}$. For a given BER, a desired value of P_{c_i} can be achieved for a range of values of $K_{tot,i}$. We select $K^*_{tot,i}$ to be the maximum value in this range (after which P_{c_i} starts to decline as a result of the limited correction capability of a fixed $E_{max,i}$). It is also obvious that there is always an optimal $K_{tot,i}$ associated with different channel states and a fixed $E_{max,i}$. In other words, the optimal $K_{tot,i}$ is the maximum $K_{tot,i}$ that can be obtained by moving right on the probability curve in Figure 6 and just before it starts to decline. The suboptimal channel-code pair ($K^*_{tot,i}$, $K^*_{par,i}$) can be obtained by a simple search, as follows. Assuming that $E^*_{max,i}$ is known and fixed (as described above), the suboptimal packet size can be obtained by solving for the largest $K_{tot,i}$ satisfying the following inequality:

$$P_{c_i} = \sum_{j=0}^{E_{max,i}} \binom{K_{tot,i}}{j} p_i^j (1-p_i)^{K_{tot,i}-j} \geq 1 - \upsilon \tag{13}$$

where $0 < \upsilon \ll 1$ is a predefined control parameter.

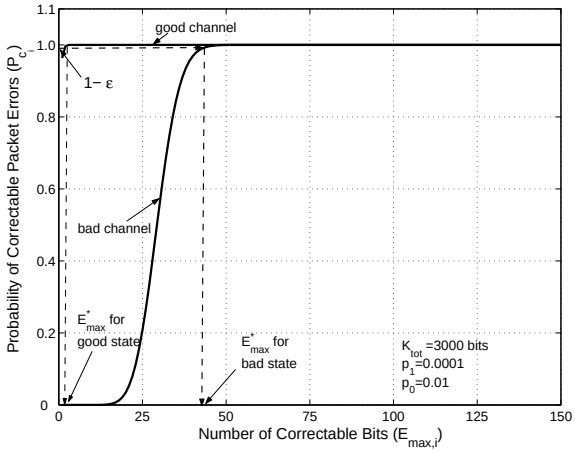


Fig. 5. P_{c_i} versus $E_{max,i}$ ($K_{tot,i} = 3000$ bits).

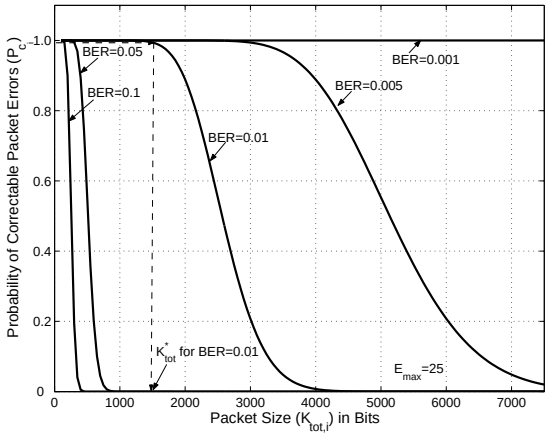


Fig. 6. P_{c_i} versus $K_{tot,i}$ for various BER values ($E_{max,i} = 25$ bits).

To simplify the packetization and channel coding process, we fix the number of information bits K_{in} for all frames and vary the amount of FEC based on the channel state. Accordingly,

$$K_{in} = K_{tot,1} - K_{par,1} = K_{tot,0} - K_{par,0}. \tag{14}$$

Hence, we first solve (13) for $K_{tot,i}$ by replacing $E_{max,i}$, $i \in \{0, 1\}$ with its expression in (12). Then, we check whether the obtained pair $(K_{tot,i}, E_{max,i})$ satisfy (14). If the condition in (14) is not satisfied, we repeat the above computations until it is satisfied.

III. Simulation Results

In this section, we study by simulations the performance of the proposed schemes. Table I summarizes the default parameter values used in the simulations. For scheme 1, we set $\epsilon = 0.9$. For scheme 2, we let $\delta = \upsilon = 0.0001$. We also set $K_{tot,1} = 2321, K_{par,1} = 1, K_{tot,0} = 2377$, and $K_{par,0} = 58$ bits, respectively. These values were determined for RS codes as described in section II-B.1. The traces used in the simulations are MPEG-4 encoded and were obtained from [26].

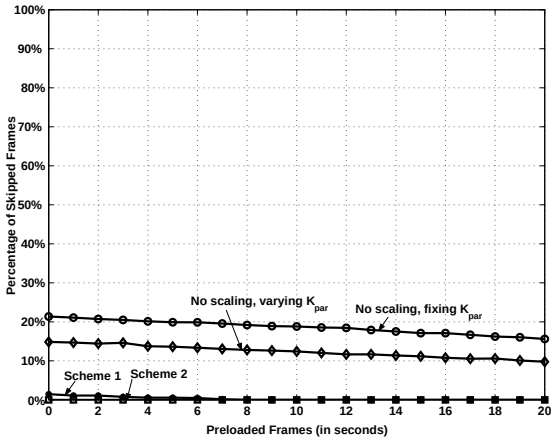
Figure 7 depicts the relative percentage of skipped frames versus the number of preloaded frames $Q(0)$ and Q^*. In the proposed schemes, a video session starts with a preloading phase in which $Q^* + 1$ frames are prefetched into the buffer before playback commences. However, in order to isolate the effect of the preloading phase and gauge its impact on the percentage of skipped frames separately from other factors, in Figure 7-a we fix Q^* at 24. The figure shows that with the proposed adaptive schemes and using a reasonable number of preloaded frames, no frames are skipped (it is obvious that scheme 2 does not require a large preloading phase). When scaling is not used, we fix the block size K_{tot} at 1000 bits and consider two cases: adaptive and nonadaptive. For the adaptive case, we allow E_{max} (and hence K_{par}) to vary with the channel state, while for the nonadaptive case, we fix E_{max} at a conservative value of $p_0 K_{tot}$. It is obvious that the proposed schemes outperform the two cases without scaling. Figure 7-b shows that the relative percentage of skipped frames for $E_{max} = 50$ is higher than when $E_{max} = 20$. Also, this percentage is higher for $E_{max} = 5$ than for $E_{max} = 10$. Therefore, to achieve higher playback rates (alternatively, fewer buffer starvation instances), the number of correctable bits cannot be arbitrarily chosen. Figure 8 depicts the relative percentage of skipped frames versus C. In this figure, we also vary $K_{tot,i}$ and $E_{max,i}$ according to channel state but without scaling the source rate. Again, the proposed schemes outperform all the adaptive and nonadaptive cases without scaling (i.e., when $E_{max,i}$ is fixed, $E_{max,i}$ is varied, and the case when both $K_{tot,i}$ and $E_{max,i}$ are varied).

Parameter	Value
f_p	24 frames/sec
p_0	10^{-2}
p_1	10^{-4}
RTT	0.04 sec
Access bandwidth	1 Mbps

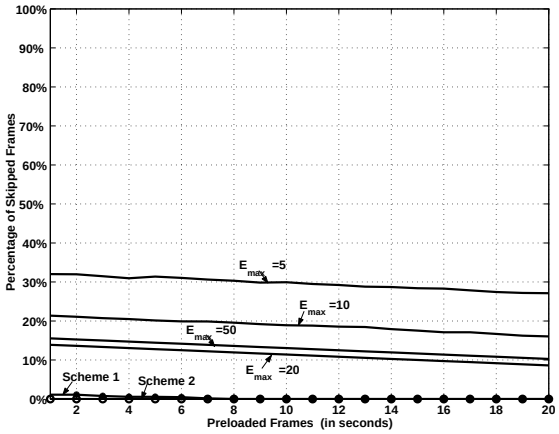
TABLE I
DEFAULT VALUES OF PARAMETERS USED IN THE SIMULATIONS.

Figure 9 depicts the relative percentage of skipped frames versus K_{tot} and E_{max}. For fixed values of E_{max}, Figure 9-a

shows that small values of K_{tot} result in higher skipping rates. This is because the overhead of FEC outweighs its benefits for small values of K_{tot}. As K_{tot} increases, the skipping rate decreases up to a certain point $K_{tot} = K^*_{tot}$ (which depends on the BER and E_{max}), which one may call the "optimal" packet size. Beyond K^*_{tot} the trend is reversed. Similarly, for fixed values of K_{tot}, Figure 9-b shows that a non-careful selection of E_{max} results in a high skipping rate. This is explained by the fact that a small amount of FEC has a limited correction capability with respect to K_{tot}. As E_{max} increases, the skipping rate decreases up to a certain $E_{max} = E^*_{max}$, after which the trend is reversed. In the same figure, it is obvious that $E_{max} = 24$ is on the border of the optimal value E^*_{max} for $K_{tot} = 1000$



(a) Varying $Q(0)$ ($Q^* = 24$)



(b) Varying Q^*

Fig. 7. Relative percentage of skipped frames versus $Q(0)$ and Q^*.

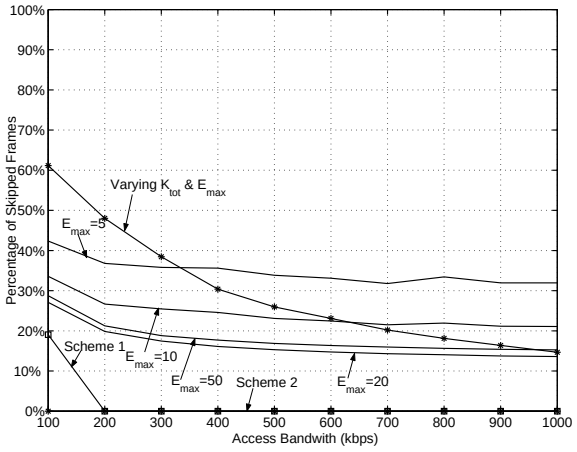
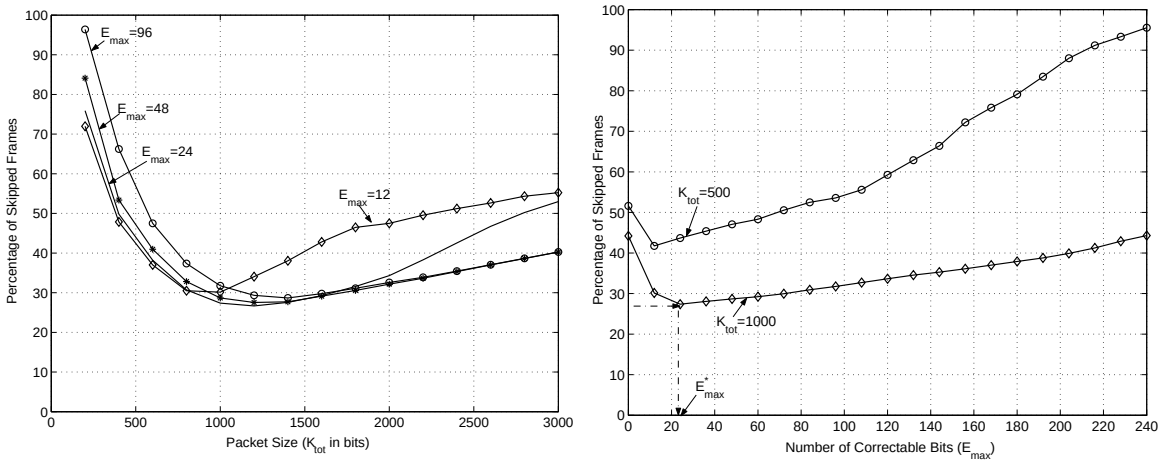


Fig. 8. Relative percentage of skipped frames versus access bandwidth (C).

In Figure 10, we depict the average length of a starvation period versus Q^* for scheme 1 with and without concealment. This figure also shows the performance of fixed FEC strategies. It can be seen that scheme 1 with concealment eliminates



(a) Varying K_{tot} (b) Varying E_{max}

Fig. 9. Impact of channel-code parameters on percentage of skipped frames.

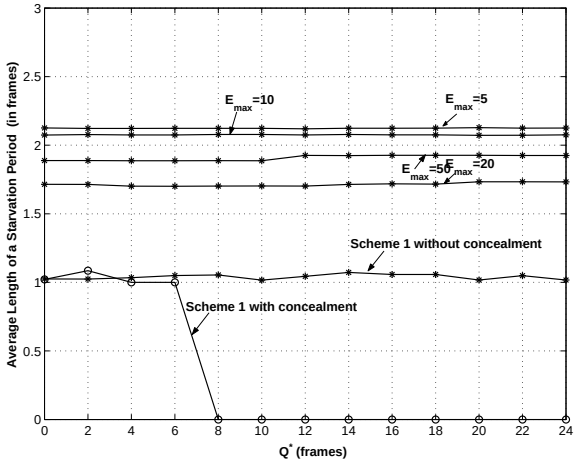


Fig. 10. Average length of a starvation period versus Q^*.

starvation for $Q^* \geq 8$ frames. For scheme 2, the obtained values are observed to be zero all the times. For scheme 1, Figures 11 and 12 show the relative sizes of the scaled frames (normalized by the actual frame sizes) for B, P, and I frames with and without concealment. When proactive concealment is not used, B frames undergo less scaling than P and I frames due the fact that they are relatively smaller in size. It is obvious that the relative percentage of the sizes of the scaled frames is higher for all frame types when applying the proposed concealment method. This is due the fact that concealing by repetition increases the value of T_c and consequently the sizes of transmitted frames. It is worth noting that we obtained the same behavior for scheme 2 but with higher percentage of the scaled frames sizes to the actual frame sizes.

Table II shows a comparison of the average normalized sizes of scaled frames using the two proposed schemes (without concealment). It can be observed that the resulting scaled frame sizes in scheme 2 are higher than those achieved with

scheme 1. This supports our argument that the channel behavior during one frame transmission time must be taken into account.

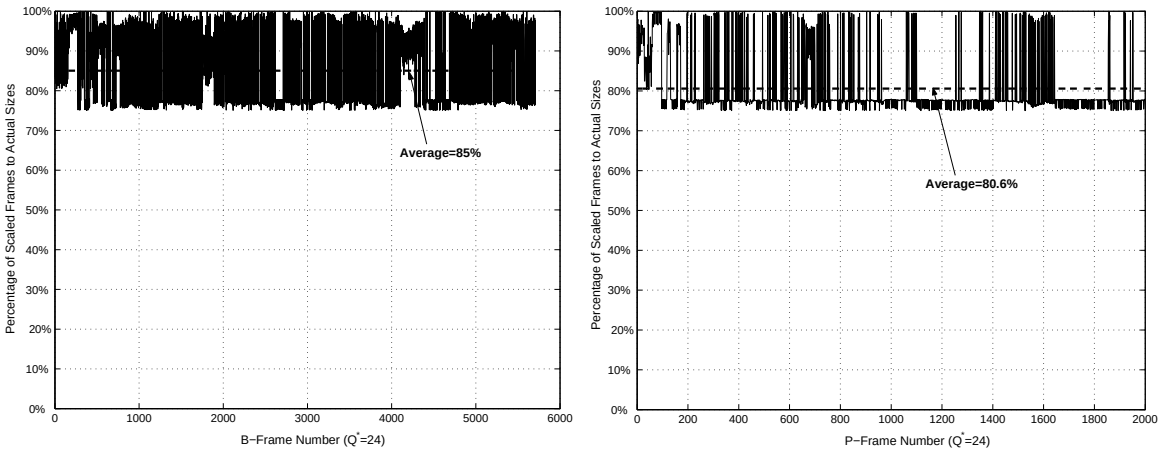
	Scheme 1 without concealment	Scheme 2 without concealment
I	78.15%	85%
P	80.6%	97%
B	85%	99.58%

TABLE II
COMPARISON OF NORMALIZED SIZES OF SCALED FRAMES.

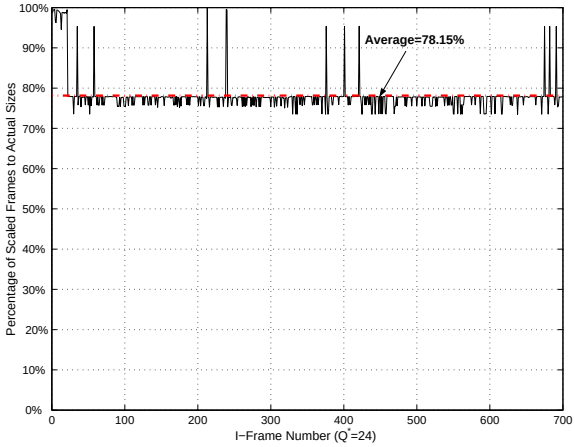
Figure 13 shows a histogram of the normalized sizes of scaled frames for the two schemes. For scheme 2, B frames undergo less scaling than I and P frames. However, it is noticed that some of the I and P frames are drastically scaled down to less than 50% of their predicted/actual sizes. Naturally, severe scaling may substantially degrade the perceived video quality. In order not to degrade the reconstructed video quality, the following simple strategy can be followed. If a frame is to undergo severe scaling, the decoder can notify the encoder to neither send that frame nor reference it when encoding future frames. Simultaneously, the decoder can replace/conceal this frame with a neighboring frame from its frame reservoir. Recall that our target is to keep the buffer occupancy $\geq Q^*$ at all times. Thus, when scheme 2 is used, underflow is very unlikely to happen. This can be seen in Figure 15 where $Q(n) \geq Q^* > 0$. We also observed that for scheme 2 larger frames (typically I frames) sometimes undergo higher scaling than scheme 1.

Figure 14 shows the evolution of the playback buffer for different values of Q^* for scheme 1, with and without concealment. The average normalized sizes of scaled frames are very close for different values of Q^*. But the absolute amounts scaled from the different frames increase with the value of Q^*. This is attributed to how T_c is selected in case of playback buffer underflow or starvation. Note that the values of Q^* used to obtain this graph are arbitrarily selected. When no concealment is employed, scheme 1 is only capable of maintaining the required occupancy around small values of Q^*. In contrast, scheme 2 is always capable of maintaining the desired occupancy for high values of Q^* even without concealment.

Figure 15 shows the evolution of the playback buffer under four different RTT values for scheme 2. We also noticed that the average normalized sizes of scaled frames (not shown) are smaller for high RTT values. Also, the smaller the value of RTT, the higher the capability of the proposed scheme to maintain the desired quality. Figure 16 depicts the percentage of scaled frames versus RTT for scheme 2. This figure shows that for a reasonably small RTT, we can boost the video quality by keeping the number of scaled frames relatively small.



(a) B frames (b) P frames

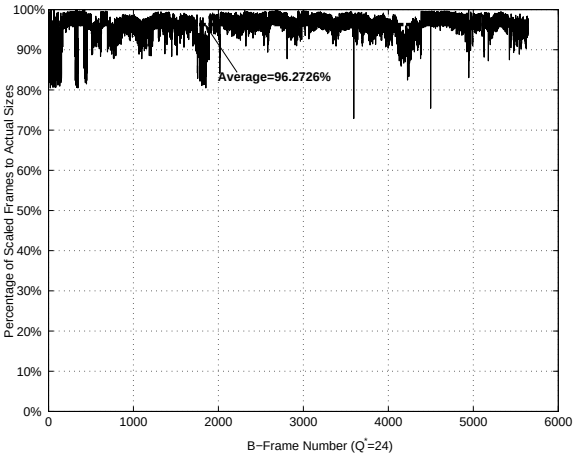


(c) I frames

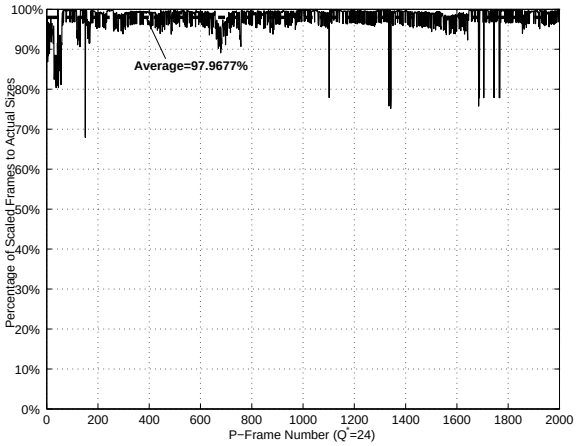
Fig. 11. Normalized sizes of scaled B, P, and I frames using scheme 1 without concealment ($Q^* = 24$).

IV. Conclusions

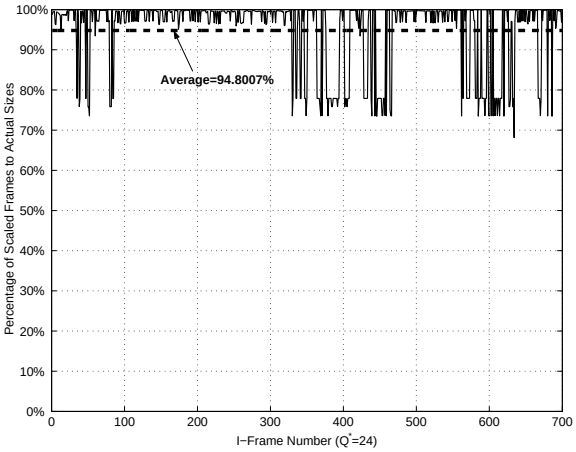
In this paper, we proposed two scalable and adaptive source-channel rate control schemes for video transmission over wireless packet networks. Analytical models were used to maximize the probability of delivering a frame within the critical time T_c. Our analysis exploited the advantages of the ARQ and FEC schemes as well as those of scalable compression formats. Assuming that the channel state does not vary during a frame transmission time, we first provided a probabilistic expression that contains the key parameters of the proposed adaptive model. Using this model, we showed that for each candidate frame, if there is no optimal or near-optimal pair (K^*_{tot} , K^*_{par}) that maximizes the probability of delivering a frame within the critical time T_c, the required probability bound can be achieved by source control. We then introduced another analytical model that takes into account the possibility that the channel can vary during a frame transmission time.



(a) B frames

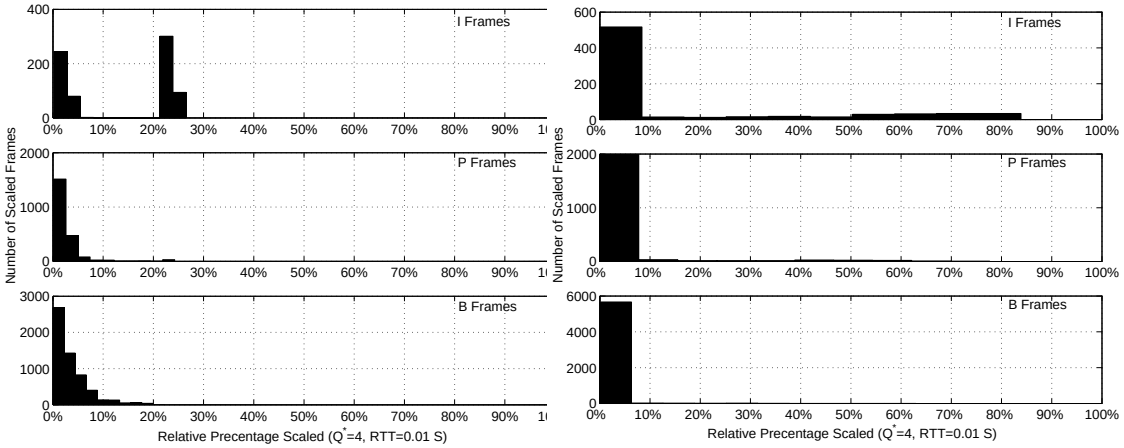


(b) P frames



(c) I frames

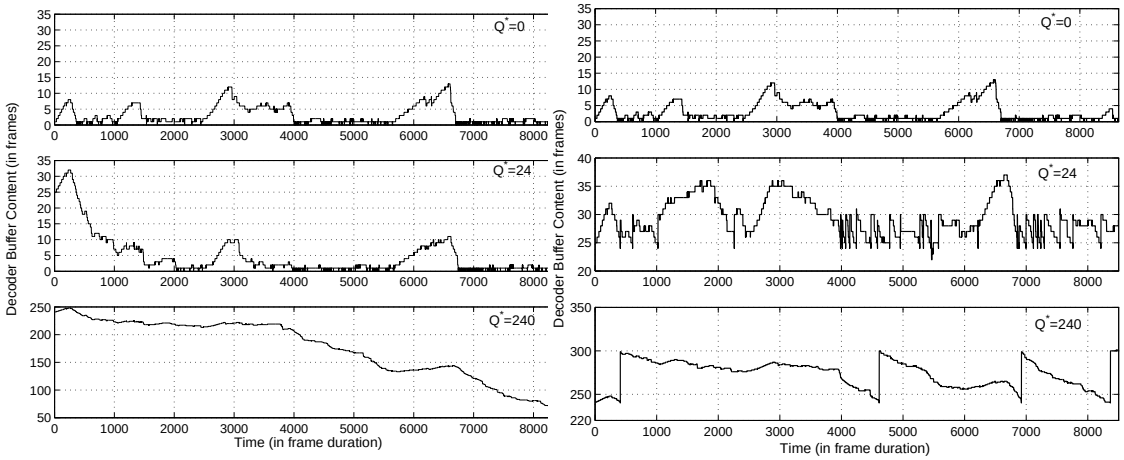
Fig. 12. Normalized sizes of scaled B, P, and I frames using scheme 1 with concealment ($Q^* = 24$).



(a) Scheme 1

(b) Scheme 2

Fig. 13. Normalized sizes of scaled B, P, and I frames using scheme 1 and 2 without concealment ($Q^* = 24$ and RTT=0.01 s).



(a) Without concealment (b) With concealment

Fig. 14. Time evolution of the playback buffer with and without concealment (scheme 1).

Based on that, we provided a time average expression that contains the key parameters of the proposed scalable model. We also introduced a proactive concealment approach that helped in avoiding starvation instants in underflow infatuations. This was achieved by extending the playback time of the available frames in the playback buffer by repeating the display of a subset these frames. Simulation results showed that an arbitrary selection of the number of correctable bits or packet sizes has a negative effect on the relative played back percentage. Simulation results also showed that scheme 2 outperforms all other scenarios. This asserts our argument that graceful quality degradation can be achieved with a less conservative (when compared to scheme 1) but more realistic approach (by allowing channel state to vary during frame transmission time). In a future work, we will study the optimal bit allocation strategy on a frame and macroblock level subject to R-D curves with more sophisticated concealment approaches.

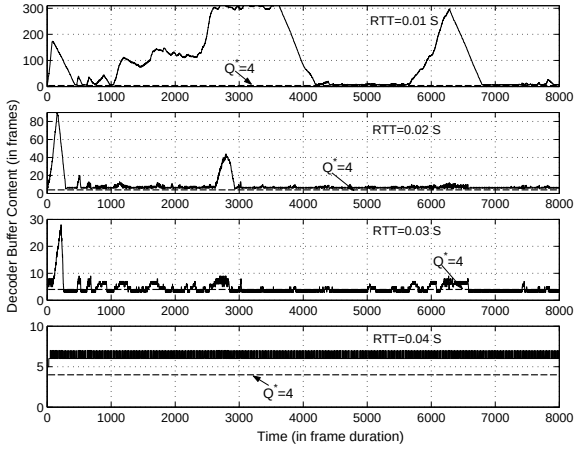


Fig. 15. Time evolution of the playback buffer (scheme 2).

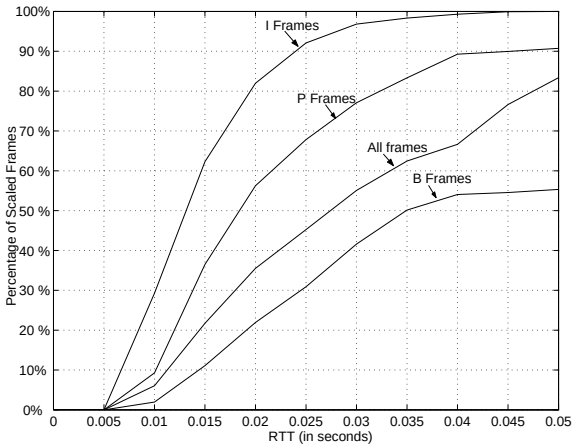


Fig. 16. Percentage of scaled frames versus RTT for scheme 2.

REFERENCES

[1] S. Aramvith, C. Lin, S. Roy, and M. Sun. Wireless video transport using conditional retransmission and low-delay interleaving. *IEEE Trans. on Circuits and Systems for Video Tech*, pages 558–565, June 2002.

[2] S. Aramvith, I. Pao, and M. Sun. A rate-control scheme for video transport over wireless channels. *IEEE Trans. on Circuits and Systems for Video Tech*, pages 569–580, May 2001.

[3] I. Busse, B. Deffner, and H. Schulzrinne. Dynamic QOS control of multimedia applications based on RTP. In *First Int'l Workshop on High Speed Networks and Open Distributed Platforms, St. Petersburg, Russia*, June 1995.

[4] J. Cabrera, A. Ortega, and J. Ronda. Stochastic rate-control of video coders for wireless channels. *IEEE Trans. on Circuits and Systems for Video Tech*, pages 496–510, June 2002.

[5] K. Chawla, Z. Jiang, X. Qiu, and A. Reibman. Transmission of streaming video over an EGPRS wireless network. In *Proceedings of the IEEE International Conference on Multimedia and Exhibition*, NEW YORK NY. IEEE, July 2000.

[6] P. Cuetos and K. Ross. Adaptive rate control for streaming stored fine-grained scalable video. In *Proceedings of the NOSSDAV'02*, pages 3–12, Miami, FL, USA, May 2002.

[7] P. Cuetos and K. Ross. Optimal streaming of layered video: Joint scheduling and error concealment. In *Proceedings of the ACM International Conference on Multimedia*, pages 55–64, Berkeley, CA, USA, November 2003.

[8] R. Deng. Hybrid ARQ schemes employing coded modulation and sequence combining. *IEEE Trans. on Commun.*, pages 2239–2245, Feb.-Apr. 1994.

[9] M. Gallant and F. Kossentini. Rate-distortion optimized layered coding with unequal error protection for robust internet video. *EEE Trans. on Circ.and Sys. for Video Tech.*, 11:357–372, March 2001.

[10] M. Ghanbari. Two-layer coding of video signals for vbr networks. *IEEE J. Select. Areas Commun.*, 7(5):771–781, June 1989.

[11] V. K. Goyal. Multiple description coding: Compression meets the network. *IEEE signal processing magazine*, 18(5):74–93, September 2001.

[12] C.-Y. Hsu, A. Ortega, and A. Reibman. Joint selection of source and channel rate for VBR video transmission under ATM policing constraints. *IEEE J. Select. Areas Commun.*, 15:1016–1028, August 1997.

[13] P.-C. Hu, Z.-L. Zhang, and M. Kaveh. Channel condition ARQ rate control for real-time wireless video under buffer constraints. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 124–127, Vancouver, BC, October 2000.

[14] M. R. Izquierdo and D. S. Reeves. A survey of statistical source models for variable-bit-rate compressed video. *Multimedia Syst.*, 7(3):199–213, 1999.

[15] S. Kallel. Analysis of memory and incremental redundancy ARQ schemes over a nonstationary channel. *IEEE Trans. on Commun.*, 40:1474–1480, 1992.

[16] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos. Joint source-channel coding for motion-compensated dct-based snr scalable video. *IEEE Transactions on Image Processing*, 11(9):1043–1052, Sept. 2002.

[17] H. Lee, T. Chiang, and Y. Zhang. Scalable rate control for MPEG-4 video. *IEEE Trans. on Circuits and Systems for Video Tech*, pages 878–894, Sep. 2000.

[18] H. Liu and M. El-Zarki. Adaptive source rate control for real-time wireless video transmission. *Mobile Networks and Applications, Special Issue: Mobile Multimedia Communications*, pages 49–60, June 1998.

[19] Ortego and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE S. Processing Magazine*, pages 819–845, 1992.

[20] C. Qingyu and K. Subbalakshmi. Joint source-channel decoding for mpeg-4 video transmission over wireless channels. *IEEE J. Select. Areas Commun.*, 21(10):1780–1789, Dec. 2003.

[21] A. Reibman, Y. Wang, X. Qiu, Z. Jiang, and K. Chawla. Transmission of multiple description and layered video over an EGPRS wireless network. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 136–139, Vancouver, BC, October 2000.

[22] J. Ribas-Corbera and S. Lei. Rate control in DCT video coding for low delay video communication. *IEEE Trans. on Circuits Syst. Video Tech*, pages 172–190, Feb 1999.

[23] H. Song, J. Kim, and C.-J. Kuo. Real-time encoding frame rate control for H.263+ video over the internet. In *Signal Processing: Image Communication*, pages 127–148, September 1999.

[24] H. Song and J. Kuo. Rate control for low-bit-rate video via variable-encoding frame rates. *IEEE Trans. on Circuits and Systems for Video Tech*, pages 512–521, April 2001.

[25] W. Tawbi, F. Horn, E. Horlait, and J.-B. Stefani. Video compression standards and quality of service. *The Computer Journal*, 36:43–54, January 1993.

[26] MPEG 4 Video Group. 'MPEG-4 and H.263 video traces for network performance evaluation. http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.

[27] T. Tian, H. Li, J. Wen, J. D., and Villasenor. Priority dropping in network transmission of scalable video. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 400–403, Vancouver, BC, October 2000.

[28] D. A. Turner and K. W. Ross. Optimal streaming of layer-encoded multimedia presentations. In *Proceedings of the IEEE International Conference on Multimedia and Exhibition*, NEW YORK NY, July 2000.

[29] Y. Wang and Q.-F. Zhu. Error control and concealment for video communication: a review. In *Proceedings of the IEEE Computer Society Press*, volume 86, pages 974–997, May 1996.

[30] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. K. Mitra. Rate-distortion optimized mode for very low bit rate video coding and emerging H.263 standard. *IEEE Trans. on Circuits Syst. Video Tech*, pages 182–190, April 1996.