Cycle-Based Rate Control for One-Way and Interactive Video Communications Over Wireless Channels

L. Atzori[§], M. Krunz[§§], and M. Hassan[§§]
[§]Department of Electrical & Electronic Engineering
University of Cagliari, Italy
l.atzori@diee.unica.it
[§§]Department of Electrical & Computer Engineering
University of Arizona, Tucson, AZ 85721
{krunz,mhassan}@ece.arizona.edu

***Abstract*— In this paper, we propose a joint source-rate/channel-code control scheme for streaming video over a dynamic wireless channel. The scheme is designed to maximize the achievable source rate while guaranteeing an upper bound on the probability of starvation at the playback buffer. It can be applied to both *one-way* and *interactive* video communications. Rate control is performed adaptively on a *per-cycle basis*, where a cycle consists of a "good" (non-fading) channel period and the ensuing "bad" (fading) period. This cycle-based approach has two advantages. First, it reduces the fluctuations in the source bit rate, ensuring smooth variations in video quality and avoiding the "saw" effect that is typically observed in frame-based rate control. Second, it makes it possible to derive simple expressions for the starvation probability at the playback buffer, which we use to determine the optimal source rate and channel code for the good and bad periods of the subsequent cycle.**

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

I. Introduction

Recent progress in wireless access technologies combined with the advent of highly efficient, scalable, and error-resilient video compression formats (e.g., MPEG-4, H.264) have made it more possible than ever to stream real-time video over wireless channels. However, there are still some major obstacles to overcome. On the one hand, video applications require sustainable network throughput along with stringent delay-jitter guarantees, particularly for interactive communications. On the other hand, wireless channels are highly dynamic, with a bit error rate (BER) that fluctuates by orders of magnitude in less than a second. 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 extends to several frames (e.g., the loss of a reference frame in MPEG impacts the reception of all subsequent frames until the next reference frame).

An abridged version of this paper was presented at the *ACM Multimedia 2004 Conference*, New York, USA, October 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.

The aforementioned challenges inspired several potential solutions, which can be employed separately or jointly. One traditional yet effective class of solutions focuses on link-layer reliability, i.e., forward error correction (FEC) and/or automatic repeat request (ARQ). "Static" FEC can provide sustained throughput and bounded delay, but when designed for the worst channel conditions, it incurs some unnecessary overhead. Adaptive FEC (code rate varies with channel conditions) is more efficient for a dynamic channel. However, tuning the FEC code rate according to the instantaneous BER in an online fashion is not straightforward. ARQ techniques have also been used, although their effectiveness in ensuring strict delay guarantees is limited. Hybrid ARQ schemes (e.g., [15], [9]) are believed to provide the best features of ARQ and FEC, and will therefore be considered in our work.

Another class of solutions is based on source-rate control, often performed at the frame level (e.g., [23], [17]) or the macroblock level (e.g., [25], [21]). Such solutions exploit the scalability offered by recently standardized compression formats. Several of these rate-control solutions have been targeted towards erroneous channels (e.g., [19], [7], [20], [14], [3], [2]; see also [26], [27], [28] and the references therein). The authors in [13] 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 operation. In [14] the authors introduced a rate-control scheme that aims at maximizing the channel utilization subject to a constraint on the playback buffer size. They suggested minimizing the probability of buffer underflow by equating the effective input and output rates of the playback buffer. The authors in [3] studied the rate control problem from the sender's point of view, and proposed rate control schemes that avoid the degradation in the peak signal to noise ratio (PSNR) due to the reduction in the bit rate. A conditional retransmission and low-delay interleaving scheme was proposed in [2], in which the encoder buffer is used as part of the interleaving memory. In [5] the authors introduced a rate control mechanism based on a priori stochastic models of the source and the underlying channel. In their problem formulation, they divided the optimization

process into 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 vast majority of research on transporting video over wireless channels has been aimed at optimizing the performance of the source and/or channel encoders, with little accommodation of the networking aspects. For instance, a primary goal of many the previous studies was to optimize the effective channel throughput, without taking into account the impacts of the source and channel codes on the transport delay. Furthermore, such studies often ignore the dynamics of the playback buffer, which are critical to maintaining continuous video playback. In addition, some of these schemes are computationally intensive, making them unattractive for real-time operation.

In this paper, we propose a joint source-rate/channel-code control scheme for transporting variable-bit-rate (VBR) encoded video over wireless channels. The scheme is designed to adapt the source bit rate and channel-code parameters on a *per-cycle basis* while guaranteeing an upper bound on the probability of starvation at the playback buffer. A cycle here refers to the combination of a "good" (non-fading) channel period and the following "bad" (fading) period. Our cycle-based approach has two main advantages. First, it reduces the fluctuations in the source rate, resulting in smooth variations in video quality. Second, it allows us to derive simple expressions for the starvation probability at the playback buffer, which we use in determining the optimal source rates for the good and bad periods of the next cycle. Because of its low computational complexity, the proposed scheme is attractive for real-time operation. We apply this scheme to both *one-way* and *interactive* video. The main difference between the two cases is in an additional constraint on the the end-to-end delay for interactive video, which translates into a bound on the probability of starvation at the *transmitter buffer*.

II. System Model

A. Framework

We consider the transport of VBR-coded video over a dynamic wireless channel. Compression may be done in real time or it may be performed offline (archived video). In the former scenario, the frame encoding and playback rates are both equal to R_f frames/second, whereas in the latter case, a video server may transport pre-encoded frames over the channel at a rate R_a (in frames/second) that may be different from the playback rate R_f. For both cases, we assume a finely scalable compression format, whereby the size of the compressed frame in bits can be reduced as needed at the expense of lowering the frame's spatial quality. At a fixed frame generation rate, say R_f, scaling down the sizes of the compressed frames amounts to adjusting the offered source bit rate R_s.

Without loss of generality, consider the case when frames are generated in real time. According to the observed channel state and the current occupancy of the playback buffer, *the receiver* executes the rate-control and channel-code optimization procedure described in the next section, and sends its feedback to the transmitter. For simplicity, we assume that the "feedback" channel is highly reliable, so that control messages arrive at the transmitter free of errors. Such a simplification is not so unrealistic, as these control messages tend to be small (e.g., tens of bytes), and hence can be strongly protected using a reasonable amount of error-correcting (FEC) code. The receiver's feedback is used by the transmitter to adjust the size of the next generated frame and determine the appropriate values for the channel-code parameters. The scaled frame is then moved to the transmitter buffer. Let S be the size of this frame in bits. When such a frame is to be transmitted over the wireless link, it is first segmented into $N = \lceil S/k \rceil$ link-layer (LL) packets, where k is the number of information bits in each LL packet. Each LL packet contains h error-correcting bits, for a total of $n = k+h$ bits. We assume that the values of k and h are fixed for all LL packets that belong to the same video frame (in fact, and as explained in section IV, both values depend on the observed channel state at the receiver). For any given FEC coding scheme (e.g., BCH codes) and a given pair (n, k), a maximum number of correctable bit errors per packet ($E_{max} = E_{max}(n, k)$) can be easily determined. In addition to the FEC code, each packet is assumed to include a very strong error-detecting code (e.g., CRC-32). In the case of an FEC decoding failure (e.g., packet contains more than E_{max} bit errors), the CRC code will detect the presence of a packet error and will trigger a packet retransmission. We assume a stop-and-wait ARQ policy. This assumption is justifiable when the round-trip time (RTT) is much smaller than the packet transmission time, as is the case in typical wireless LAN environments.

The wireless channel is modeled using a 2-state (good/bad) continuous-time stochastic process in which the sojourn times for the good (g) and bad (b) periods are gamma distributed. Let Γ_g and Γ_b denote these sojourn times. Two-state channel models, in general, have been widely used in the literature, but often under Markovian assumptions (i.e., exponentially distributed sojourn times). They are regarded as a reasonable first-order approximation of slowly varying fading channels [6] (see [24] for a discussion of more general Markov channel models). Recently, some studies (e.g., [1], [10], [11]) have questioned the appropriateness of the Markovian assumption in the two-state model and suggested the use of lognormal and power-law distributions for the sojourn times. For analytical tractability, we consider in our study a gamma distribution for both the good and bad periods, which has roughly the same shape as the lognormal distribution (note also that the exponential distribution is a special case of the gamma distribution). We emphasize here that while the use of such a distribution is meant to provide a better representation of the wireless medium, it has no bearing on our specific rate-control design. For $i \in \{g, b\}$, let p_i be the BER during state i ($p_g \ll p_b$). Let α^{-1} and γ be, respectively, the scale and shape parameters of the gamma distribution of the good period, and let β^{-1} and δ be such parameters for the bad period

(without loss of generality, we assume that δ and γ take integer values). These parameters, which correspond to the means and variances of the good and bad periods, can be determined off-line using representative channel measurements or they can be estimated online at the receiver using, for example, time series models. Let N_i, n_i, h_i, and $E_{max,i}$ be the corresponding N, n, h, and E_{max} values for the ith channel state, $i = g, b$. The optimization of these values is discussed in section IV.

Once all the LL packets of a particular frame have correctly arrived at the receiver (possibly after several retransmission attempts), the frame is assembled and moved to the playback buffer. Let R_{in} be the input *frame rate* into this buffer. In general, R_{in} varies according to the channel conditions, the FEC code, and the source rate. The playback buffer is drained at a fixed rate of R_f (in frames/second).

We assume that the video session starts with a preloading phase in which ΔN frames are first prefetched and decoded before playback commences. These frames correspond to $\Delta T = \Delta N / R_f$ seconds of delay between the time the first frame is generated and the time it is played back. We consider two *real-time* communications scenarios: one-way and interactive. In the first scenario, there is no stringent constraints on ΔT, while in the second scenario, the application imposes a strick upper bound on ΔT ($\Delta T \leq 150$ ms). In the case of archived video, there is no constraint on ΔT, which can be changed during the streaming by varying R_a. We treat this scenario as a special case of the one-way real-time video communication.

Figure 1 depicts the behavior of the transmission and playback buffers. Because the frame generation and playback rates are both equal to R_f, we have

$$\Delta N = Q_e(t) + Q_p(t) \quad \forall t \tag{1}$$

where $Q_e(t)$ and $Q_p(t)$ are the encoder and playback buffer occupancies at time t. In other words, the number of frames in the system is constant at any time, and what varies is the distribution of ΔN between the transmitter and playback buffers.

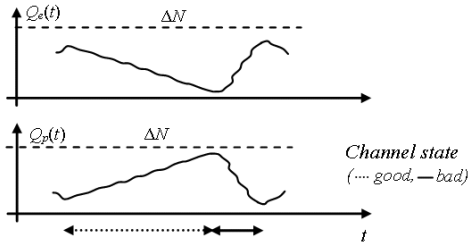


Fig. 1. $Q_e(t)$ and $Q_p(t)$ versus time.

B. Overview of the Proposed Scheme

The main goal of our joint rate-control/channel-code scheme is to maintain the continuity of the playback process by limiting the likelihood of playback-buffer starvation, while simultaneously maximizing the spatial quality. To do that, we allow the playback buffer to build up during the good state and to shrink (and occasionally starve) during the bad states. Infrequent events of buffer starvation (e.g., once every 10000 frames) will have negligible impact on video quality, and can be easily compensated for by means of frame concealment. By allowing for such events to occur, we are able to achieve a significant improvement in the overall video quality by maximizing the source rate.

Following the preloading phase, the receiver starts executing the rate-control and channel-code-optimization procedures. It continuously monitors the channel and accordingly determines its state (*g* or *b*). Several channel-state-estimation approaches are available in the literature (e.g., [16], [8], [18]), and any of them can be used in our work. For example, level-crossing analysis can be used to partition the continuous SNR range into a discrete number of sets, each of which corresponds to one channel state (see [12] and the references therein). Once the partitioning is done, the receiver can map the measured SNR into an appropriate channel state. Another common channel-state estimation technique relies on the observed number of bit errors in the packet and the last observed channel state. Specifically, suppose that the channel was previously determined to be in state i, $i \in \{g, b\}$. Let l be the number bit errors observed in an N_i-bit packet before FEC decoding. Note that if $l \in [0, E_{max,i}]$, all bit errors in the packet are correctable. The goal is to determine the most likely channel state that the received packet has encountered. To do that, the chi-squared test is applied to establish whether the channel is still in state i (hypothesis H_0) or has switched to the other state (hypothesis H_1). The receiver computes the chi-squared value from l and the associated probability P_{χ^2} of having a chi-squared value equal to or greater than l by chance only. The receiver then rejects hypothesis H_0 if P_{χ^2} is lower than a given significance level and decides that the channel has changed to the other state. The goodness of this approximation technique can be assessed using the probability of an erroneous detection. For example, if the channel was previously in the good state, then the probability of making a wrong detection decision is given by the sum of the probability of false detection ($P_{g,f}$) and that of omitted detection ($P_{g,o}$):

$$P_{g,f} = \sum_{j \geq Th_g} \binom{n_g}{j} p_g^j (1-p_g)^{n_g - j} \tag{2}$$

$$P_{g,o} = \sum_{j \leq Th_g} \binom{n_g}{j} p_b^j (1-p_b)^{n_g - j} \tag{3}$$

where Th_g is the threshold for accepting or rejecting H_0 (if $l \leq Th_g$, the channel is determined to be in the good state; otherwise, it is determined to be in the bad state). A similar treatment is used if the channel was previously in the bad state (a different threshold, Th_b, is used in this case). In section VI we provide numerical examples of the resulting probability of error under this estimation approach.

Based on the continuously estimated channel state, the receiver determines when the channel has switched from bad

to good. This signals the start of a new cycle[1]. At that time, the receiver computes the "optimal" values of the source rate R_s and the channel-code parameters that maximize the frame's spatial quality while simultaneously guaranteeing a bound ϵ on the starvation probability of the playback buffer. These values are sent back to the transmitter, which uses them to adjust the frame sizes and FEC code for the subsequent cycle. Note that when frames are encoded on-the-fly, fixing R_s over a cycle implies that the quantization level will vary on a frame-by-frame basis even during the same cycle. The impact of that on video quality is examined in section VI.

III. Analysis of Starvation Probability

In this section, we derive analytical expressions for the starvation probability of the playback buffer. Such expressions are later used in optimizing R_s and the channel-code parameters. Consider an arbitrary cycle. To simplify the analysis, we treat the frame arrival and playback processes as continuous-time functions (i.e., we use a fluid approximation to characterize the dynamics of the playback buffer). It is then reasonable to assume that $Q_p(t)$ increases during the good period and decreases during the bad period (see Figure 1). The validity of this assumption will be experimentally verified in Section VI. So if the playback buffer is to starve anytime during the cycle, it must also starve by the end of the bad period of that cycle. The continuity of the fluid-flow approximation implies that only one starvation period (consisting of one or more frame intervals) can occur in one cycle. If such starvation period is to occur, it must include the end of the cycle. In other words, a starvation interval occurs in a given cycle if and only if such an interval includes the end of that cycle. Let Φ be the probability of a starvation event at the end of an arbitrary cycle. In our design, we require that $\Phi \leq \epsilon$, where $0 < \epsilon \ll 1$ is a predefined constraint.

A. One-Way Video Communication

In this scenario, ΔT can be large enough to avoid starvation of the *transmission* buffer. Consider an arbitrary cycle. Let Q_0 be the number of frames in the playback buffer at the start of the cycle. Starvation occurs if Q_0 plus the number of frames transmitted during the cycle is less than the number of frames played back during the same cycle. Formally, starvation occurs if:

$$Q_0 + \Gamma_g \bar{R}_g + \Gamma_b \bar{R}_b < (\Gamma_g + \Gamma_b) R_f \tag{4}$$

where $\bar{R}_g$ and $\bar{R}_b$ are *the average rates* at which frames are correctly received during the good and bad periods of the underlying cycle, respectively. As explained later, these rates can be controlled by appropriately setting the source rate and channel-code parameters. The constraint on the probability of starvation is then given by:

$$\Phi \stackrel{\text{def}}{=} \Pr\left[Q_0 + \Gamma_g \bar{R}_g + \Gamma_b \bar{R}_b < (\Gamma_g + \Gamma_b) R_f\right] \leq \epsilon. \tag{5}$$

[1]Without loss of generality, we assume that the cycle starts with the good period and is followed by the bad period.

To simplify the analysis, we rewrite the condition in (4) as $(R_f - \bar{R}_b)\Gamma_b > (\bar{R}_g - R_f)\Gamma_g + Q_0$ and consider the following possible cases:

- **Case 1:** $\bar{R}_g \geq R_f \geq \bar{R}_b$. This is the most common case. Intuitively, in this case the playback buffer is expected to build up during the good periods and to shrink during the bad ones.
- **Case 2:** $\bar{R}_g \leq R_f$ **and** $\bar{R}_b \leq R_f$. A solution is possible in this case when Q_0 is considerably large. Since $\bar{R}_g$ and $\bar{R}_b$ are smaller than R_f, the playback buffer will shrink throughout the entire cycle.
- **Case 3:** $\bar{R}_g \geq \bar{R}_b \geq R_f$. Since the average rates $\bar{R}_g$ and $\bar{R}_b$ are higher than R_f, the buffer occupancy is expected to increase throughout the entire cycle. In this case, Φ will be zero irrespective of the initial buffer occupancy.

Note that the case $(\bar{R}_g - R_f) < 0$ and $(R_f - \bar{R}_b) < 0$ is an impossible one. Only the first and second cases are of practical significance. So, we compute Φ for these two cases. Let $\tilde{\Gamma}_g \stackrel{\text{def}}{=} (\bar{R}_g - R_f)\Gamma_g$ and $\tilde{\Gamma}_b \stackrel{\text{def}}{=} |R_f - \bar{R}_b|\Gamma_b$. Consider the first case. Starvation occurs when $\tilde{\Gamma}_b > \tilde{\Gamma}_g + Q_0$, which corresponds to the gray area in Figure 2-a. The probability of starvation in this case is given by:

$$\Phi = \int_0^\infty \left(\int_{Q_0+x}^\infty f_{\tilde{\Gamma}_b}(y)dy \right) f_{\tilde{\Gamma}_g}(x)dx \tag{6}$$

where $f_{\tilde{\Gamma}_i}(.)$, $i \in \{g, b\}$, is the pdf of $\tilde{\Gamma}_i$ and is easily obtained from $f_{\Gamma_i}(.)$. By defining $\tilde{\alpha} = |\bar{R}_g - R_f|\alpha$ and $\tilde{\beta} = (R_f - \bar{R}_b)\beta$, Φ can be written as:

$$\int_0^\infty \left(\int_{Q_0+x}^\infty \frac{\left(\frac{y}{\tilde{\beta}}\right)^{\delta-1}}{(\delta-1)!\tilde{\beta}} e^{-\frac{y}{\tilde{\beta}}} dy \right) \frac{\left(\frac{x}{\tilde{\alpha}}\right)^{\gamma-1}}{(\gamma-1)!\tilde{\alpha}} e^{-\frac{x}{\tilde{\alpha}}} dx. \tag{7}$$

Solving this integral, we obtain (detailed solution is shown in the appendix):

$$\Phi = \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \left[\sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1}(\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \sum_{j=0}^{i} \binom{i}{j} Q_0^{i-j} \frac{(j+\gamma)!}{\theta_1^{\gamma+j}} \right] \tag{8}$$

where $\theta_1 \stackrel{\text{def}}{=} \frac{1}{\tilde{\beta}} + \frac{1}{\tilde{\alpha}}$. As for the second case, starvation occurs when $\tilde{\Gamma}_b > -\tilde{\Gamma}_g + Q_0$, as shown in Figure 2-b. The probability of starvation is given by:

$$\begin{aligned} \Phi &= \int_0^{Q_0} \left(\int_{Q_0-x}^\infty f_{\tilde{\Gamma}_b}(y)dy \right) f_{\tilde{\Gamma}_g}(x)dx \\ &+ \int_{Q_0}^\infty \left(\int_0^\infty f_{\tilde{\Gamma}_b}(y)dy \right) f_{\tilde{\Gamma}_g}(x)dx. \end{aligned} \tag{9}$$

Solving this integral, we obtain (detailed solution is shown in the appendix):

$$\begin{aligned} \Phi &= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1}(\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \\ &\times \sum_{j=0}^{i} \binom{i}{j} (-1)^i (-Q_0)^{i-j} \frac{(\gamma+j)!}{\theta_2^{\gamma+j}} \end{aligned}$$

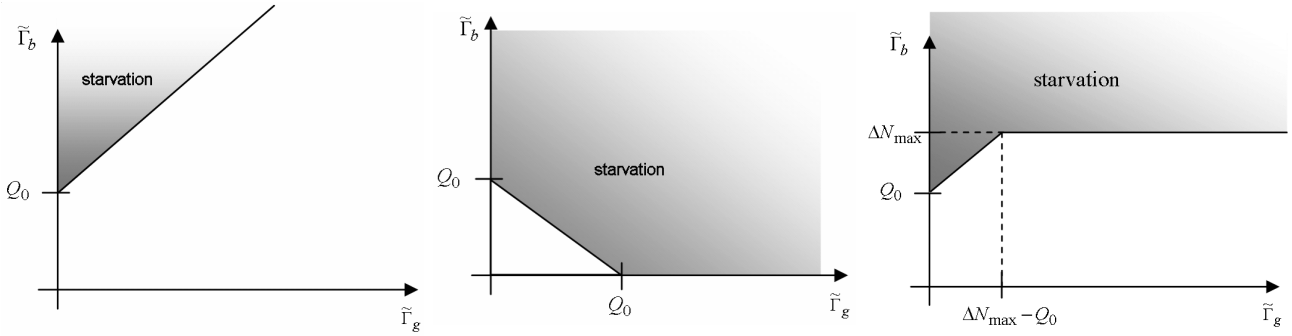


Fig. 2. Ranges of values that result in starvation: (a) Case 1 for one-way communications, (b) Case 2 for one-way and interactive communications, (c) Case 1 for interactive communications.

$$\times \quad \frac{(\gamma+j-1)!}{\theta_2^{\gamma+j}}\left(1-e^{-\theta_2 Q_0}\sum_{k=0}^{\gamma+j-1}\frac{(\theta_2 Q_0)^k}{k!}\right)$$
$$+ \quad e^{-\frac{Q_0}{\tilde{\alpha}}}\sum_{i=0}^{\gamma-1}\frac{(\frac{Q_0}{\tilde{\alpha}})^i}{i!} \tag{10}$$

where $\theta_2 \stackrel{\text{def}}{=} \frac{1}{\tilde{\beta}} - \frac{1}{\tilde{\alpha}}$.

B. Interactive Video Communication

In this section, we compute the probability of starvation at the playback buffer for the case of interactive video. In the case when $\bar{R}_g \geq R_f$, it is possible for the transmitter buffer to starve. Note that this happens only in case 1 of Section III-A. To compute Φ for this case, we redefine (5) as follows:

$$\Phi \stackrel{\text{def}}{=} \begin{cases} \Pr\left[Q_0 + \tilde{\Gamma}_g - \tilde{\Gamma}_b < 0\right], & \text{if } \tilde{\Gamma}_g + Q_0 < \Delta N \\ \Pr\left[\Delta N - \tilde{\Gamma}_b < 0\right], & \text{otherwise} \end{cases} \tag{11}$$

Accordingly,

$$\Phi = \int_0^{\Delta N - Q_0}\int_{Q_0+x}^{\infty} f_{\tilde{\Gamma}_b}(y) f_{\tilde{\Gamma}_g}(x)\,dy\,dx + \int_{\Delta N - Q_0}^{\infty}\int_{\Delta N}^{\infty} f_{\tilde{\Gamma}_b}(y) f_{\tilde{\Gamma}_g}(x)\,dy\,dx. \tag{12}$$

Solving this integral, we obtain:

$$\Phi = \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}^{\gamma}(\gamma-1)!}\sum_{i=0}^{\delta-1}\frac{1}{i!\tilde{\beta}^i}\sum_{j=0}^{i}\binom{i}{j}Q_0^{i-j}\frac{(\gamma+j-1)!}{\theta^{\gamma+j}}$$
$$\times \left[1-e^{-\theta(\Delta N-Q_0)}\sum_{k=0}^{\gamma+j-1}\frac{(-\theta(\Delta N-Q_0))^k}{k!}\right]$$
$$+ e^{-\frac{\Delta N}{\tilde{\beta}}}\sum_{i=0}^{\delta-1}\frac{(\Delta N)^i}{i!\tilde{\beta}^i}$$
$$\times e^{-\frac{\Delta N - Q_0}{\tilde{\alpha}}}\sum_{j=0}^{\gamma-1}\frac{(\Delta N-Q_0)^j}{j!\tilde{\alpha}^j}. \tag{13}$$

Note that this expression is similar to (8) but with an additional term that is a function of ΔN. As expected, as ΔN increases Φ decreases.

IV. Adaptive Computation of Source Rate and Channel-Code Parameters

In the previous section, Φ was expressed as a function of $\tilde{\alpha} = |\bar{R}_g - R_f|\alpha$ and $\tilde{\beta} = (R_f - \bar{R}_b)\beta$. The parameters R_f, α, and β are fixed, so Φ is essentially a function of $\bar{R}_g$ and $\bar{R}_b$. The parameters to be optimized are the source rate R_s, and the channel-code parameters n_g, h_g, n_b, and h_b. So we must first express $\bar{R}_g$ and $\bar{R}_b$ in terms of R_s, n_g, h_g, n_b, and h_b. This is done as follows. Conditioned on channel state i, $i \in \{g, b\}$, the probability that a received LL packet contains a correctable error is given by:

$$P_i = \sum_{j=0}^{E_{max,i}}\binom{n}{j}p_i^j(1-p_i)^{n-j}, \quad i \in \{g,b\}. \tag{14}$$

This quantity depends on n_i, h_i, and the employed FEC scheme. The average number of transmission attempts required to successfully transmit a packet is given by:

$$L_i = \frac{1}{P_i}, \quad i \in \{g,b\}. \tag{15}$$

The time needed to transmit one LL packet is given by:

$$T_i = \frac{n_i}{C} + \text{RTT}, \quad i \in \{g,b\}$$

where C is the channel bit rate. Hence, the average number of packets N_i per frame during state i is given by:

$$N_i = \frac{R_s}{(n_i - h_i)R_f}, \quad i \in \{g,b\}. \tag{16}$$

Accordingly,

$$\bar{R}_i = \frac{1}{N_i L_i T_i} = \frac{(n_i-h_i)R_f}{L_i T_i R_s} = \eta_i \frac{R_f}{R_s}, \quad i \in \{g,b\} \tag{17}$$

where

$$\eta_i \stackrel{\text{def}}{=} \frac{n_i - h_i}{L_i T_i} \tag{18}$$
$$= \frac{(n_i-h_i)\sum_{j=0}^{E_{max,i}}\binom{n_i}{j}p_i^j(1-p_i)^{n_i-j}}{n_i/C + \text{RTT}} \tag{19}$$

To determine the optimal channel-code parameters and optimal R_s, we exploit the *decomposability* of the source/channel sub-systems. First, note that Φ is inversely proportional to

$\bar{R}_g$ and $\bar{R}_b$ (the higher the frame arrival rate at the receiver, the less likely that the playback buffer will starve), and both parameters are inversely proportional to R_s. In other words, Φ is monotone in R_s. This is shown in Figure 3 for different values of Q_0 (part (b) of the figure depicts the trend when Φ is below 10^{-4}, which is typical for real-time streaming applications). As for the channel-code parameters, Φ depends on them exclusively through η_g and η_b. So one can optimize the channel-code parameters by maximizing η_i, $i \in \{g, b\}$, independent of R_s, subject to the constraint $(n_i - h_i)R_f \leq R_s$ (which ensures that a frame consists of at least one LL packet). This is done as follows. First, note that η_i depends only on the BER, the transmission rate, the amount of FEC, and the RTT. In typical streaming scenarios over wireless links, where packet sizes range from 800 to 8000 bits, access bandwidth from 100 to 1000 kbps, and transmitter-receiver separation is up to 1 Km, the RTT is quite negligible compared to the packet transmission time. So we can safely ignore it when computing the link throughput η_i. Accordingly, $\eta_i \approx C(n_i - h_i)/(L_i n_i) = C\xi_i$, where $\xi_i \stackrel{\text{def}}{=} \frac{n_i - h_i}{L_i n_i}$ is called the *line efficiency* for state i. Hence, η_i can be maximized by maximizing ξ_i with respect to n_i and h_i, which can be done offline for a given FEC scheme and for several BER values. For a given BER, the optimal values for n_g, h_g, n_b, and h_b can be obtained using a simple table-lookup. At these optimal values, R_s can be maximized subject to satisfying the given bound on Φ. Because of the (inverse) monotonic relationship between R_s and Φ, the optimal R_s can be easily found using a binary search conducted in logarithmic time. Figure 4-a depicts the behavior of η_i as a function of $E_{max,i}$ using a fixed packet size and BCH code. It is obvious that there is an optimal point associated with each given BER. Figure 4-b depicts η_i versus n_i for a fixed $E_{max,i}$. The plot can be used to maximize η_i with respect to n_i for a given BER.

V. Selection of ΔT in One-way Video Communications

For real-time video, ΔT is selected by the client before the commencement of the video session and it remains constant during the entire session. For one-way video, no stringent constraints are imposed on ΔT. In this scenario, the proposed control scheme assumes that with an appropriate selection of ΔT it is possible to neglect the encoder buffer starvation. In this section, we mathematically validate this assumption and provide an expression for the minimum allowable value of ΔT. Note that a small ΔT is desirable, as ΔT is the time the client has to wait before the video decoding starts.

Let $\Psi(t, \Delta N)$ denote the probability of buffer starvation *at the encoder* at time t:

$$\Psi(t, \Delta N) \stackrel{\text{def}}{=} \Pr[Q_p(t) \geq \Delta N]. \tag{20}$$

We are interested in computing a threshold (ΔN_{th}) on ΔN that results in a very low probability of starvation at the encoder buffer:

$$\Psi(t, \Delta N_{th}) = \Pr[Q_p(t) \geq \Delta N_{th}] \leq \nu \ \ \forall\, t, \tag{21}$$

where $0 < \nu \ll 1$ is a predefined constraint. It is obvious that starvation at the transmitter can be encountered only in channel cycles with $\bar{R}_g \geq R_f$ (case 1 in Section III-A). In these cycles, the buffer occupancy decreases at the encoder and increases at the receiver during the good period. Accordingly, the highest value for $\Psi(t, \Delta N)$ is encountered at the end of the good periods. Let Ψ^* be this probability:

$$\Psi^* \stackrel{\text{def}}{=} \Pr[Q_0 + (\bar{R}_g - R_f)\Gamma_g \geq \Delta N]. \tag{22}$$

It is computed as follows:

$$\Psi^* = \int_{\Delta N - Q_0}^{\infty} f_{\tilde{\Gamma}_g}(x)dx = e^{-\frac{\Delta N - Q_0}{\tilde{\alpha}}} \sum_{i=0}^{\gamma-1} \frac{1}{i!} \left(\frac{\Delta N - Q_0}{\tilde{\alpha}}\right)^i. \tag{23}$$

Using $\tilde{\alpha} = |\bar{R}_g - R_f|\alpha$ in the above equation and solving the integral we obtain:

$$\Psi^* = e^{-\frac{\Delta N - Q_0}{\alpha R_f(\frac{\xi_g C}{R_s} - 1)}} \sum_{i=0}^{\gamma-1} \frac{1}{i!} \left(\frac{\Delta N - Q_0}{\alpha R_f(\frac{\xi_g C}{R_s} - 1)}\right)^i. \tag{24}$$

Note that Ψ^* is a function of Q_0, which in this context denotes the playback buffer occupancy at the beginning of the underlying cycle. We now need to express Ψ^* for every cycle, independent of Q_0. This is required because ΔN has to be set at the beginning of the streaming session and is kept constant afterwards. To this aim, we compute an upper bound on this probability ($\hat{\Psi}$). From (24), Ψ^* increases as Q_0 increases and as R_s decreases. An upper bound on Ψ^* is obtained by using the maximum buffer occupancy and the minimum source coding bit rate within the range of values allowed by Case 1. The minimum source bit rate is $C\xi_b$. The maximum Q_0 value, denoted by $Q_{0,max}$, is the value that results in $R_s = C\xi_g$. In fact, R_s is an increasing function of Q_0. Accordingly, $Q_{0,max}$ is obtained from (8) with Φ being set to the target starvation constraint ϵ and R_s set to $C\xi_g$. For example, assuming that the sojourn times of the good and bad states are exponentially distributed, ϵ is given by:

$$\epsilon = e^{-\frac{Q_{0,max}}{\beta(1-\xi_b/\xi_g)R_f}} \Rightarrow Q_{0,max} = -\beta \ln(\epsilon)(1 - \xi_b/\xi_g)R_f. \tag{25}$$

From which we obtain an expression for $\hat{\Psi}$:

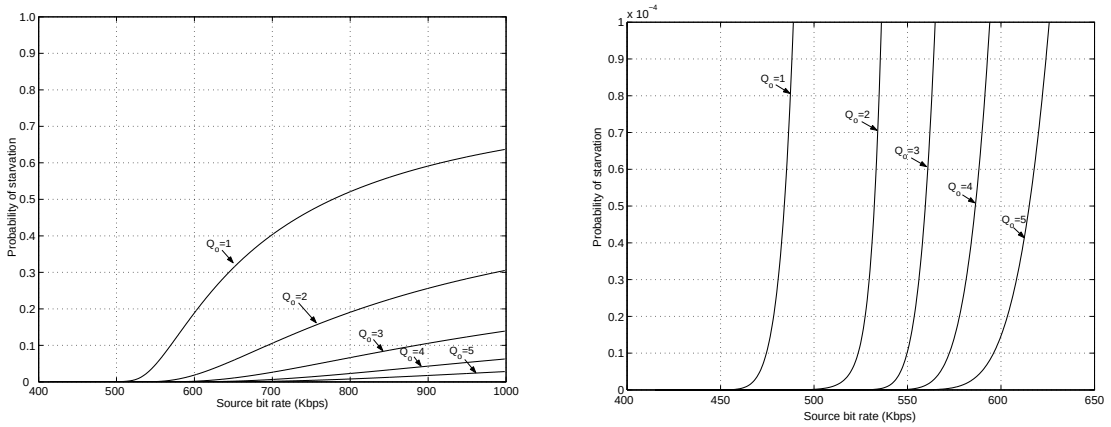
$$\hat{\Psi} = e^{-\frac{\Delta N + \beta \ln(\epsilon)(1-\xi_b/\xi_g)R_f}{-\alpha(\xi_g/\xi_b - 1)R_f}}. \tag{26}$$

In this case, the constraint in (21) can be satisfied by replacing $\Psi(t, \Delta N_{th})$ with its upper bound $\hat{\Psi}$. From this, ΔN_{th} and the corresponding ΔT_{th} are obtained:

$$\Delta N_{th} = -\ln(\nu)\alpha(\xi_g/\xi_b - 1)R_f - \ln(\epsilon)\beta(1 - \xi_b/\xi_g)R_f. \tag{27}$$

$$\Delta T_{th} = -\ln(\nu)\alpha(\xi_g/\xi_b - 1) - \ln(\epsilon)\beta(1 - \xi_b/\xi_g). \tag{28}$$

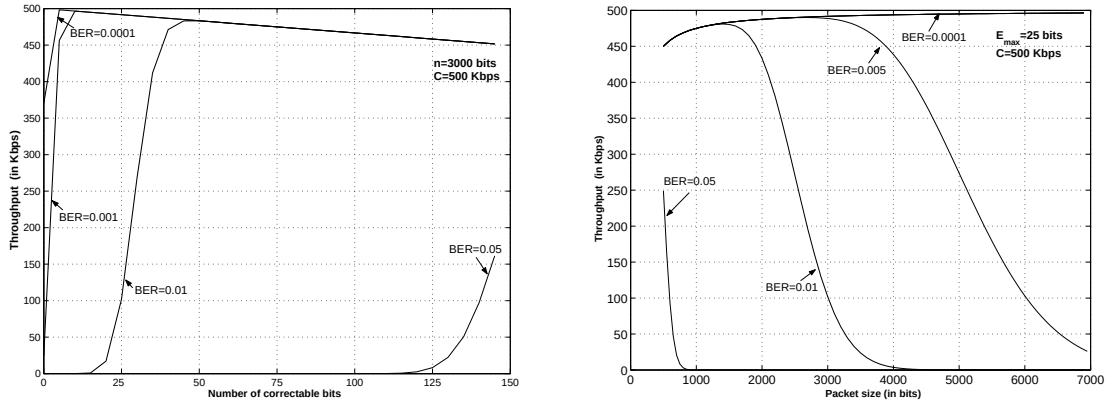
As an example, let $\nu = 10^{-6}$. Using the default values provided in Table I for the channel parameters, the expression of ΔT_{th} becomes $\Delta T_{th} = -0.208 + 0.0117 \ln(\epsilon)$. With ϵ in the range of 10^{-2} to 10^{-8}, ΔT_{th} varies from 300 to 400 msec.



(a) Large-scale variations of Φ

(b) Small-scale variations of Φ

Fig. 3. Probability of starvation versus R_s for different initial buffer occupancy values (Q_0).



(a) η_i versus $E_{max,i}$ (n_i = 3000 bits).

(b) η_i versus n_i ($E_{max,i}$ = 25 bits).

Fig. 4. Link throughput versus channel parameters.

VI. Simulation Results

In this section, we simulate the proposed scheme and contrast its performance with other rate-control schemes. Every simulation experiment was run for 100,000 channel cycles. Table I summarizes the parameter values used in the simulations (when not specified, the default values are used). For the channel code, we use BCH codes with codeword lengths n_i in the range [255-4095] bits. This class of cyclic codes is commonly used because of its low decoding complexity. Within the chosen range of codewords, we consider every possible positive integers m and t of the BCH scheme such that $n_i = 2^m - 1$ and $h_i \leq mt$. The maximum number of correctable bit errors $E_{max,i}$ is then set to t.

TABLE I

Values of parameters used in the simulations

Parameter	Default value	Range
α	100 msec	50-500 msec
β	30 msec	$10 - 100$ msec
p_g	10^{-5}	$10^{-6} - 10^{-4}$
p_b	10^{-2}	$10^{-3} - 10^{-1}$
C	500 kbps	100-1000 kbps
R_f	25 frames/sec	20-30 frames/sec
ϵ	10^{-4}	$10^{-5} - 10^{-3}$
ΔT	100 msec	≥ 80 msec

A. Validation of Channel Model

First, we evaluate the goodness of the 2-state channel model. As discussed in section II-B, one approach to estimate the channel state is based on the observed number of bit errors in a packet. The estimation error using this technique was expressed in (2) and (3) for a good-to-bad state transition. Using the default parameter values in Table I and the corresponding optimal packet sizes $n_g = 511$ bits and $n_b = 4095$ bits (obtained from the channel-code optimization process), we apply the chi-squared test at a significance level of 0.001, which results in threshold values $Th_g = 2$ and $Th_b = 19$ bit errors. These values give rise to the following probabilities of wrong detection: $P_{g,f} = 2 \times 10^{-5}$, $P_{g,o} = 3 \times 10^{-2}$, $P_{b,f} = 6 \times 10^{-4}$, and $P_{b,o} \approx 0$. Note that the effect of falsely detecting the end of a cycle is more severe than that of missing the start of a new cycle. In the first case, the receiver instructs the transmitter to switch from a robust channel code that has been used during the bad state to a lighter code to be used for the wrongly detected good state. Clearly, this increases packet losses and impacts the playback-buffer starvation probability.

The other case results in an unnecessary reduction in the rate R_s, as more-than-needed channel code is being applied, but that has no effect on the starvation probability (it mainly impacts the spatial quality of the displayed video). For the default system parameters, the probability of falsely detecting the start of a cycle (6×10^{-4}) is quite low. Additionally, the resulting packet length during the good period is small, so the negative impact of falsely detecting a new cycle lasts for only a short time period. Note that the likelihood that the end of a cycle is still falsely detected after two or more successive packet transmissions is significantly small.

B. Evaluation of Rate-Control Scheme

To evaluate the efficacy of our scheme, we perform two types of simulations. In the first type, we use a fluid approximation of the packet arrival process. In this case, the throughput during the good and bad periods is set to its average value for each of the two periods. This model reflects accurately our starvation probability analysis. We then consider a "discrete" model, in which we account for the packetization effect. Table II shows the results for one-way video communications, averaged over several runs. The observed starvation probability for the continuous model is very close to the target $\epsilon = 10^{-4}$, confirming the goodness of our analysis. In the case of the discrete model, the observed starvation probability is slightly larger than ϵ, which can be attributed to the variability of the channel throughput in this model (such variability produces unaccounted for buffer starvation instances). Note that the average value of Q_0 is quite small, indicating that starvation can be prevented/controlled with only a small buffer size. To evaluate the quality of the streamed video, we also compute the average value of R_s. For both discrete and continuous simulation models, the average R_s values are very close to the mean channel throughput of 482.5 kbps. This is expected, since streaming is never interrupted. The standard deviations for the two types of simulations are very low. Low variations in R_s are highly desirable, as they reduce the variations in the resulting video quality. To better evaluate this aspect, we compute the absolute difference in the bit rate between adjacent cycles ($|\Delta R_s|$). As shown in Table II, the average value of $|\Delta R_s|$ is even smaller than the standard deviation of R_s.

TABLE II

SIMULATION RESULTS OF THE PROPOSED ALGORITHM FOR ONE-WAY VIDEO COMMUNICATIONS

Parameter	Continuous model	Discrete model
Observed starvation probability	1.1×10^{-4}	2.8×10^{-4}
Average Q_0	0.64 frame	0.66 frame
Standard deviation of Q_0	0.21 frame	0.28 frame
Average R_s	479 kbps	480 kbps
Standard deviation of R_s	16 kbps	21 kbps
Average $\lvert\Delta R_s\rvert$	8 kbps	11 kbps

Figure 5 depicts the temporal behavior of R_s for 200 successive cycles. All the cycles in these simulations ended without starving the buffer. Figure 6 shows the evolution of the buffer *within* a cycle for several successive cycles. The observed trend confirms our previous assertion in section II-B that the buffer occupancy within a cycle reaches its lowest value by the end of that cycle. The figure also shows that for most cycles, the rate-control algorithm selects values for $\bar{R}_g$ and $\bar{R}_b$ that are respectively higher and lower than the frame playback rate R_f. Infrequently, both values are lower than R_f (e.g., 19th and 20th cycles), which happens when the buffer occupancy at the start of a cycle is relatively large.

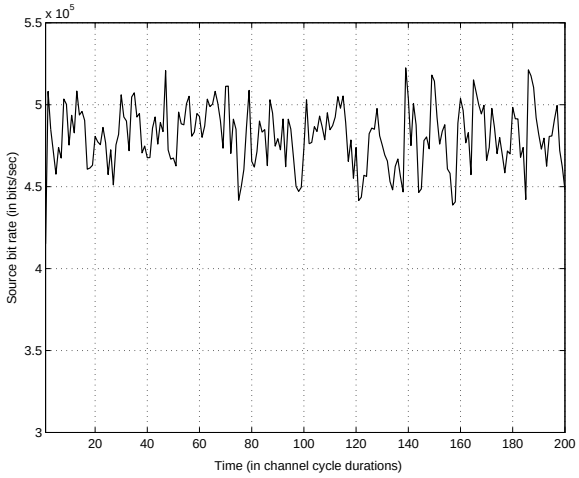


Fig. 5. Source bit rate versus time.

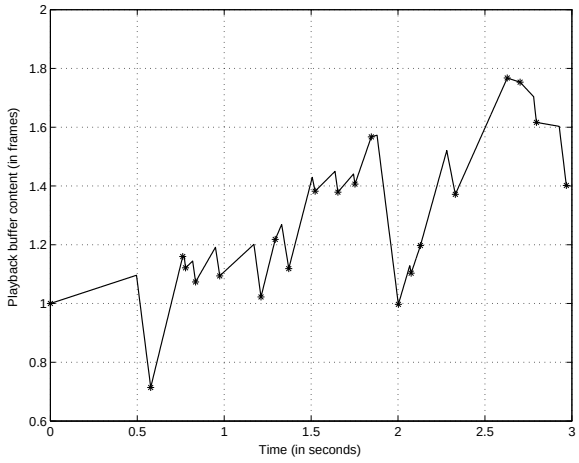


Fig. 6. Playback buffer occupancy evolution *during* a channel cycle (a 'star' indicates the start of a cycle).

Table III depicts the observed starvation probability for various values of α, β, p_b, p_g, C, R_f, and ϵ. When varying the value for a given parameter, the other parameters are set to their default values given in Table I. We observe a slight mismatch between the target and measured values. The mismatch decreases as the values of α, β, and C increase.

Enforcing the constraint $\Phi \leq \epsilon$ on a per-cycle basis sometimes leads to low R_s values, and subsequently high $\bar{R}_g$ values. For interactive (two-way) video, too high $\bar{R}_g$ values can result in starvation at the *transmitter* buffer. Such an

TABLE III

STARVATION PROBABILITY FOR VARIOUS VALUES OF THE SYSTEM PARAMETERS

Parameter	Values Used	Observed Starvation Probability at the Playback Buffer				
α	(50, 80, 100, 200, 500) msec	4.2×10^{-4}	3.4×10^{-4}	4×10^{-4}	2.4×10^{-4}	1.6×10^{-4}
β	(10, 20, 30, 80, 100) msec	6.2×10^{-4}	4.6×10^{-4}	2.6×10^{-4}	3.8×10^{-4}	2.8×10^{-4}
p_g	(10^{-6},5×10^{-6},10^{-5},5×10^{-5},10^{-4})	2.8×10^{-4}	3.2×10^{-4}	2.4×10^{-4}	3.6×10^{-4}	3.4×10^{-4}
p_b	(10^{-3},5×10^{-3},10^{-2},5×10^{-2},10^{-1})	55.2×10^{-4}	7.6×10^{-4}	3.6×10^{-4}	3.4×10^{-4}	2.3×10^{-4}
C	(10^2, 3×10^2, 5×10^2, 8×10^2, 10^3) kbps	3×10^{-4}	2.4×10^{-4}	2.4×10^{-4}	3.8×10^{-4}	3×10^{-4}
R_f	(20, 22, 25, 28, 30) frames/sec	1.6×10^{-4}	4.4×10^{-4}	2.4×10^{-4}	1.6×10^{-4}	2×10^{-4}
ϵ	(10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3})	4×10^{-5}	18×10^{-5}	2.4×10^{-4}	8.6×10^{-4}	2.3×10^{-3}
ΔT	(80, 100, 120, 150, 1000) msec	4×10^{-4}	2.4×10^{-4}	2.8×10^{-4}	3×10^{-4}	2.6×10^{-4}

event is not as unsettling as playback buffer starvation. It mainly impacts (rather slightly) the utilization of the channel. Table IV shows the average number of starvation events per cycle at the encoder for various channel parameters. Note that more than one starvation event can occur during a cycle. A transmitter-buffer starvation event ends when a new frame is generated. During the good periods $\bar{R}_g > R_f$ (on average), so once the newly generated frame is transmitted, the transmitter buffer starves again. From this table, we observe that the average number of starvation events per cycle increases as the lengths of the good and bad periods increase. For the good period, this trend can be justified through (24), which shows that an increase in α leads to an increase in Ψ. As for β, it influences the selection of R_s; low values of R_s are selected when β is large (see (8)). As mentioned before, a decrease in R_s leads to an increase in Ψ. Both C and R_f seem to have little impact on the number of starvation events at the encoder. In general, as R_f increases, the frame generation process becomes more continuous, reducing the resultant Ψ.

Next, we study the performance of the cycle-based rate control scheme when applied to Version JM 8.4 of the emerging JVT/H.264 video coding standard. At the end of each cycle, the value of R_s as determined by our rate-control scheme is used to compute the quantization level (q) for the next cycle. For this we need the relationship between R_s and q, i.e., $R_s(q)$, which is difficult to quantify mathematically. Moreover, different frames will require different q values for the same R_s (which is to be applied to all frames in the next cycle). In our experiments, we empirically determine $R_s(q)$ based on the average q value of the most recent N frames ($N = 10$ in the experiments), each being encoded at a quantization level q. This is done for each q value (i.e., the transform coefficients of every frame in the sequence are quantized several times, each time using a different q). During the operation of the system, when a new frame is to be encoded at rate R', the the quantization level is set to q^*, where $R_s(q^*) < R' < R_s(q^* - 1)$ (note that the higher the value of q, the smaller is the resulting bit rate). Other approaches such as those in [22] could also be used. Tables V and VI depict the resulting performance for two video sequences ("coastguard" and "foreman," in CIF format), each encoded at a rate of $R_f = 30$ frames per second. The simulations were run for 100,000 cycles using different values of ΔT. It can be noted that ΔT has a very negligible impact on image quality, as measured by the PSNR. Also, with $\Delta T = 80$ msec (typical for interactive applications), the difference in the PSNR between interactive and one-way video is only 0.8 dB. In terms of the average q, the difference between the two cases is about 1.7 and 0.2 for the "coastguard" and "foreman" video sequences, respectively.

In the proposed per-cycle scheme, the quantization level q varies on a *per-frame* basis. To evaluate the impact of such variability on video quality, we show in Figure 7 the PSNR for 300 frames ("Foreman" sequence with $\Delta T = 0.1$ msec). We contrast the per-cycle scheme (variable q) with a scheme that uses a fixed quantization level ($q = 17$) that represents the average q value in the per-cycle scheme. Note that even when q is fixed, the PSNR still varies from one frame to another, depending on the scene dynamics and the "richness" of the encoded frame. The two plots exhibit similar temporal behaviors. Relative to a fixed-quantization approach, the per-cycle scheme does not cause noticeably degradation in the PSNR. In fact, the average absolute difference in PSNR between two successive frames is 0.12 dB and 0.099 dB for the cases of variable q and fixed q, respectively. The 300 frames correspond to about 15 channel cycles, so the depicted PSNR values cover multiple cycles.

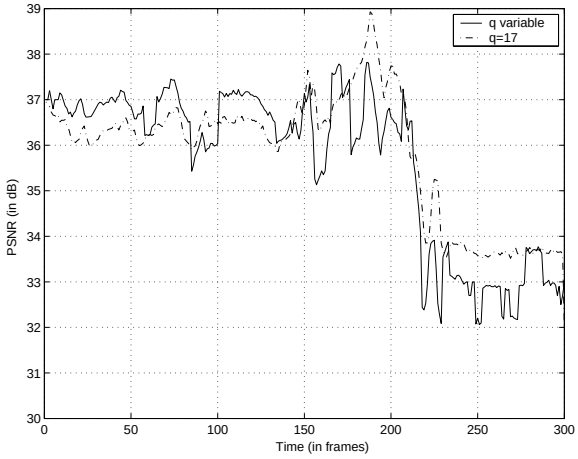


Fig. 7. PSNR versus frame index (Foreman sequence).

Finally, we conduct simulation experiments in which we compare the per-cycle scheme with the TMN8 model [4] as well as with the DDP and SDP [5] algorithms. DDP and SDP

TABLE IV

AVERAGE NUMBER OF STARVATION EVENTS PER CYCLE AT THE ENCODER FOR VARIOUS VALUES OF THE SYSTEM PARAMETERS

Parameter	Values Used	Observed Starvation Events per Cycle at the Encoder				
α	(50, 80, 100, 200, 500) msec	0.00086	0.00738	0.01962	0.11116	0.59206
β	(10, 20, 30, 80, 100) msec	0.00112	0.00436	0.0188	1.501	1.7137
p_g	(10^{-6},5×10^{-6},10^{-5},5×10^{-5},10^{-4})	0.01636	0.01618	0.02	0.019	0.01724
p_b	(10^{-3},5×10^{-3},10^{-2},5×10^{-2},10^{-1})	0.0002	0.00034	0.01964	2.8015	2.712
C	(10^2, 3×10^2, 5×10^2, 8×10^2, 10^3) kbps	0.1373	0.0542	0.0195	0.01292	0.01216
R_f	(20, 22, 25, 28, 30) frames/sec	0.23764	0.0236	0.0195	0.01788	0.01788
ϵ	(10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3})	0.11386	0.0268	0.0195	0.01326	0.01112
ΔT	(80, 100, 120, 150, 1000) msec	0.82424	0.0195	0.00218	0.00038	0

TABLE V

PERFORMANCE RESULTS FOR THE COASTGUARD VIDEO SEQUENCE

ΔT	PSNR	Average q	Channel Utilization	Encoder Starvation Events per Cycle	Φ
0.08	30.603	22.0598	0.95973	3.7109	0.0008
0.1	30.759	21.81	0.98432	3.576	0.00054
0.12	30.862	21.6517	0.93782	3.4258	0.0004
0.15	31.014	21.41	0.95689	2.98	0.00012
one-way	31.448	20.7085	1.0	0.0	4.0×10^{-5}

TABLE VI

PERFORMANCE RESULTS FOR THE FOREMAN VIDEO SEQUENCE

ΔT	PSNR	Average q	Channel Utilization	Encoder Starvation Events per Cycle	Φ
0.08	35.896	17.1842	0.94825	2.5337	0.0004
0.1	35.996	16.996	0.98927	0.39748	0.0012
0.12	36.062	16.8724	0.99734	0.09766	0.0011
0.15	36.145	16.7385	0.99923	0.02566	0.0007
one-way	36.674	16.9716	1.0	0.0	0.0005

are two important rate-control schemes that were developed based on stochastic dynamic programming. To ensure a fair comparison, we conduct our experiments under a comparable setup to that of [5]. In particular, we set $\alpha = 110$ msec, $\beta = 19$ msec, $p_b = 8\times10^{-2}$, $C = 65.5$ kbps, and $R_f = 6$ frames/second. The other parameters are kept at their default values in Table I. This results in the same average sojourn times for the good and bad periods (550 msec and 150 msec, respectively) and the same channel throughputs (η_g and η_b) as those in [5]. Table VII depicts the PSNR and probability of starvation under various rate-control algorithms when applied to the "mother& daughter" sequence (used in [5]). The table reveals the following. In terms of the average PSNR, our cycle-based approach achieves comparable performance to that of the SDP and DDP algorithms, and a slightly better performance than that of the TMN8 algorithm. However, in terms of the observed probability of starvation, the cycle-based approach achieves a much better performance (lower starvation rate) than the three other algorithms. This is expected given that the cycle-based algorithm aims at controlling the starvation probability whereas the SDP and DDP algorithms aim at finding the "optimal" balance between compression and distortion. Note that "optimizing" the average PSNR by itself is not enough to ensure satisfactory perceptual quality, especially if it comes at the expense of frequent buffer starvation instances (which cause undesirable video artifacts).

VII. CONCLUSIONS

In this paper, we proposed a low-complexity source-channel rate control scheme for video streaming over wireless channels. The coding rate is computed by introducing a constraint on the probability of starvation at the playback buffer. We derived an expression for the starvation probability as a function of the coding parameters. To reduce quality variations among neighboring frames, the computed source bit-rate is used by the sender over a complete channel cycle. Experimental results proved the efficacy of the proposed approach in controlling the buffer underflow frequency in both cases of interactive and one-way video communications. From the experiments, we observed some starvation events at the encoder buffer in case of interactive video communications. This is caused by the constraint on the maximum end-to-end delay, resulting in a waste of the channel capacity. However, our algorithm allowed for obtaining channel utilization values lower than 90% only in few scenarios. Slight mismatches are encountered in cases of high variance in the channel throughput that increases with the increase in packets length and a decrease in the expected lengths of the good and bad periods. Replacing the constant-throughput model with a more realistic one in our analysis is expected to improve the performance of the proposed

TABLE VII

COMPARISON OF VARIOUS RATE-CONTROL SCHEMES FOR SEVERAL VALUES OF ΔT (*mother& daughter* SEQUENCE IN QCIF FORMAT).

Scheme	$\Delta T = 333$ msec		$\Delta T = 500$ msec		$\Delta T = 666$ msec	
	PSNR	Starvation Prob.	PSNR	Starvation Prob.	PSNR	Starvation Prob.
Cycle-based	31.02	7.0×10^{-4}	31.06	6.6×10^{-4}	31.08	5.710×10^{-4}
SDP	30.82	0.1019	30.99	0.0633	31.02	0.0429
DDP	30.84	0.1008	31.12	0.0666	30.85	0.0395
TMN8	27.13	0.4216	29.23	0.1699	30.09	0.0644

algorithm with only a slight increase in the computational complexity.

APPENDIX

First, consider case 1:

$$\Phi = \int_0^\infty \left(\int_{Q_0+x}^\infty f_{\tilde{\Gamma}_b}(y)dy \right) f_{\tilde{\Gamma}_g}(x)dx \tag{29}$$

$$= \int_0^\infty \left(\int_{Q_0+x}^\infty \frac{\left(\frac{y}{\tilde{\beta}}\right)^{\delta-1}}{(\delta-1)!\tilde{\beta}} e^{-\frac{y}{\tilde{\beta}}} dy \right) \frac{(\frac{x}{\tilde{\alpha}})^{\gamma-1}}{(\gamma-1)!\tilde{\alpha}} e^{-\frac{x}{\tilde{\alpha}}} dx \tag{30}$$

$$= \int_0^\infty \left[e^{-\frac{(x+Q_0)}{\tilde{\beta}}} \sum_{i=0}^{\delta-1} \frac{(\frac{Q_0+x}{\tilde{\beta}})^i}{i!} \right] \frac{\left(\frac{x}{\tilde{\alpha}}\right)^{\gamma-1}}{(\gamma-1)!\tilde{\alpha}} e^{-\frac{x}{\tilde{\alpha}}} dx \tag{31}$$

$$= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \left[\sum_{i=0}^{\delta-1} \int_0^\infty \frac{e^{-\frac{x}{\tilde{\beta}}-\frac{x}{\tilde{\alpha}}} (\frac{x}{\tilde{\alpha}})^{\gamma-1} (\frac{Q_0+x}{\tilde{\beta}})^i}{i!(\gamma-1)!} dx \right] \tag{32}$$

$$= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1} (\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \times \int_0^\infty e^{-(\frac{1}{\tilde{\beta}}+\frac{1}{\tilde{\alpha}})x} x^{\gamma-1} (Q_0+x)^i dx \tag{33}$$

$$= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1} (\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \times \sum_{j=0}^{i} \binom{i}{j} Q_0^{i-j} \int_0^\infty e^{-\theta_1 x} x^{j+\gamma-1} dx \tag{34}$$

where $\theta_1 \stackrel{\text{def}}{=} \frac{1}{\tilde{\beta}} + \frac{1}{\tilde{\alpha}}$. Thus,

$$\Phi = \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \left[\sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1} (\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \left(\sum_{j=0}^{i} \binom{i}{j} Q_0^{i-j} \frac{(j+\gamma)!}{\theta_1^{\gamma+j}} \right) \right]. \tag{35}$$

For case 2, we have

$$\Phi = \int_0^{Q_0} \left(\int_{Q_0-x}^\infty f_{\tilde{\Gamma}_b}(y)dy \right) f_{\tilde{\Gamma}_g}(x)dx + \int_{Q_0}^\infty \left(\int_0^\infty f_{\tilde{\Gamma}_b}(y)dy \right) f_{\tilde{\Gamma}_g}(x)dx \tag{36}$$

$$\stackrel{\text{def}}{=} I_1 + I_2$$

where

$$I_1 = \int_0^{Q_0} \left(\int_{Q_0-x}^\infty \frac{\left(\frac{y}{\tilde{\beta}}\right)^{\delta-1}}{(\delta-1)!\tilde{\beta}} e^{-\frac{y}{\tilde{\beta}}} dy \right) \frac{\left(\frac{x}{\tilde{\alpha}}\right)^{\gamma-1}}{(\gamma-1)!\tilde{\alpha}} e^{-\frac{x}{\tilde{\alpha}}} dx \tag{37}$$

$$= \int_0^{Q_0} \left[e^{-\frac{(Q_0-x)}{\tilde{\beta}}} \sum_{i=0}^{\delta-1} \frac{(\frac{Q_0-x}{\tilde{\beta}})^i}{i!} \right] \frac{\left(\frac{x}{\tilde{\alpha}}\right)^{\gamma-1}}{(\gamma-1)!\tilde{\alpha}} e^{-\frac{x}{\tilde{\alpha}}} dx \tag{38}$$

$$= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \left[\sum_{i=0}^{\delta-1} \int_0^{Q_0} \frac{e^{\frac{x}{\tilde{\beta}}-\frac{x}{\tilde{\alpha}}} (\frac{x}{\tilde{\alpha}})^{\gamma-1} (\frac{Q_0-x}{\tilde{\beta}})^i}{i!(\gamma-1)!} dx \right] \tag{39}$$

$$= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1} (\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \times \int_0^{Q_0} e^{(\frac{1}{\tilde{\beta}}-\frac{1}{\tilde{\alpha}})x} x^{\gamma-1} (Q_0-x)^i dx \tag{40}$$

$$= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1} (\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \sum_{j=0}^{i} \binom{i}{j} (-1)^i (-Q_0)^{i-j} \times \int_0^{Q_0} e^{-\theta_2 x} x^{j+\gamma-1} dx \tag{41}$$

where $\theta_2 \stackrel{\text{def}}{=} \frac{1}{\tilde{\beta}} - \frac{1}{\tilde{\alpha}}$. Note that $\int_0^{Q_0} e^{-\theta_2 x} x^{j+\gamma-1} dx$ can be written as $\frac{(\gamma+j-1)!}{\theta_2^{\gamma+j}} \int_0^{Q_0} \frac{\theta_2 (\theta_2 x)^{\gamma+j-1}}{(\gamma+j-1)!} dx$ which is equal to $\frac{(\gamma+j-1)!}{\theta_2^{\gamma+j}} \left[1 - e^{-\theta_2 Q_0} \sum_{k=0}^{\gamma+j-1} \frac{(\theta_2 Q_0)^k}{k!} \right]$. Thus,

$$I_1 = \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1} (\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \times \sum_{j=0}^{i} \binom{i}{j} (-1)^i (-Q_0)^{i-j} \frac{(\gamma+j)!}{\theta_2^{\gamma+j}} \times \frac{(\gamma+j-1)!}{\theta_2^{\gamma+j}} \left(1 - e^{-\theta_2 Q_0} \sum_{k=0}^{\gamma+j-1} \frac{(\theta_2 Q_0)^k}{k!} \right) \tag{42}$$

For I_2, we have:

$$I_2 = \int_{Q_0}^\infty \left(\int_0^\infty f_{\tilde{\Gamma}_b}(y)dy \right) f_{\tilde{\Gamma}_g}(x)dx \tag{43}$$

$$= \int_{Q_0}^{\infty} \left(\int_0^{\infty} \frac{\left(\frac{y}{\tilde{\beta}}\right)^{\delta-1}}{(\delta-1)!\tilde{\beta}} e^{-\frac{y}{\tilde{\beta}}} dy \right) \frac{\left(\frac{x}{\tilde{\alpha}}\right)^{\gamma-1}}{(\gamma-1)!\tilde{\alpha}} e^{-\frac{x}{\tilde{\alpha}}} dx \quad (44)$$

$$= \int_{Q_0}^{\infty} \frac{\left(\frac{x}{\tilde{\alpha}}\right)^{\gamma-1}}{(\gamma-1)!\tilde{\alpha}} e^{-\frac{x}{\tilde{\alpha}}} dx \quad (45)$$

$$= e^{-\frac{Q_0}{\tilde{\alpha}}} \sum_{i=0}^{\gamma-1} \frac{(\frac{Q_0}{\tilde{\alpha}})^i}{i!}. \quad (46)$$

Hence, for case 2:

$$\begin{aligned} \Phi &= \frac{e^{-\frac{Q_0}{\tilde{\beta}}}}{\tilde{\alpha}} \sum_{i=0}^{\delta-1} \frac{(\frac{1}{\tilde{\alpha}})^{\gamma-1}(\frac{1}{\tilde{\beta}})^i}{i!(\gamma-1)!} \\ &\times \sum_{j=0}^{i} \binom{i}{j} (-1)^i (-Q_0)^{i-j} \frac{(\gamma+j)!}{\theta_2^{\gamma+j}} \\ &\times \frac{(\gamma+j-1)!}{\theta_2^{\gamma+j}} \left(1 - e^{-\theta_2 Q_0} \sum_{k=0}^{\gamma+j-1} \frac{(\theta_2 Q_0)^k}{k!}\right) \\ &+ e^{-\frac{Q_0}{\tilde{\alpha}}} \sum_{i=0}^{\gamma-1} \frac{(\frac{Q_0}{\tilde{\alpha}})^i}{i!}. \quad (47) \end{aligned}$$

REFERENCES

[1] C. Amaya and D. V. Rogers. Analysis of fade duration characteristics based on attenuation measurements in Canada and Southeast Asia. In *Proceedings of COST272 – European Union Forum for Cooperative Scientific Research*, May 2003.

[2] 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.

[3] 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.

[4] J. Cabrera, F. Jaureguizar, and S. Lei. Rate control in DCT video coding for low delay communications. *IEEE Trans. on Circuits and Systems for Video Technology*, Feb. 1999.

[5] 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*, 12(6):496–510, June 2002.

[6] H. Chaskar, T. V. Lakshman, and U. Madhow. TCP over wireless with link level error control: Analysis and design methodology. *IEEE/ACM Transactions on Networking*, 7:605–615, Oct. 1999.

[7] 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.

[8] G. Chen, X.-H. Yu, and J. Wang. Adaptive channel estimation and dedicated pilot power adjustment based on the fading-rate measurement for a pilot-aided cdma system. *IEEE Transactions on Selected Areas in Communications*, 19(1):132–140, Jan. 2001.

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

[10] J. Goldhirsh and W. J. Vogel. Handbook of propagation effects for vehicular
and personal mobile satellite systems: Overview of experimental and modeling results. http://www.utexas.edu/research/mopro/index.html.

[11] Y. Hase, W. J. Vogel, and J. Goldhirsh. Fade-durations derived from land-mobile-satellite measurements in Australia. *IEEE Trans. on Communications*, 39(5):664–668, May 1991.

[12] M. Hassan, M. Krunz, and I. Matta. Markov-based channel characterization for tractable performance analysis in wireless packet networks. *IEEE Transactions on Wireless Communications*, 3(3):821–831, May 2004.

[13] 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.

[14] 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.

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

[16] T. Krauss and M. D. Zoltowski. Bilinear approach to multiuser second-order statistics-based blind channel estimation. *IEEE Transactions on Signal Processing*, 48(9):2473–2486, Sept. 2000.

[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] W. Lee and Y. Yeh. On the estimation of the second-order statistics of log normal fading in mobile radio environment. *IEEE Transactions on Communications*, 22(6):869–873, June 1974.

[19] 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.

[20] 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.

[21] 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.

[22] J. Ronda, M. Eckert, F. Eckert, and N. Garcia. Rate control and bit allocation for mpeg-4. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:1243–1258, Dec. 1999.

[23] 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.

[24] H. S. Wang and N. Moayeri. Finite-state Markov channel – A useful model for radio communications channels. *IEEE Trans. Veh. Tecnol.*, 44:163–171, Feb. 1995.

[25] 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.

[26] D. Wu, Y. T. Hou, and Y.-Q. Zhang. Scalable video coding and transport over broadband wireless networks. *Proceedings of the IEEE – Special Issue on Multi-Dimensional Broadband Wireless Technologies and Applications*, 89(1):6–20, Jan. 2001.

[27] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha. Streaming video over the Internet: Approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:282–300, March 2001.

[28] Q. Zhang, W. Zhu, and Y.-Q. Zhang. End-to-end QoS for video delivery over wireless Internet. *Proceedings of the IEEE*, 93(1):123–134, Jan. 2005.