Adaptive Rate Control Scheme for Video Streaming Over Wireless Channels*

Marwan M. Krunz and Mohamed Hassan
Department of Electrical & Computer Engineering
University of Arizona, Tucson, AZ 85721
Tel.: (520) 621 8731
{krunz,mhassan}@ece.arizona.edu

Abstract

Providing continuous video playback with graceful quality degradation over wireless channels is fraught with challenges. Video applications require stringent delay guarantees and a relatively high throughput. Wireless channels are error prone, time varying, and bandwidth limited. To improve the reliability of the wireless link, forward error correction (FEC) and automatic repeat request (ARQ) are often used. If designed for the worst channel conditions, FEC can provide constant throughput and bounded delay. However, this causes unnecessary overhead and reduces the maximum achievable throughput when the channel is in good conditions. On the other hand, it is difficult to achieve strict delay guarantees using ARQ schemes alone, especially when the channel is in deep fading. Playback buffer occupancy plays a major role in the target video quality. The retransmission of erroneous packets and the reduction in throughput due to FEC overhead can lead to playback buffer starvation as well as transmitter buffer fullness. Therefore, it is desirable to reduce the bit rate of the transmitted video signal and increase error protection when the channel is anticipated to be bad or the receiver playback buffer starvation is predicted. In this study, we introduce a scalable and adaptive source-channel rate control scheme for video transmission over wireless packet networks. In this scheme, the level of adaptiveness is optimized to reduce the bandwidth requirement while guaranteeing delay and loss bounds. Simulation and numerical investigations are carried out to study the interactions among various key parameters and verify the adequacy of the analysis.

1 Introduction

The recent remarkable growth in wireless access speeds combined with the advent of scalable compression schemes (e.g., MPEG-4 [1]) have paved the way towards enabling video streaming services over wireless media. Historically, the cost of such services has been too prohibitive for commercialization, mainly due to the high bandwidth demand of digitized video and its stringent transport-delay requirements. High-speed wireless LANs (e.g., the IEEE 802.11a standard [2]) can now offer tens of Mbps of bandwidth, triggering significant business interest in multimedia-capable wireless services and devices.

*This work was supported by NSF under grants ANI 0095626, ANI-0313234, and ANI-0325979 and 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.

Yet, the road to providing continuous video playback over wireless channels is still fraught with challenges [3]. Wireless channels are highly dynamic, with a bit error rate (BER) that fluctuates by orders of magnitude in less than a second. Timely delivery of video frames is being hampered by the contention-based nature of common wireless LANs, which gives rise to multi-access interference and packet collisions. User mobility further complicates the situation, causing the channel state to vary in time and necessitating occasional handover between access points (or base stations). In variable bit rate (VBR) video compression, the perceptual quality is maintained at the expense of adjusting the number of bits per encoded frame. The resulting frame size varies depending on the scene dynamics and the types of compression involved (e.g., intra-coding, motion prediction, etc.). So when the video stream is generated and transported at a constant *frame rate* (e.g., 30 frames/sec), it displays a VBR traffic pattern that is difficult to efficiently transport over any packet network, not to mention a wireless network. To make matters worse, in MPEG schemes frames exhibit certain inter-dependencies, whereby correct decoding of a given frame requires correct decoding of a previous (and sometimes future) "reference" frame. Hence, timely delivery of reference frames must be guaranteed with a higher probability than for other frames.

To address the above challenges, a number of possible techniques can be used, separately or in combination. One approach is to exploit the scalable format of MPEG video for rate control on a frame-by-frame basis. For example, in MPEG-4 a gradual reduction in the size of an already encoded frame produces a graceful degradation in video quality. Another direction is to improve link reliability using channel coding (i.e., forward error correction), automatic repeat request (ARQ), or both. If designed according to the worst-channel conditions, FEC can by itself ensure sustained throughput and bounded delay. But fixing the code rate leads to high inefficiencies when the channel is dynamic. Adaptive FEC is more appropriate in this case. However, deciding on the appropriate code rate is nontrivial, since the channel state has to be conveyed back to the transmitter, which uses it to decide on the new code rate (by that time, the channel state may have changed). On the other hand, it is difficult to achieve strict delay guarantees using ARQ alone, especially when the channel is in deep fade. Hybrid ARQ schemes (e.g., [4, 5, 6, 7]) provide the best features of ARQ and FEC, and will therefore be used in our work. Joint source/channel coding (e.g., [8, 9, 10]) can also be used to "optimally" allocate the available channel capacity between the encoded bitstream and the channel code. This can be done as a part of the encoding process or as a post-encoding step. The latter approach, which is adopted in this paper, amounts to joint rate control/channel coding. The low processing of this approach makes it particularly attractive for real-time delivery of scalable video over wireless channels.

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, with little or no accommodation of the networking requirements. For instance, many of these studies are primarily aimed at optimizing the effective throughput of the channel, without considering the impact of source and channel coding on the transport delay. The delay performance of hybrid ARQ schemes (both type I and II) has been studied [4, 6]), but independent of the video content (i.e., without regard to source coding). Most studies on joint source/channel coding address the problem from an information theoretic point of view, and do not account for network performance and protocol issues, including packetization and retransmissions. Such studies often overlook the need to prevent buffer starvation and overflow at the decoder, both of which are critical to maintaining continuous video playback. In general, we believe that the literature on video streaming over wireless channels still lacks a comprehensive treatment of the topic, 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.

In this paper, we take a first step towards providing an integrated approach for streaming video traffic over wireless links. More specifically, we devise an adaptive source/channel rate control scheme that aims at optimizing the bandwidth requirement over the wireless channel while providing soft guarantees on frame delay. As shown in Figure 1, we consider a system in which the video server delivers archived or "real-time" (encoded-on-the-fly) video to a mobile client through a base station (BS) or an access point (AP). The video is encoded using a scalable VBR compressor such as MPEG-4, and can be rate controlled without any transcoding. The mobile receiver acts as a *bandwidth manager*, and continuously monitors the channel state, the playback buffer occupancy, and the quality of the played back video. Note that the receiver can use the history of the received frames to predict the size of the next video frame (due to space limitation, the prediction process is beyond the scope of this study). Based on this 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, ensuring that the feedback control channel is almost error-free.

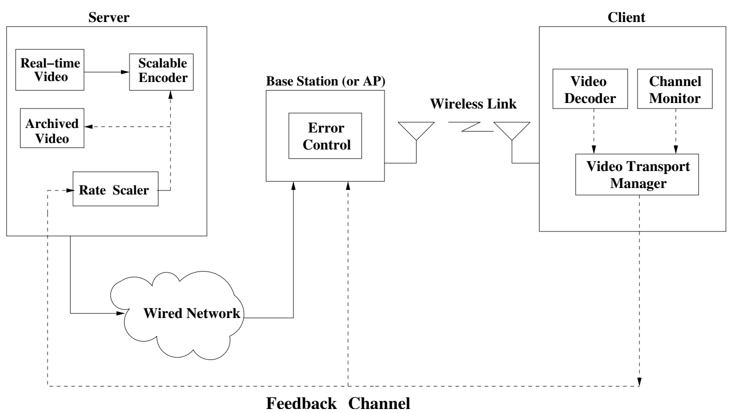


Figure 1: Overall architecture of the video streaming system.

The rest of the paper is organized as follows. In section 2 we present the proposed adaptive source/channel rate control scheme. Performance evaluation of this scheme is given in section 3. Finally, section 4 summarizes the results of this study and outlines our future work.

2 Joint Source-Channel Rate Control Scheme

2.1 Stabilizing the playback Buffer

When a video frame is to be transmitted over the wireless link, 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 code 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 $S(m)$ be the size of the mth video frame. This frame is segmented into $N_p(m) = \lceil S(m)/K_{in} \rceil$ LL packets, where K_{in} is the number of *information bits* in each LL packet. When the context is clear, we drop off the dependence on m from the notation. Besides the information bits, each LL packet contains K_{par} *parity bits*, for a total of $K_{tot} = K_{in} + K_{par}$ bits. For now, we assume that K_{in} and K_{par} are fixed for all LL packets comprising a frame, but can vary from one frame to another. Determining the appropriate K_{par} and K_{in} values for a given frame is done by the receiver, as explained in Section 2.2. In general, such determination requires the receiver to use a predicted value of S, denoted by $\hat{S}$. However, if the "next" frame is already available at the transmitter by the time the current frame completes its transmission (as in the case of archived video or when the transmitter buffer is backlogged), then there is no need to forecast the frame size. Instead, the size of the next frame can be piggybacked onto the current frame.

Consider the situation at the playback buffer, which contains correctly received video frames. The video session starts with a preloading phase in which $Q^* + 1$ frames are prefetched into the buffer before playback commences. The preloading phase provides a cushion against variations in the frame arrival rate at the playback buffer, allowing packet retransmissions (and, optionally, interleaving) to be used without starving the buffer. The value of Q^*, which is referred to as the *playback buffer threshold*, is selected depending on the average channel BER, the channel coherence time, and the target video quality. One reasonable choice is to set Q^* to a value that is slightly larger than the number of frames generated within an average fade duration. 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 $Q(i)$ be the number of frames in the playback buffer *right after* the playback of the ith video frame, $i = 1, 2, \ldots$. Note that $Q(1) = Q^*$. The occupancy of the playback buffer evolves according to:

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

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

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 the threshold Q^*. The value of

T_c is used in the subsequent determination of the source-rate and channel-code parameters (as explained in the next section).

Depending on $Q(i)$, the receiver selects the value of T_c for the next frame as follows:

- **Case I:** $Q(i) \geq Q^*$ **(Stable Regime)**: In this case, T_c is set to $1/f_p$.
- **Case II:** $0 \leq Q(i) < Q^*$ **(Underflow Regime)**: In this case, T_c is set to $1/(f_p(Q^* - Q(i)))$.

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. Note that in this case, $f_r(i) = f_p$ in (1), and the buffer reaches its steady-state $Q(i+1) = Q(i) = Q^*$, as desired. In contrast, when the channel is bad, $f_r(i)$ becomes smaller than f_p (at least, temporarily), so the queue length starts to decrease away from Q^*. To compensate for this, the subsequent $Q^* - Q(i)$ frames need to arrive faster than usual, at an average rate of $f_p(Q^* - Q(i))$ frames per second. If that happens, the queue length will build up to Q^* in one frame period. Of course, the channel state could change in the mean time, so it is safer to decide on T_c for a single observation period only.

While the above scheme tries 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, whereby the most recently played frame is played back again (which amounts to briefly pausing the video). Incorporating more sophisticated concealment approaches will be addressed in a future work.

2.2 Adaptive Computation of Source Rate and Channel-Code Parameters

Once T_c has been updated, the receiver uses it along with the size of the next frame (predicted or actual) and the current channel state to determine the "optimal" channel-code parameters (denoted by K^*_{tot} and K^*_{par}) 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 its LL packets) when the channel is in state i. We assume that the wireless channel fluctuates according to a 2-state continuous-time Markov chain, where state 0 is the good state and state 1 is the bad state. For $i = 0, 1$, let p_i be the BER during state i ($p_0 \ll p_1$). The sojourn times for the "good" and "bad" states are exponentially distributed with means α_0^{-1} and α_1^{-1}, respectively. 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 scaled down. So the receiver reduces the value of S (or $\hat{S}$, if the frame size is predicted), and repeats the computation for the optimal channel-coding parameters. The value of ϵ can be selected depending on the relative importance of the transmitted frame, the current channel state, and the number of frames in the playback buffer. The process of scaling down the frame size and computing the optimal channel-code parameters continues until an appropriate frame size is found for which the optimal channel-code parameters are sufficient to ensure $F_{tot}(T_c, i) \geq \epsilon$. At this point, the scaled frame size and optimal channel-code parameters are fed back to the video server and the

BS, respectively. Let ξ be the scaling factor, $0 < \xi \leq 1$. The video server uses the fed back information to scale down the size of the ensuing frame to $\min\{S, \xi\hat{S}\}$.

The optimization procedure is now explained in more detail. First, we compute $F_{tot}(x, i)$. Then, we search for that optimal pair $(K^*_{tot}\ , K^*_{par})$ that results in $F_{tot}(x, i) \geq \epsilon$ for the predicted frame size. If such a pair does not exist, we rely on source control by gradually decreasing the frame size until we reach a triple $(S^*, K^*_{tot}\ , K^*_{par})$ for which $F_{tot}(x, i) \geq \epsilon$.

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

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

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

For simplicity, we assume that the channel state does not change during the transmission of a video frame. Conditioned on channel state i, the number of retransmissions that a given LL packet undergoes (including the first transmission attempt) is a geometric rv 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 $\hat{S}$:

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

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 2 depicts the effects of K_{tot} and K_{par} on $F_{tot}(T_c, i)$ for RS code. The LHS of the figure depicts the delay performance as a function of E_{max} when $K_{tot} = 750$ bits and the BER is 0.15 (bad channel state). As E_{max} is gradually increased, $F_{tot}(T_c, i)$ increases (performance becomes better) up to some optimal point, E^*_{max}, beyond which the overhead of FEC starts to overshadow its benefit. Note that increasing E_{max} 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 behavior in this figure. The staircase behavior for large values of E_{max} is attributed to the truncation effect of the ceil function in (3). A somewhat similar trend is observed on the RHS of Figure 2, where an increase in K_{tot} improves the delay performance up to some point, after which the trend is reversed as a result of the limited correction capability for a fixed E_{max}. However, in this case, instead of a single optimal K_{tot} value, there is a set of optimal (or near-optimal) values.

When the optimal $(K^*_{tot}\ , K^*_{par})$ pair results in $F_{tot}(T_c, i)$ that is still less than ϵ, we resort to source control. A scaled frame size can be obtained subject to a minimized distortion level [11, 12]. Alternatively, based on the scaled frame size and its sensitivity (i.e., its type and effect on quality), the type of scaling (signal-to-noise ratio, spatial, or temporal

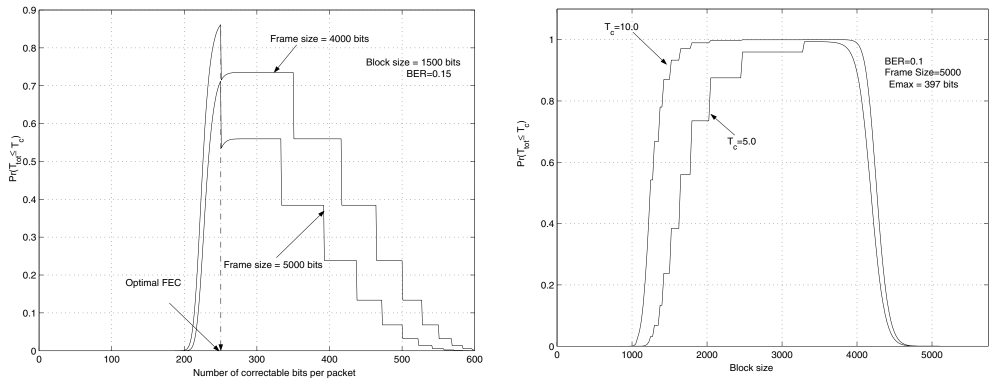


Figure 2: Impact of channel-code parameters on frame delay.

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.

3 Simulation Results

In this section, we describe the simulation setup and give results for the performance of the proposed scheme. Table 1 summarizes the default values used in the simulations, unless otherwise specified. Recall that ϵ is used as the minimum probabilistic bound when searching for the optimal pair (K^*_{tot}, K^*_{par}) and the scaled frame size using Equation 4. The traces used in the simulations were obtained from [13].

Parameter	Value
f_p	24 frames/sec
p_0	10^{-5}
p_1	10^{-2}
α_0^{-1}	0.1 sec
α_1^{-1}	0.0333 sec
ϵ	0.9
RTT	0.04 sec
Access bandwidth	1 Mbps

Table 1: Values of parameters used in the simulations.

Figure 3 depicts the relative percentage of played back frames. The LHS of this figure is intuitive and shows that with the proposed adaptive scheme and a reasonable preloading phase, 100% of the frames are played back. 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 FEC, we allow E_{max} to vary with the channel state, while for nonadaptive FEC, we fix E_{max} at a conservative value of $p_1 \times K_{tot}$. It is obvious that the proposed scheme outperforms the two cases without scaling. The RHS of this figure shows the relative percentage of played frames versus Q^*. It can be seen that the relative percentage of played back frames for $E_{max} = 50$ is less than the percentage when $E_{max} = 20$. Therefore, to achieve higher playback rates, the number of correctable bits cannot be blindly chosen or increased.

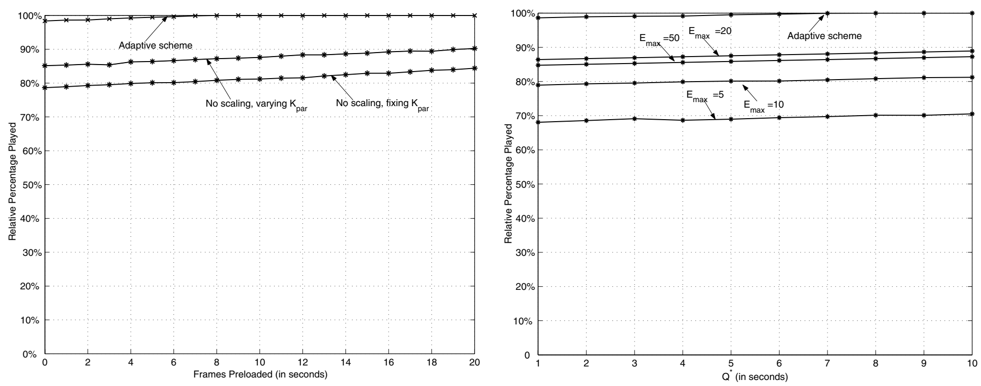
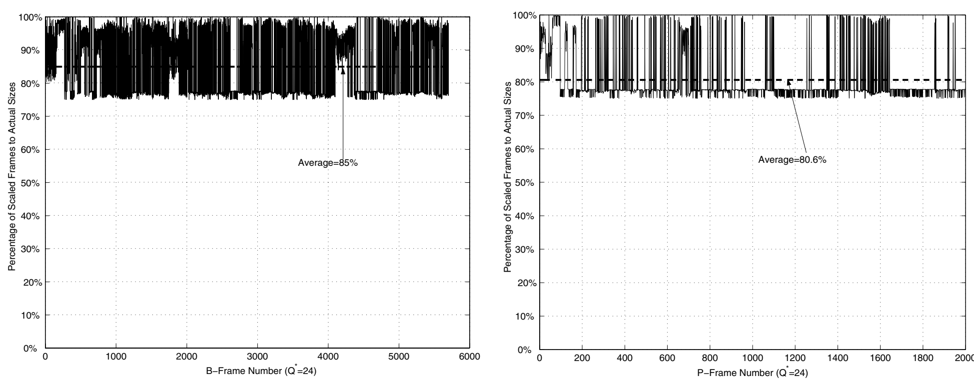


Figure 3: Relative percentage of played back frames.



(a) (b)

Figure 4: Normalized sizes of scaled B & P frames ($Q^* = 24$).

Figures 4 and 5 show the relative percentage of the scaled frames to the actual frame sizes for B, P, and I frames, respectively. Although B frames are less important than P and I frames, it is obvious that they undergo less scaling due the fact that they are relatively smaller compared to P and I frames. Figure 6 shows the playback buffer evolution with time for three different values of Q^*. We noticed that the average normalized sizes of scaled frames (normalized to the actual frame sizes) are very close for different values of Q^*. But the absolute amounts scaled from the different frames increase with the value of Q^*. This results from how we choose the value of T_c to recover from underflow or starvation situations. Note that the values of Q^* used to obtain this graph are arbitrarily selected to study the effect of Q^*. Also, it is worth noting that the smaller the value of Q^*, the higher the capability of the proposed scheme to maintain a desired Q^*.

4 Conclusions

In this paper, we proposed a scalable and adaptive source-channel rate control scheme for video transmission over wireless packet networks. An analytical model was used to maximize the probability of sending a frame in what we called the critical time T_c. Our

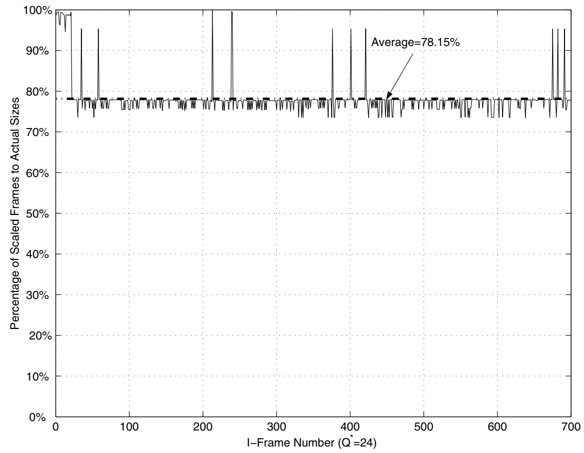


Figure 5: Normalized sizes of scaled I frames ($Q^* = 24$).

analysis exploited the advantages of the ARQ and FEC schemes as well as those of the scalable compression schemes. We provided a probabilistic expression that contains the key parameters of the proposed adaptive model. We showed that for each transmission-candidate frame if there is no optimal or no near optimal pair (K^*_{tot} , K^*_{par}) that maximizes the obtained probabilistic expression, the required probability bound can be achieved by source control. Simulation results showed that the blind choice of the number of correctable bits or packet sizes has a counter effect on the relative playedback percentage. In a future work, we will study the effect of channel variations during frame transmission time, We will also consider scaling subject to R-D curves with more sophisticated concealment approaches.

References

[1] MPEG-4 Video Group, "Overview of the MPEG-4 Standard,"(http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm).

[2] ISO/IEC 8802-11:1999/Amd 1:2000(E), IEEE Standards for Local and Metropolitan Area Networks: Interoperable LAN/MAN Security (SILS).

[3] P. Bahl, "Supporting digital video in a managed wireless network," *IEEE Communications Magazine*, vol. 36, pp. 94–102, June 1998.

[4] S. Kallel, "Analysis of a type II hybrid ARQ scheme with code combining," *IEEE Trans. Commun.*, vol. 38, pp. 1133–1137, 1990.

[5] S. Kallel, "Analysis of memory and incremental redundacny ARQ schemes over a nonstationary channel," *IEEE Trans. Commun.*, vol. 40, pp. 1474–1480, 1992.

[6] R.H. Deng, "Hybrid ARQ schemes employing coded modulation and sequence combining," *IEEE Trans. on Communications*, vol. 42, pp. 2239–2245, Feb-Apr. 1994.

[7] Q. Zhang, and S.A. Kassam, "Hybrid ARQ with selective combining for fading channels," *IEEE J. Select. Areas Commun.*, vol. 17, Issue: 5 , pp. 867-880, May 1999.

[8] P. Frossard and O. Verscheure, "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," *IEEE Trans. on Image Processing*, vol. 10, Issue 12, pp. 1815–1825, Dec. 2001.

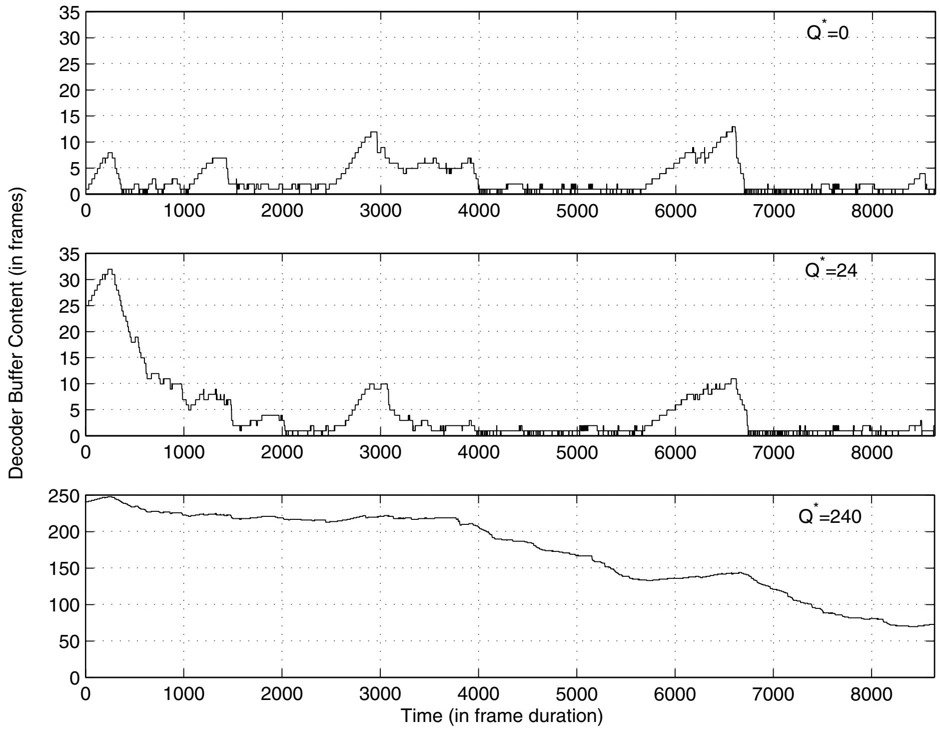


Figure 6: Time evolution of the playback buffer.

[9] T. Fingscheidt, T. Hindelang, R.V. Cox, and N. Seshadri, “Joint source-channel (De)coding for mobile communications,” *IEEE Trans. on Commun.*, vol. 50, pp. 200–212, Feb. 2002.

[10] Y. Su, J. Lu, J. Wang, K.B. Letaief, and J. Gu, “Adaptive segmentation based joint source-channel coding for wireless video transmission,” *IEEE 53rd Vehicular Technology Conference*, vol. 3, pp. 2076–2080, 2001.

[11] A. Ortego and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE S. Processing Magazine*, Volume: 15 Issue: 6, 1998.

[12] M. Gallant and F. Kossentini, “Rate-distortion optimized layered coding with unequal error protection for robust Internet video,” *IEEE Trans. on Circ.and Sys. for Video Tech.*, vol. 11, pp. 357–372, March. 2001.

[13] 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).