In Proc. of the 2nd. IEEE Intl. Conf. on Image Proc. ICIP'95, Washington, DC, Oct. 1995

RATE CONTROL FOR VIDEO CODING OVER VARIABLE BIT RATE CHANNELS WITH APPLICATIONS TO WIRELESS TRANSMISSION

Antonio Ortega

Signal and Image Processing Institute
Dept. of EE-Systems, USC,
Los Angeles, CA 90089
ortega@sipi.usc.edu

Masoud Khansari

Dept. of EECS
University of California,
Berkeley, CA 94720
masoud@diva.eecs.berkeley.edu

ABSTRACT

Video transmission over wireless links is an emerging application which involves a time-varying channel. In this paper we propose that rate control algorithms should be used at the video encoders, along with models of the channel behavior, to improve the performance of such systems. Rather than letting information be lost as the channel conditions change, in our scheme channel state information is fedback to the encoder. We propose a method, based on dynamic programming, to compute the rate-distortion performance for a given channel and source realization. We show how the rate-distortion performance changes with end-to-end delay and feedback delay.

1. INTRODUCTION

The most common scenario for digital video transmission has been that of constant bit rate channels (CBR). This situation arises in point to point transmission over wired channels as well as in reading video data from storage devices such as computer disks and CD-ROMs. With CBR transmission, the system has to operate under constant end-to-end delay in order to ensure that correct decoding can be achieved in real time. Due to the variable rate nature of typical video encoder outputs, buffering is required to absorb the rate variations and enforce the end-to-end delay constraint [1]. Under these conditions, a rate control algorithm is used to modify the encoder mode of operation so as to avoid buffer overflow.

The rate control (or buffer control) problem has been studied in the literature [2, 3]. In [4] the problem was formulated as a deterministic bit allocation with constraints and solved using dynamic programming. A deterministic formulation can be achieved by increasing the encoding delay so as to allow the encoder to base its rate control strategy on several "future" frames or blocks, rather than only on the current frame as most algorithms do. In particular, for encoding delay sufficiently long, one can determine an upper bound on the performance of the video encoder under buffer constraint.

While CBR transmission seems to be the dominant application nowadays, recently some emerging variable bit rate (VBR) transmission environments are receiving a great deal of attention. This motivates us to study the problem of buffer control under the time-varying conditions. While the application we describe here is transmission over wireless channels (where the channel rate diminishes during fade periods) our ideas would also be applicable, with the proper models, to VBR transmission over Asynchronous Transfer Mode (ATM) networks. It has to be noted that in a wireless scenario, due to the fading conditions, a variable rate operation is all but unavoidable. Similarly, in the ATM case, the variable channel rate could result from network congestion conditions.

2. PROBLEM DEFINITION

We can formulate the problem as follows. A video sequence consisting of N frames is transmitted over a time-varying channel. Since both encoder and decoder are attached to synchronous devices, the system operates with a constant end-to-end delay, ΔT. Thus frame i, encoded and placed in the encoder output buffer at time T_i, will have to be available at the decoder at time $T_i + \Delta T$. We can consider as our basic time units either the picture frame intervals, i.e. the time it takes to process a frame (e.g. 1/30 secs.), or the data-frame intervals, i.e. the time it takes to transmit a data "cell", including the header information, through the channel (5 ms in the system considered here). For simplicity, we start by assuming that the encoder receives instantaneous feedback on the *effective* channel rate for the previous frame interval but we also consider the case where feedback involves a delay. The channel rate observed during interval i is R_i, i.e. R_i is the actual number of bits transmitted during a data-frame interval. Call r_i, d_i the rate and distortion for block i. Denote B_i the buffer occupancy after the i-th block has been put in the buffer. In the CBR case the buffer state can be computed as

$$B_i = \max(B_{i-1} + r_i - R_i, 0), \tag{1}$$

and, in order to guarantee that the information will arrive at the decoder on time to be decoded, it is sufficient to choose a buffer size $B_{max} = R \cdot \Delta N$, where R is the channel rate in bits per frame interval and ΔN is the number of frame intervals corresponding to a ΔT delay [1]. To ensure proper decoding we just need to prevent buffer overflow. In a VBR channel situation this no longer holds. However, if the channel rate at every instant, R_i, is known beforehand then we can define the effective buffer size at time i as

$$B_{max}(i) = \sum_{j=i}^{i+\Delta N-1} R_i. \qquad (2)$$

If $B_{max}(i)$ is never exceeded then the delay constraint will be met. Note that if the maximum channel rate is R_{max}, the physical buffer size required at both encoder and decoder would be $R_{max} \cdot \Delta N$. Obviously the problem with this formulation is that it entails knowledge of *future* channel rates R_i through $R_{i+\Delta N-1}$. Thus in practice it cannot be used as such. If we assume for a moment that this channel rate information is available, our formulation for optimizing the transmission becomes identical to the one in [4] *except for the fact that both the buffer constraint and the channel rate are time varying.* Thus a trellis based optimization such as the Viterbi Algorithm can be used to find the optimal solution (see [4] for details), i.e. the choice of operating point for each frame, r_i, d_i such that the cost function $\sum_{i=1}^{N} d_i$ is minimized *and* the buffer occupancy never exceeds $B_{max}(i)$.

The deterministic methods of [4] allow us to trade-off encoding delay for performance and provide bounds on the performance of any buffer control scheme operating with the given delay. Our goal in this work is to obtain bounds in the variable-rate channel case as well. We can define two cases:

Case I (Known channel): Here we assume that the channel rates (even the future ones!) are known. Thus we can obtain the best solution and performance for a given encoding delay. This will serve as a bound on the achievable performance of any other scheme.

Case II (Channel model): In a real life scenario the coder would not have access to the information assumed in Case I. Instead, it is likely that the encoder will use a model of the channel behavior (the model could be either fixed through training or learnt "on the fly"). The model would allow the encoder to estimate expected values of rate for the following frame intervals.

3. SYSTEM DESCRIPTION

3.1. Video coder

As our video coder we select for simplicity a conditional replenishment coder based on JPEG. The decision on whether to replenish a block is made based on a fixed threshold on the original images so that we introduce no coding dependencies. Each of the replenished blocks can be encoded using one out of three possible JPEG-like quantization settings. Combined with the information needed to encode the replenishment decision we need a total of 2bits of overhead per block.

In our simulations we encounter situations where data-frames are altogether lost (this occurs when there is overflow of the effective buffer.) We assume that the corresponding blocks will be replaced by a uniform block with value 128 and we compute the resulting distortion accordingly. Obviously this is a pessimistic assumption since lost blocks will be more likely replaced using error concealment methods. Especially for low activity sequences such as the "Miss America" used in our experiments, this latter approach should be superior.

3.2. Uplink and downlink transceivers

The system under consideration is patterned after $IS-95$ standard proposed for digital wireless communication. This system is a wide-band CDMA spread spectrum system with processing gain of 128 and chip rate of 9.83 Mchips/s, resulting in a data rate 76.8 Kbits/s. The data-frames are of 5 msec or 48 bytes duration of which 41 bytes is the payload. As a result the information transmission rate is 65.6 Kbits/s.

In general, a wireless link consists of two radio links, namely uplink or reverse link (mobile-to-base) and downlink or forward link (base-to-mobile). Because of the asymmetry in the information available at the base and the mobile and also the different power constraints and hence processing power, the design of these two links have different requirements.

On the uplink transmitter, the user data is first spread using 1/3 convolutional encoder. The encoded bits are then further spread using 64-ary orthogonal signalling followed by symbol interleaving and QPSK PN spreading. The receiver employs both antenna and multipath diversity where a number of correlators (each correlator corresponds to a path) are assigned to each antenna. A fast closed loop power control is used to combat Rayleigh fading.

On the downlink transmitter, user information is encoded using half-rate convolutional encoder and the encoded bits are QPSK spread and transmitted using BPSK modulation. Artificial multi-path thought use of different antennas (3 in our case) is used to combat fading. Also, the signal transmission is pilot assisted. Using the pilot signal received at the receiver, maximal ratio combining of different paths is achieved. The combiner is followed by de-interleave and soft decision Viterbi decoder.

Because of the employment of fast power control at the uplink channel, the bit error bursts tend to be of shorter duration. Therefore, for a given BER, there are more number of data-frames in error for the uplink channel – there are, however, less number of bits in error in each erroneously received data-frame. The downlink channel has smaller number of frames in error but due to the bursty nature of the errors, the data-frame error bursts tend to be more persistent and hence of longer duration.

3.3. Channel Model

Using simulation results of both the uplink and the downlink transceivers, a Markov chain model was used to model the behavior of the links. We use a similar model as in [5] where it is used to evaluate the performance of different speech coders. The model has N states where the channel is in state $i = 0, \cdots, N-1$ if the last i frames are received erroneously. Therefore, for all $1 \leq i \leq N-2$, the next transition is either to the next higher state (state $i+1$) or back to state 0 based on the status of the currently received data-frame, see Fig. 1. If the channel is in state $N-1$, it will always return to state 0. Note that, using this model, it is only possible to generate burst errors of at most of length $N-1$. The error statistics can then be controlled by properly assigning values to the probabilities $p_0, p_1, \cdots, p_{N-2}$. Table 1 shows these values for the uplink and downlink channels for BER $= 10^{-3}$, where N was found to be 15 and 7 for the downlink and the uplink channel, respectively. These values are found by matching the parameters of the Markov chains to simulations of the transceivers.

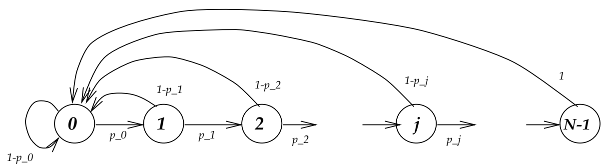

Figure 1: Markov model for the channel behavior

	Downlink	Uplink
p_0	0.001469	0.064292
p_1	0.516068	0.100324
p_2	0.778388	0.164083
p_3	0.854118	0.149606
p_4	0.936639	0.526316
p_5	0.873529	1.000000
p_6	0.905724	
p_7	0.881041	
p_8	0.831224	
p_9	0.893401	
p_{10}	0.863636	
p_{11}	0.717105	
p_{12}	0.853211	
p_{13}	0.763441	
p_{14}	1.000000	

Table 1: Markov Model transitional probability parameters for the downlink and the uplink channels

4. EXPERIMENTAL RESULTS

In our experiments we use the video compression and channel schemes described in the previous section. We use data-frames as our basic time units, and consider that they can either be (i) transmitted successfully, in which case we have a rate of 328 bits/data-frame (effective rate, i.e. not counting headers and overhead, except for the source coding overhead), or (ii) in error, in which case 0 effective bits are transmitted.

To justify this "all or nothing" model for the rate, consider how, as discussed in [6], when it comes to combat fading, automatic repeat request (ARQ) techniques should be chosen over forward error correction (FEC). Given that channel errors are not uniformly distributed, a FEC technique providing reliability in fade situations would require a prohibitive cost. A variable rate technique such as ARQ is better in that the rate at which information is reliably transmitted varies so that the overhead is only high during fade periods. A typical ARQ system will maintain a buffer including information waiting to be transmitted and information that has been transmitted but not acknowledged. In this way, when the channel conditions deteriorate the number of bits transmitted is kept constant but, due to the retransmissions, the effective bit rate is lower. We can see thus that an ARQ system has already built-in a way to measure the effective channel rate, namely, measuring the length of the ARQ queue [6].

4.1. Case I: Known channel

Figs. 2 and 3 show examples of optimization for two channel realizations for, respectively, uplink and downlink at BER 10^{-3}. The top curve in each figure represents the effective buffer size to be used in the optimization. The bottom curve is the resulting optimal buffer occupancy in *Case I*. Note how the uplink channel is characterized by relatively short but frequent fades while the downlink shows relatively few but long fades.

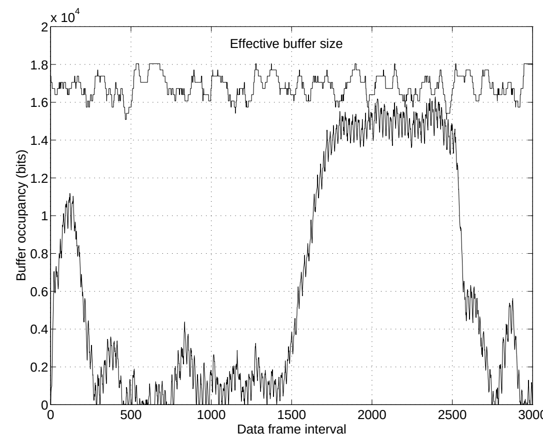

Figure 2: Example of solutions (buffer occupancy and effective buffer size for uplink channel at $BER = 10^{-3}$

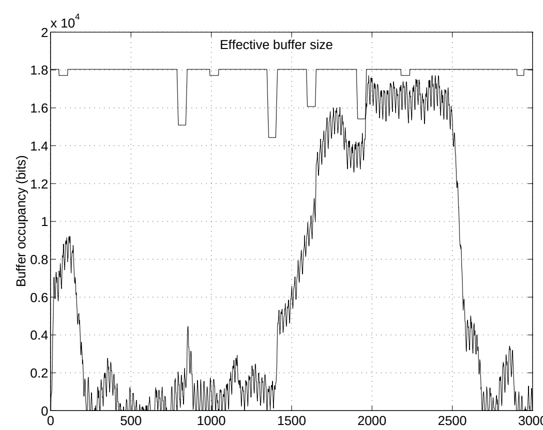

Figure 3: Example of solutions (buffer occupancy and effective buffer size) for downlink channel at $BER = 10^{-3}$

Figs. 4 and 5 depict the rate-distortion performance for various values of the end-to-end delay ΔT. Note that, as expected, the performance improves as end-to-end delay increases. These specific curves are particular to the channel realization we consider. By generating a sufficiently large number of channel realizations and averaging the performance we could obtain upper bounds on the average performance of the channel. This result can then be applied, for example, to choose the end-to-end delay for a given type of channel and source.

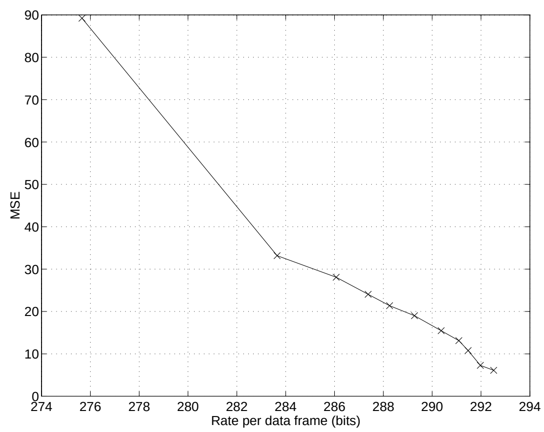

Figure 4: Upper bound on R-D performance for a sample Uplink channel at $BER = 10^{-3}$ for a range of end-to-end delays from 25ms (left) to 275ms (right). Note that increases in delay result in decreases in MSE and increases in rate.

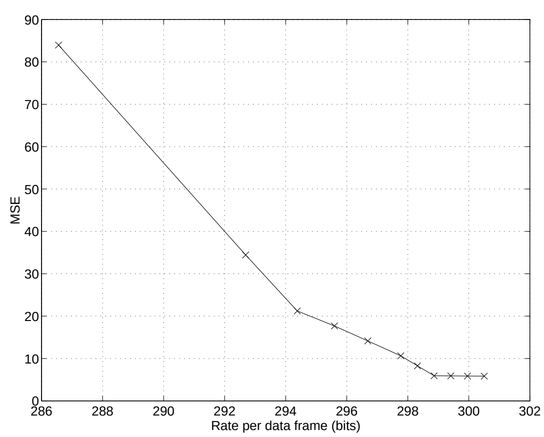

Figure 5: Upper bound on R-D performance for a sample Downlink channel at $BER = 10^{-3}$ for a range of end-to-end delays from 25ms (left) to 275ms (right)

4.2. Case II: Channel model

Finally, as mentioned earlier, the above results provide performance bounds but are not implementable since they require non-causal knowledge of the channel state. To obtain a more realistic bound we assume that the encoder has access to an accurate probabilistic channel model. We then use the same optimization technique except that the effective buffer size is estimated based on the most recent observed channel (which need not be the most recent one as there might be a lag in the observation feedback) and the corresponding channel model. In this case we still use a decision-delayed optimization for the video data (i.e. we assume the whole sequence is known) but we force the algorithm to operate with the information that will actually be available to the coder in a real time scenario. Thus at time i the coder can only estimate the expected effective buffer size, since the future channel rates are not known. We use the above described Markov-models to produce the estimated effective buffer size. In this case the channel information is not exact and there will thus be instances where the effective buffer size constraint will be violated and thus cells will be lost. Our results indicate that performance is very close, within $0.5dB$, to that achieved with exact knowledge of the channel. Thus, the upper bound on the performance in the causal case is close to that achievable in the non-causal case.

5. CONCLUSIONS

In this paper we have formalized the problem of end-to-end delay constrained allocation for time varying channels. Using our technique it is possible to find an upper bound on achievable SNR for a given source and channel realization, thus allowing us to choose the right end-to-end delay for the given application. We have also studied the effect of using a model of the channel instead of the actual channel data. Our results indicate that a real time system (where causality is obviously required) would benefit from implementing a rate control which employs a channel model. In future work we will consider using a finite memory window for the video encoder as well, and we will compare our approach to buffer control techniques that require no encoding delay.

6. REFERENCES

[1] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. on CAS for video tech.*, vol. 2, pp. 361–372, Dec. 1992.

[2] J. Zdepsky, D. Raychaudhuri, and K. Joseph, "Statistically based buffer control policies for constant rate transmission of compressed digital video," *IEEE Trans. on Comm.*, vol. 39, pp. 947–957, June 1991.

[3] C.-T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudoconstant bit rate video coding," *IEEE Trans. on Image Proc.*, vol. 2, pp. 50–59, Jan. 1993.

[4] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. on Image Proc.*, vol. 3, pp. 26–40, Jan. 1994.

[5] V. K. Varma, "Testing speech coders for usage in wireless communications system," in *Proc. of IEEE Speech Coding Workshop*, (Montreal), 1993.

[6] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit-rate video transmission over fading channels for wireless microcellular systems," *IEEE Trans. on Circ. and Sys. for Video Tech.*, 1995. To appear.