Perceptual ARQ for H.264 Video Streaming over 3G Wireless Networks

Paolo Bucciol*, Enrico Masala† and Juan Carlos De Martin*
†Dipartimento di Automatica e Informatica / *IEIIT-CNR — Politecnico di Torino
Corso Duca degli Abruzzi, 24 — I-10129, Torino, Italy
Email: [paolo.bucciol | masala | demartin]@polito.it

Abstract— **We present a new ARQ algorithm for video streaming over wireless channels. The algorithm takes into account the perceptual and temporal importance of each packet to determine the packet scheduling which maximizes the perceived quality. We propose a simple and flexible function to combine the perceptual importance with the real-time constraints of each packet to determine which is the best packet to transmit at each transmission opportunity. The perceptual importance is evaluated using the analysis-by-synthesis technique. The performance of the proposed algorithm has been analyzed by simulating the transmission of H.264 encoded sequences over a 144 kbit/s UMTS channel. We compared the proposed method with time-driven ARQ techniques, using PSNR as distortion measure. The results show that for the considered channel conditions the proposed method delivers gains up to 2 dB with respect to the time-driven ARQ technique.**

I. Introduction

Video streaming is becoming one of the most interesting wireless applications. Delivering good video quality over wireless channels, however, is difficult because of channel noise and bandwidth limitations. In the case of streaming an end-to-end delay of a few seconds is acceptable, therefore the effect of transmission errors can be mitigated using a playout buffer and Automatic Repeat reQuest (ARQ) techniques to recover lost or corrupted data.

To optimize bandwidth usage, most multimedia ARQ techniques carefully consider one or both of the main features of multimedia traffic: its being time-sensitive and its highly non-uniform perceptual importance. The Soft ARQ proposal [1], for instance, avoids retransmitting late data that would not be useful at the decoder, thus saving bandwidth. Variants of the Soft ARQ technique have been developed for layered coding [1].

Other techniques suggest to assign different priorities to the syntax elements of the compressed multimedia bitstream. In [2], video packets are protected by error correcting codes whose amount depends on the kind of frame to which the video packets belong. Channel adaptation is achieved by an additional ARQ scheme that privileges the most important classes of data. Scheduling of video frames according to the priority given by their position inside the Group of Pictures (GOP) in presented in [3]. The technique is further enhanced by assigning different priorities to the various kinds of data (i.e. motion and texture information) contained in each packet.

Optimizing the transmission policy for each single packet [4] [5] leads to improvements with respect to techniques based on a priori determination of the average importance of the elements of the compressed bitstream. The low-delay wireless video transmission system presented in [6] includes an ARQ scheme where packets are retransmitted or not depending on whether the distortion caused by their loss is above a given threshold; however, it is not clear how to optimally determine such threshold. Given a way to associate distortion values to each packet, rate-distortion optimization of the transmission policies has also been proposed [7] [8].

Our proposal is to implement an ARQ scheme in which the retransmission policy is driven by the information about the *perceptual* and the *temporal* importance of each packet. The best packet to transmit at each transmission opportunity is selected by means of the importance value of each single packet, determined using a simple and flexible formula, that combines perceptual importance and the maximum delay constraint. Perceptual importance is evaluated using the analysis-by-synthesis technique described in Section III.

In this work we design a new ARQ protocol for wireless transmission and we simulate it in its entirety, including the acknowledgement packets, in the specific case of a UMTS channel at 144 kbit/s. We model both forward and backward channel losses through a Gilbert model whose parameters are extracted from wireless channel simulations in different fading conditions. The sequences are encoded using the state-of-the-art H.264 video coding standard [9]. Detailed PSNR results are reported, analyzing the influence of various parameters on the performance.

This paper is organized as follows. Section II describes the scenario, including the H.264 setup for the wireless transmission. In Section III we present an analysis-by-synthesis approach to evaluate the perceptual importance of the video packets, we analyze the transmission constraints and we design the ARQ algorithm. Section IV shows performance comparisons with other methods as well as the influence of some key algorithm parameters. Finally, conclusions are drawn in Section V.

II. Video Streaming Transmission

This paper focuses on the streaming of multimedia data from a base station to a mobile device. The aim is to design an ARQ transmission policy for the base station based on feedback information, and at the same time to keep the mobile device algorithms as simple as possible.

0-7803-8533-0/04/$20.00 (c) 2004 IEEE

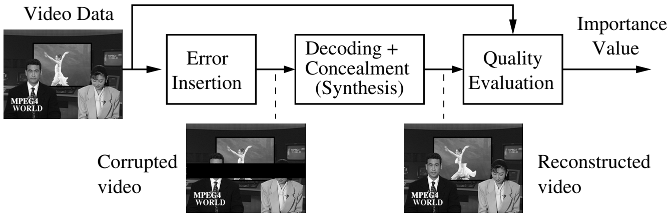

Fig. 1. Block diagram of the analysis-by-synthesis technique.

In a typical streaming transmission the time interval between the transmission and the playback of a packet, being several times the Round Trip Time, can be used to recover transmission errors with an ARQ technique. The client periodically sends to the transmitter acknowledgements for each packet that has been correctly received. The sender uses such information to decide which packets to retransmit. Due to channel noise, the sender does not always receive the acknowledgements.

We focus on the transmission of video data compressed according to the new ITU-T H.264 standard [9]. In the H.264 Video Coding Layer (VCL), consecutive macroblocks are grouped into *slices*, that are the smallest independently decodable units. They are useful to subdivide the coded bitstream into packets at locations that allow independent reconstruction, that is, the loss of a packet does not affect the ability of the receiver to decode the others. To transmit the video data over an IP network, the H.264 provides a Network Adaptation Layer (NAL) [10] for the Real-Time Transport Protocol (RTP), which is well suited for real-time multimedia transmissions and also for transmission over wireless channels in conjunction with header compression schemes [11] that reduce the header overhead.

Some dependencies exist between the VCL and the NAL. The packetization process is an example. Error resilience, in fact, is improved if the VCL is instructed to create slices of about the same size of the packets and the NAL told to put only one slice per packet, thus creating independently decodable packets. Note that in H.264 the subdivision of a frame into slices can vary for each frame of the sequence. However slices cannot be too short due to the resulting overhead that would reduce coding efficiency.

This paper is focused on a wireless transmission scenario where the lower protocol layers are designed to transmit packets of fixed size, called *frames*. Our approach is to create each slice of the video sequence adding macroblocks until the slice size approaches the frame size, then to begin another slice.

III. The Perceptual ARQ

A. The Analysis-by-Synthesis Classification Technique

Multimedia data, and video in particular, exhibit non-uniform perceptual importance. When video is transmitted over a noisy channel, each loss event causes a decrease of the video quality that depends on the perceptual importance of the lost data.

A simple way to define the importance of a certain video coding element such as a macroblock or a packet is to consider the distortion that would be introduced at the decoder by the loss of that element. Given a compressed video stream already subdivided into packets, the following steps are performed for each packet to compute the distortion values:

1) decoding of the bitstream simulating the loss of the packet being analyzed, including the concealment applied by the decoder (synthesis stage);
2) computation of the distortion between the reconstructed and the original sequence, using the Mean Squared Error (MSE) as distortion measure;
3) storage of the obtained value as an indication of the perceptual importance of the analyzed video packet.

Figure 1 shows the block diagram of the described analysis-by-synthesis technique, when a packet corresponding to a row of macroblocks is analyzed.

This approach is completely independent of the video coding standard. Since it includes the synthesis stage in its body, it can accurately evaluate the effect of both error propagation and error concealment. Some applications of the analysis-by-synthesis approach to MPEG coded video can be found in [4] [5] [8].

The complexity and delay of the analysis-by-synthesis classification technique depend on the frame types the sequence is composed of. If only I-type frames are present, the technique becomes simpler because each frame is coded independently of the others. The potential distortion of a macroblock is the MSE value between the macroblock in the original frame and the one used for the concealment. Here we always assume the availability of that macroblock, which is realistic for moderate packet losses. If the sequence contains also predicted frames such as in the case of H.264, the algorithm is more complex because the temporal propagation of the errors must be taken into account until the intra refresh mechanism has significantly reduced them.

The values computed by the analysis-by-synthesis technique are specific of any given coded sequence. In the case of stored video (e.g. non-live streaming scenarios), the distortion values can be precomputed and stored, then later used to optimize the transmission policy in real-time.

B. The Importance Function

In a real-time streaming scenario each packet must be available at the decoder a certain amount of time before it is played back to allow the decoder to process it. Let t_n be the time the n-th frame is played. All packets containing data needed to synthesize the n-th frame must be available at the decoder at time $t_n - T_P$ where T_P is the decoder processing time. For simplicity T_P is supposed to be equal for each frame. Note that the temporal dependencies present in the coded video (e.g. due to B-type frames) must also be taken into account.

For each packet i belonging to the n-th frame we define its deadline as $t_{i,n} = t_n - T_P$. If a packet never arrives, or arrives after $t_{i,n}$, it produces a distortion $D_{i,n}$ that can be evaluated using the analysis-by-synthesis technique. The sender should always select a packet for transmission only among the ones that can arrive before their deadline, i.e. $t_{i,n} > t_s + FTT$, where t_s is the time when the packet would be sent and FTT (Forward Trip Time) is the time needed to transmit the packet including the propagation time. FTT is assumed constant because the packet size is fixed. Defining the distance from the deadline as $\Delta t_{i,n} = t_{i,n} - t_s$, the previous condition can be rewritten as $\Delta t_{i,n} > FTT$.

Assuming that the bandwidth required by the encoded video is less than the available wireless channel bandwidth towards the receiver, the remaining part is used to retransmit the packets that have not been correctly received according to a policy that will be described in the following.

At any given time a number of packets satisfy the condition $\Delta t_{i,n} > FTT$. A policy is needed to choose which packet must be transmitted and in which order. Consider the packets containing the video data of a certain frame: each packet has the same $\Delta t_{i,n}$. Within a frame the sender should transmit, or retransmit, the packet with the highest $D_{i,n}$ that has not been yet successfully received. The decision is not as clear when choosing between sending an element A with low distortion $D_{A,n-1}$ in an older frame and an element B with high distortion $D_{B,n}$ in a newer frame. In other words, there is a tradeoff between the importance of the video data and its distance from the deadline (which can be seen as a sort of temporal importance.) A reason in favor of sending A is because its playback time is nearer ($\Delta t_{A,n-1} < \Delta t_{B,n}$) so there will be fewer opportunities to send it. On the other hand, if B arrives at the decoder, it will reduce the potential distortion of a value greater than A (because $D_{B,n} > D_{A,n-1}$.) A detailed study of this problem can be found in [1].

A good transmission policy is needed to select at each transmission opportunity the video packet that optimizes a given performance criterion. To simplify the problem we propose to compute, for each packet, a function of both its potential distortion and its distance from the deadline:

$$V_{i,n} = f(D_{i,n}, \Delta t_{i,n}). \tag{1}$$

The retransmission policy consists of sending packets in decreasing order of $V_{i,n}$. The problem is to find a good, and if possible simple, function to combine the distortion value with the distance from the deadline. We propose to use the following function:

$$V_{i,n} = D_{i,n} + wK \frac{1 \text{ s}}{\Delta t_{i,n}}. \tag{2}$$

The K coefficient is a normalization factor, computed as the product of the mean value of the distortion produced in case of loss by each packet and the inverse of the receiver buffer length T_B in seconds as in the following formula

$$K = \overline{D_{i,n}} \cdot \frac{1 \text{ s}}{T_B}. \tag{3}$$

The normalization factor is designed to give more importance to the distance from the deadline, i.e. the time constraint, if the size of the receiver buffer T_B is small. The weighting factor w in (2) is introduced to easily change the relative importance of the two parts of the formula. Initially, its value is set to 1. Its influence on the overall performance is studied in Section IV.

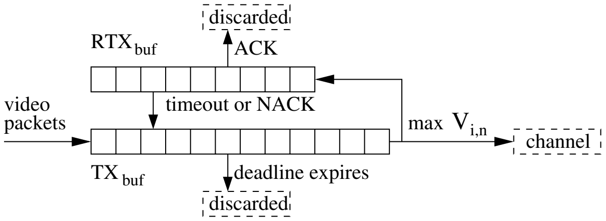

Fig. 2. Diagram of the sender-side scheduling algorithm.

C. The Scheduling Algorithm

Assume to transmit over an asymmetric bidirectional wireless link, as in a UMTS TDD transmission. Let B be the forward channel bandwidth and B/r the feedback channel bandwidth, with r being the feedback channel ratio. Every r sent packets the sender receives an acknowledgement packet containing an ACK/NACK flag for each received packet. A NACK is inserted when the receiver detects a missing packet by means of the RTP sequence number. The sender matches the RTP sequence number with the corresponding packet because it keeps all the information for each sent packet until it receives the corresponding acknowledgement.

The algorithm used by the sender to implement the retransmission policy is based on two buffers: a transmission buffer TX$_{buf}$ and a retransmission buffer RTX$_{buf}$. A packet is put in the TX$_{buf}$ when the condition $\Delta t_{i,n} < T_B + FTT$ is satisfied. This implements a window mechanism that prevents the transmission of packets the receiver could not buffer because they are too ahead in time. Packets belonging to TX$_{buf}$ or RTX$_{buf}$ whose $\Delta t_{i,n} < FTT$ are discarded because they will never arrive at the decoder in time for playback. When a packet is sent, it is put in the RTX$_{buf}$, waiting for its acknowledge. When an ACK is received, the corresponding packet in the RTX$_{buf}$ is discarded because it has been successfully transmitted. If a NACK is received for a packet in RTX$_{buf}$ or its timeout expires, the packet is put in the TX$_{buf}$ again if $\Delta t_{i,n} > FTT$, otherwise it is discarded.

Each time it is possible to send a new packet, the value of Equation (2) is computed for each packet in the TX$_{buf}$ and the one with the highest value is transmitted. Figure 2 illustrates the scheduling algorithm. Note that each packet has the same size, therefore the ARQ algorithm does not take into account that aspect.

IV. Results

The proposed technique has been implemented and tested using the H.264 test model software [12]. The sequence was encoded using one bi-directional predicted frame for each P-type frame. Only the first frame was coded as I-type. To improve error resilience, the macroblocks of a frame where cyclically coded as intra, sending one intra packet each two

TABLE I

GILBERT MODEL PARAMETERS FOR DIFFERENT CHANNEL CONDITIONS.

Channel #	Packet loss probability	Average error burst length
1	0.078	3.645
2	0.128	3.139
3	0.176	3.660
4	0.232	3.862
5	0.265	4.487
6	0.292	4.425
7	0.333	5.023

frames. All the tested sequences were QCIF at 10 fps. The parameters of the simulation were chosen among the ones provided by the UMTS specifications. The channel bandwidth was 144 kbit/s; the resulting packet size was 180 bytes, that corresponds to the fixed slot time T_S of 10 ms. The target coding rate was set to about 120 kbit/s, in order to leave 20% of the bandwidth available for retransmission purpose. The target rate was met by adjusting the quantization parameter. The encoder was instructed to make slices whose size was as close as possible, but not greater, than the packet size. Assuming to use a header compression protocol, we allocated ten bytes to each packet header. The packets sent on the feedback channel contain information about the last 100 received packets. Each packet is a vector containing, for each received packet, the least significant bits of its RTP sequence number and its reception status. The packet size is the same as for the forward channel. The acknowledgement information is repeated in different packets to increase error robustness.

The decoder implements a simple temporal concealment technique that replaces a corrupted or missing macroblock with the most recently decoded macroblock in the same position in the previously decoded frame. The packet corruption process has been simulated using a Gilbert model whose parameters, reported in Table I, are based on UMTS MAC and radio channel simulations in different fading conditions. The forward and feedback channels share, for simplicity's sake, the same Gilbert parameters, but different statistical realizations have been used. The results shown in the figures are the average of several channel realizations to obtain valid statistical results. The processing time T_P was kept constant and equal to 10 ms in all the simulations. The w parameter of (2) has been set equal to 1 in all cases except where otherwise indicated. In the following simulations, the transmissions were assumed to start only after a setup interval during which the buffer of the receiver was filled.

A. Comparison with Time-Driven ARQ Methods

Comparisons have been carried out to show the performance gain of the proposed method with respect to two ARQ approaches that do not take into account the perceptual importance of the packets. With the *time-driven ARQ* method only the distance from the deadline $\Delta t_{i,n}$ is considered to decide which packet should be retransmitted. The *perfect feedback time-driven ARQ* algorithm is similar to the previous one, but no losses are experienced in the feedback channel.

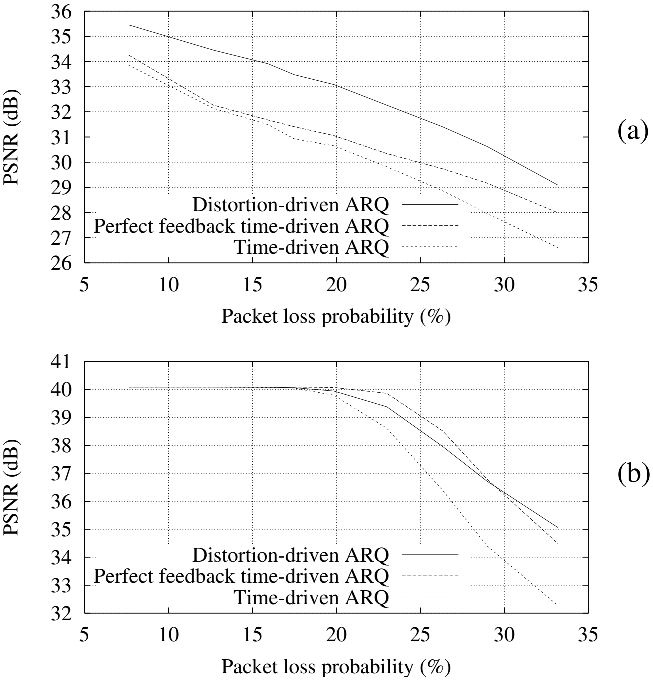

Fig. 3. Performance of the *distortion-driven ARQ*, *perfect feedback time-driven ARQ* and *time-driven ARQ*: *Foreman* (a) and *News* (b) sequence; FTT 10 ms; buffer length 10 frames; timeout 80 ms; feedback channel ratio $r = 5$.

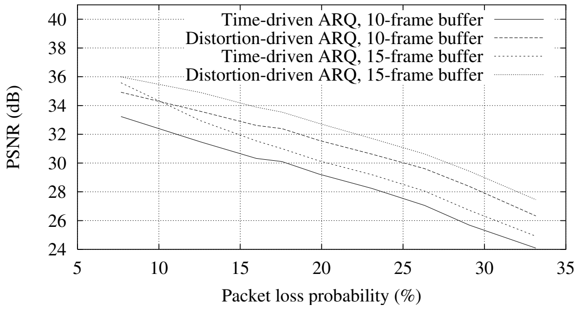

Fig. 4. PSNR values as a function of packet loss probability for playout buffer lengths of 10 and 15 frames; *Foreman* sequence; FTT 10 ms; timeout 80 ms; feedback channel ratio $r = 5$.

Figure 3 shows the PSNR performance of the three algorithms transmitting the well-known *Foreman* (a) and *News* (b) sequences. The PSNR is computed as the average of the PSNR of each frame with respect to the original uncompressed sequence. The proposed ARQ method clearly outperforms the *time-driven ARQ* one. In *Foreman*, due to a moderate degree of motion, the use of the perceptual importance information gives a good advantage also on the *perfect feedback time-driven ARQ*. In *News* this happens for high error probabilities because a significant share of the images is static, thus the temporal concealment technique is often effective. For low error probabilities, the ARQ protocols were able to recover most of the lost packets.

B. Influence of Parameters

The performance of the proposed ARQ algorithm depends on several parameters. The first one is the receiver buffer size T_B. If a larger playout buffer is used, the PSNR increases

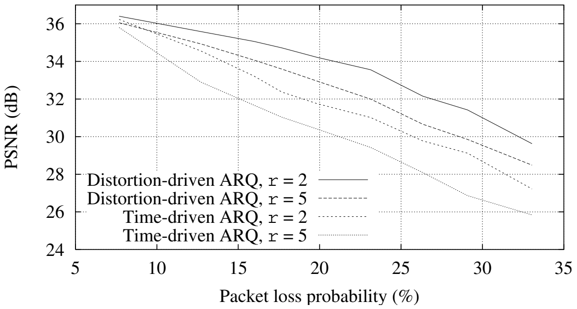

Fig. 5. Effect of feedback channel bandwidth as a function of packet loss probability; *Foreman* sequence; FTT 10 ms; buffer length 10 frames; timeout 80 ms.

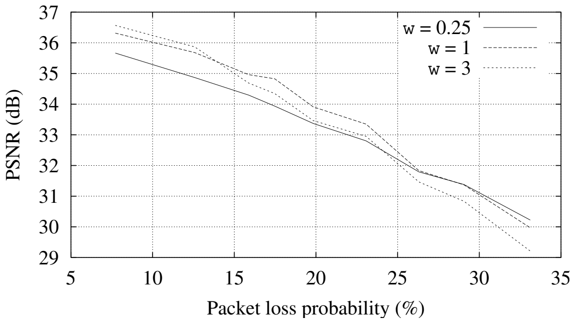

Fig. 6. PSNR as a function of packet loss probability for w=0.25, 1, 3; *Foreman* sequence; FTT 10 ms; buffer length 5 frames; timeout 80 ms; feedback channel ratio $r = 5$.

because packets become available for transmission at an earlier time, thus increasing their transmission opportunities. Figure 4 shows the PSNR value for the *Foreman* sequence using buffer lengths of 1 s and 1.5 s, corresponding to 10 and 15 frames. Note that the proposed ARQ method obtains a higher PSNR value than the time-driven ARQ method even using a shorter buffer. This makes the proposed algorithm suitable for mobile receivers, whose memory and power resources are limited.

Another interesting parameter to consider is the amount of bandwidth reserved to the feedback channel. Figure 5 shows the performance of the system in the case of two different feedback channel ratio. A high feedback channel bandwidth permits to send more frequent acknowledgement packets, therefore the sender can choose the best packet to transmit based on more updated receiver status information. Moreover, a high feedback channel bandwidth allows more repetitions of the acknowledgement information. This minimizes the probability that the ACK or NACK information does not reach the sender. The graph shows that the proposed ARQ method can deliver higher performance in comparison with the time-driven ARQ methods with much less frequent information updates about the status of the receiver.

The weighting factor w introduced in Eq. (2) allows to control the relative importance of the perceptual and temporal terms of the formula. If the value is large, the proposed scheduling algorithm gives more priority to the packets that are nearer to their deadlines rather than to those whose perceptual, rather than temporal, importance is high. A small value of w has the opposite effect: the packets will be preferably selected if their perceptual importance is high. Figure 6 shows the PSNR for the *Foreman* sequence for different values of w. In case of high error probabilities, high PSNR values are obtained when w is low. In fact, it is better to privilege packets that would produce a high distortion in case of loss, because they will probably be corrupted or lost and more opportunities are needed to carry out a successful transmission. For low error probabilities, the opposite consideration holds.

V. Conclusions

In this paper we proposed and analyzed a new ARQ technique that takes into account both the temporal and the perceptual importance of each packet when selecting which packets to retransmit. An analysis-by-synthesis technique has been presented to evaluate the perceptual importance of packets by simulation of the decoding process in case of loss. A function has been proposed to weight the resulting values taking into account the real-time constraints of the receiver playback process. Simulations of wireless transmission over a UMTS channel showed that the proposed ARQ method outperforms the standard approaches and in particular the time-driven ARQ technique.

References

[1] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for layered streaming media," in *Tech. Rep. UCB/CSD-98-1024, University of California, Computer Science Division, Berkeley*, November 1998.

[2] Y. Shan and A. Zakhor, "Cross layer techniques for adaptive video streaming over wireless networks," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, vol. 1, August 2002, pp. 277–280.

[3] S. H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *Proc. Packet Video Workshop*, Pittsburgh, PA, April 2002.

[4] E. Masala, D. Quaglia, and J. D. Martin, "Adaptive picture slicing for distortion-based classification of video packets," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Cannes, France, October 2001, pp. 111–116.

[5] F. De Vito, L. Farinetti, J.C. De Martin, "Perceptual classification of MPEG video for Differentiated-Services communications," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, vol. 1, Lausanne, Switzerland, August 2002, pp. 141–144.

[6] S. Aramwith, C.-W. Lin, S. Roy, and M.-T. Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 558–565, June 2002.

[7] J. Chakareski, P. A. Chou, B. Aazhang, "Computing rate-distortion optimized policies for streaming media to wireless clients," in *Proceedings of Data Compression Conference*, April 2002, pp. 53–62.

[8] E. Masala and J. D. Martin, "Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, Baltimore, MD, July 2003.

[9] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services," *ITU-T*, May 2003.

[10] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, July 2003.

[11] C. Bormann et al., "RObust header compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed," *RFC 3095*, July 2001.

[12] ITU-T VCEG, "Test Model Long Term (TML) 9.7," *URL: ftp://standard.pictel.com/video-site*, February 2002.