Cross-Layer Perceptual ARQ for H.264 Video Streaming over 802.11 Wireless Networks

P. Bucciol*, G. Davini*, E. Masala†, E. Filippi‡ and J.C. De Martin*

*IEIIT-CNR / †Dipartimento di Automatica e Informatica — Politecnico di Torino, Italy
‡Advanced System Technologies, STMicroelectronics — Cornaredo (Milano), Italy
Email: [paolo.bucciol | gabriele.davini | masala | demartin]@polito.it, enrica.filippi@st.com

Abstract— **We present a new cross-layer ARQ algorithm for video streaming over 802.11 wireless networks. The algorithm combines application–level information about the perceptual and temporal importance of each packet into a single priority value, which drives packet selection at each retransmission opportunity. Hence, only the most most perceptually important packets are retransmitted, delivering higher perceptual quality and less bandwidth usage compared to the standard 802.11 MAC-layer ARQ scheme. H.264 video streaming based on the proposed technique has been simulated using *ns* in a realistic home network scenario, using the standard ARQ technique for all interfering traffic. Results show that the proposed method consistently outperforms the standard MAC-layer 802.11 retransmission scheme, delivering more than 1.5 dB PSNR gains using approximately half of the retransmission bandwidth.**

I. Introduction

An increasing number of mobile devices currently uses wireless interfaces based on the IEEE 802.11 WLAN standard [1] for network access. The 802.11 standard Medium Access Control (MAC) implements a link-layer retransmission scheme to cope with channel noise and transmission collisions. While the standard link-layer ARQ can be an adequate solution for generic data transmissions, more efficient ARQ techniques would be highly desirable for high-volume, delay-sensitive multimedia traffic. A multimedia-optimized ARQ technique could, in fact, deliver higher perceptual quality as well as optimize the use of network resources.

Two of the most important characteristics of multimedia streams are their highly non-uniform perceptual importance and their strong time sensitivity. One or both characteristics are usually considered by most ARQ techniques designed for multimedia communications. For instance, the *Soft ARQ* proposal [2] saves bandwidth avoiding retransmission of late data that would not be useful at the decoder.

Compressed multimedia bitstreams are composed of syntax elements of varying preceptual importance. Some techniques exploit this feature by assigning different priorities to the syntax elements. For instance, in [3] video packets are protected by error correcting codes whose strength 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. Other schemes change the scheduling of video frames according to the priority given by their position inside the Group of Pictures (GOP), as in [4]. In that work, 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 has been shown to further improve performance [5]. For instance, packets could be retransmitted or not depending on whether the distortion caused by their loss is above a given threshold, as in the low-delay wireless video transmission system presented in [6]. However, it is not clear how to optimally determine such threshold. Rate–distortion optimization of the transmission policies has also been proposed [7][8].

In this paper, we focus on the specific case of video streaming over 802.11 networks. Unlike the 802.11 MAC-level ARQ which retransmits all packets regardless of their importance, we propose an ARQ scheme at the application level to exploit information about the *perceptual* and the *temporal* importance of each packet. In our proposal, a set of retransmission opportunities is determined on a GOP-by-GOP basis, then the algorithm retransmits unacknowledged packets according to their priority. Each packet's priority is computed using a simple and flexible formula, that combines perceptual and temporal importance. Perceptual importance is evaluated using the analysis-by-synthesis technique. Temporal importance is a function of how distant is the playout deadline.

The proposed technique has been thoroughly analyzed by means of H.264 [9] video streaming simulations in a realistic home network scenario, in presence of several concurrent interfering flows that were transmitted using the standard ARQ technique. Both perceptual and network performance results show the considerable gains achieved by the proposed scheme with respect to the standard 802.11 retransmission technique. The results also analyze the impact of the main parameters of the algorithm on the performance.

This paper is organized as follows. Section II and Section III review the H.264 standard and analysis-by-synthesis distortion estimation, respectively. In Section IV the proposed perceptual ARQ technique is presented in detail. Results are discussed in Section V, while conclusions are drawn in Section VI.

II. H.264 Video Transmission

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

0-7803-8794-5/04/$20.00 (C) 2004 IEEE

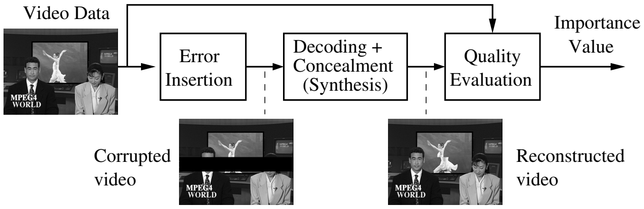

Fig. 1. Block diagram of the analysis-by-synthesis technique.

grouped into *slices*, that are the smallest independently decodable units. They are useful to subdivide the coded bitstream into independent packets, so that 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 wired and wireless multimedia transmissions.

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.

III. Analysis-by-Synthesis Distortion Estimation

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. Such importance can be defined *a priori*, based on the average importance of the elements of the compressed bitstream, as with the data partitioning approach.

At a finer level of granularity, the importance of a video coding element, such as a macroblock or a packet, could be considered proportional to the distortion that would be introduced at the decoder by the loss of that specific element. The distortion estimate associated to each packet could, therefore, be computed as follows:

1) decoding (including concealment) of the bitstream simulating the loss of the packet being analyzed (synthesis stage);
2) computation of the distortion (e.g. MSE) between reconstructed and original sequence.

The obtained value is then stored as an indication of the perceptual importance of the analyzed video packet. Figure 1 shows the block diagram of the above described analysis-by-synthesis approach.

The analysis-by-synthesis distortion estimation scheme is independent of the video coding standard. Since it includes the synthesis stage in its body, it can accurately evaluate the effect of both the error propagation and the error concealment. Some applications of the analysis-by-synthesis approach to MPEG coded video can be found in [11] [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 is quite simple since each frame is coded independently of the others. If the sequence contains also predicted frames such as in the case of H.264, the algorithm is more complex because error propagation must be taken into account until the end of the GOP; a model-based approach, however, can be used to drastically reduce complexity [12].

IV. Cross-Layer Perceptual ARQ

To take into account the perceptual and temporal importance of each multimedia packet, an application-level, end-to-end ARQ technique using the IP-UDP-RTP/RTCP protocol stack is proposed. Every packet is transmitted once, then it is stored in a retransmission buffer RTX$_{buf}$ waiting for its acknowledgement. The receiver periodically generates RTCP receiver reports (RR) containing an ACK or a NACK for each transmitted packet. A NACK is generated when the receiver detects a missing packet by means of the RTP sequence number. Packets in the retransmission buffer are sent in the order given by their combined temporal-perceptual priority, as defined in Section IV-B. The performance of the proposed technique depends on a few key parameters, such as the maximum amount of bandwidth B_{max} granted to the transmission, the relative weights given to temporal and perceptual importance, and the receiver reports frequency.

A. The Retransmission Scheduling Algorithm

At the beginning of each GOP, the transmission time of each packet produced by the encoder is determined by equispacing the packets of each frame inside their respective frame interval. Let B_{GOP} be the bandwidth needed to transmit the current GOP and B_{max} the maximum amount of bandwidth granted to the transmission. N_{rtx} retransmission opportunities are available for the current GOP, where $N_{rtx} = (B_{max} - B_{GOP})/\overline{S}_{pck}$ and $\overline{S}_{pck}$ is the average packet size. The time instants corresponding to the retransmission opportunities are determined as follows. The total size of each frame is first computed and then the smallest one is identified. The time instant of the first retransmission opportunity is set to be midway between the time instant of the first packet of the smallest frame interval and the last packet of the previous frame. The procedure is repeated until N_{rtx} opportunities have been determined, considering at each step the opportunities filled by packets of size $\overline{S}_{pck}$. This procedure may create retransmission bursts between each frame, but has the advantage to be simple to implement; if desired, a more uniform distribution of the retransmission opportunities is achievable. Note also that the opportunities will not be necessarily completely used.

The algorithm used by the sender to implement the retransmission policy is based on a retransmission buffer RTX$_{buf}$. When a packet is sent, it is placed in the RTX$_{buf}$, waiting for its acknowledgement, and marked as *unavailable* for retransmission. 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, the packet is marked as *available* for retransmission. Packets belonging to the RTX$_{buf}$ that will never arrive at the decoder in time for playback are discarded. To limit the impact of receiver report losses, the sender piggybacks the highest sequence number for which it received an ACK or NACK. The receiver always repeats in the receiver reports the status information for all the packets whose sequence number is less than the piggybacked one.

When a retransmission opportunity approaches, a priority function (see Section IV-B) is computed for each packet marked as *available* in the RTX$_{buf}$ and the one with the highest priority is transmitted. It is important to stress that the retransmission opportunities computed according to B_{max} *not necessarily* will be actually used by the algorithm, leading to an actual bandwidth usage which can be considerably lower than B_{max}.

B. The Priority 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 back. 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. 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 (i.e. the time instant by which the packet must reach the decoder) as $t_{i,n} = t_n - T_P$. If a packet never arrives, or arrives after $t_{i,n}$, it produces a distortion increase $D_{i,n}$ 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 instant of the next retransmission opportunity and FTT (Forward Trip Time) is the time needed to transmit the packet, which is typically time-varying, due to the network state. Defining the distance from the deadline as $\Delta t_{i,n} = t_{i,n} - t_s$, the previous condition can be rewritten as $\Delta t_{i,n} > FTT$.

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 retransmitted 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 ($\Delta t_{A,n-1} < \Delta t_{B,n}$), that reduces the number of 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 the problem can be found in [2].

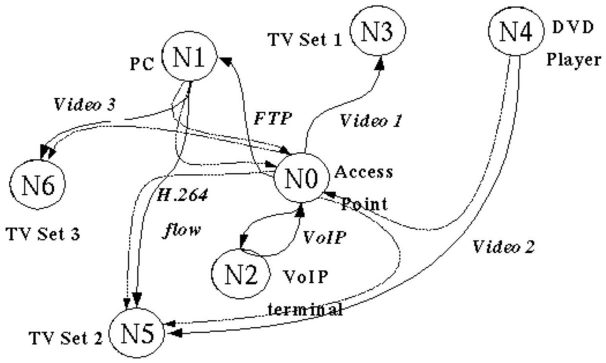

Fig. 2. The 802.11 network scenario.

TABLE I
CHARACTERISTICS OF THE STREAMS.

Stream	Bandwidth	Retry limit
Video 1 / 2 / 3	1.5 / 1.5 or 3 / 6 Mbit/s	3
FTP	variable	3
VoIP	70 kbit/s	3
Tested H.264 stream	0.128 or 0.765 Mbit/s	0
Receiver Reports (RR)	max 6 kbit/s	0 or 3

A retransmission policy is needed to select at each retransmission opportunity the video packet that optimizes a given performance criterion. We propose to compute, for each packet, a priority 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 priority $V_{i,n}$. The issue is to find an effective, and, if possible, simple, function that combines the distortion value with the distance from the deadline. We propose to use the following function:

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

where K is a normalization factor, computed as the product of the mean value of the distortion and the receiver buffer length T_B in seconds as in the following formula

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

The normalization factor, K, is designed to balance the perceptual and temporal importance of the packet for the average case. The weighting factor w in Eq. (2) is introduced to control the relative importance of the perceptual and temporal terms of the formula.

V. Results

The proposed technique has been implemented and tested using *ns*. The simulator implements an 802.11e MAC layer [13] over an 802.11a physical layer with a channel bandwidth of 36 Mbit/s. A packet error model has been implemented

TABLE II

PERFORMANCE OF THE PROPOSED ARQ SCHEME AS A FUNCTION OF THE MAXIMUM TRANSMISSION BANDWIDTH; FOREMAN SEQUENCE

B_{max} (%)	Used bandwidth (%)	PSNR (dB)	Avg. MAC-level packet loss rate (%)	Transport-layer throughput (%)	Application-layer throughput (%)	Application-layer packet loss rate (%)
115	117	32.00	17.12	96.89	92.66	7.34
130	119	35.27	16.79	99.16	96.19	3.81
140	120	35.78	17.30	99.23	96.82	3.18
150	120	37.01	16.94	99.67	98.49	1.51
160	121	37.77	17.34	99.78	99.14	0.86
170	121	38.33	17.56	99.92	99.11	0.89
200	122	38.46	17.90	99.99	99.98	0.02

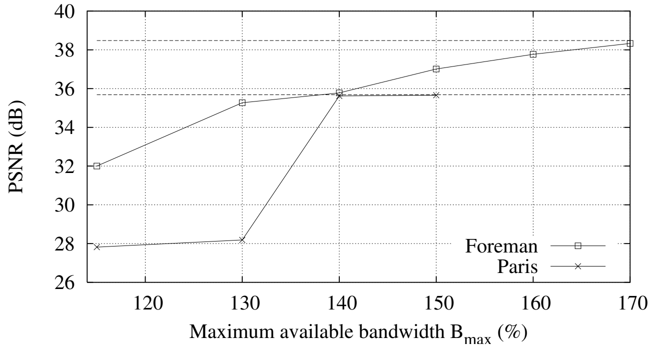

Fig. 3. Performance as a function of the maximum available transmission bandwidth B_{max}; w=1; RR interval is 200 ms; Video2 bandwidth is 1.5 Mbit/s. The horizontal lines show the encoding distortion.

in *ns* based on BER curves obtained from 802.11 channel measurements, with different noise levels and packet sizes. The network scenario is shown in Figure 2.

The retry limit is set to three for all the flows except the tested H.264 stream, which is transmitted using no retransmissions; both 0 and 3 retransmissions have been used for the Receiver Report (RR) RTCP flow. For implementation simplicity, 802.11e Access Categories (AC) have been used to differentiate the retry limit of the various flows, but wireless access parameters (hence access priority) are the same for all the AC's. Table I reports the bandwidth of the concurrent flows. The rate of the RTCP flow due to the receiver reports is very modest: it ranges between 3 and 6 kbit/s for a 100 ms RR interval, and, if needed, could be further reduced by packing ACK and NACK information more efficiently than in the current implementation.

The standard *Foreman* (QCIF, 176×144, 15 fps) and *Paris* (CIF, 352×288, 30 fps) test sequences have been encoded using version 6.1e of the H.264 test model software [9] with a fixed quantization parameter, resulting in an average bitrate of respectively 128 kbit/s and 765 kbit/s. The GOP encoding scheme is IBBPBBPBBPBB. Each sequence is concatenated with itself to reach a length of approximately 500 s. The video encoder is instructed to make RTP packets whose size is approximately constant. The playout buffer size is 1 s long. The decoder implements a simple temporal concealment technique that replaces a corrupted or missing macroblock with the macroblock in the same position in the previous frame.

TABLE III

PERFORMANCE OF THE STANDARD 802.11 ARQ SCHEME.

Sequence	Used bandwidth (%)	PSNR (dB)	Application-layer packet loss rate (%)
Foreman	140%	36.93	1.24%
Paris	158%	33.80	2.46%

The first set of results shows the performance of the proposed ARQ scheme as a function of the maximum bandwidth parameter, B_{max}, expressed as a percentage of the sequence average bitrate. Figure 3 shows the PSNR performance for the *Foreman* and *Paris* sequences. For the *Foreman* case, when the maximum available bandwidth is about 170% the quality nearly reaches the error-free encoder performance, represented by the 38.48 dB horizontal line. The actual bandwidth used by the algorithm is much lower (121%), as shown by the second column of Table II. The bandwidth value shown in Figure 3 is, in fact, the *peak* transmission bandwidth, fully used only when a GOP is particularly difficult to transmit. Therefore, the PSNR gain comes from the peak bandwidth increase that allows the algorithm to timely retransmit a higher number of packets when it is more needed. This behavior is illustrated by the throughput values in Table II, columns 5 and 6. Increasing the maximum retransmission bandwidth benefits the transport level throughput and also the application-layer throughput, that considers the packets as useful only if they arrive on time at the decoder. The PSNR performance and the application-layer throughput are directly related to the peak retransmission bandwidth B_{max}. For the *Paris* sequence, the increase is sharper because the number of packets with a high perceptual importance for that sequence is limited, therefore, when the maximum transmission bandwidth is higher than 130%, the proposed algorithm can retransmit all the important packets on time, nearly achieving the error-free encoder quality.

A second set of results regards the comparison with the standard 802.11 MAC level ARQ scheme. Table III shows the PSNR results achieved by the MAC level ARQ scheme. The maximum number of retransmissions at MAC level has been set to three, which is a good tradeoff between error-robustness, delay and network usage. The results indicate that for both sequences the proposed cross-layer perceptual ARQ technique achieves a considerably higher PSNR value using about half or less retransmission bandwidth with respect to the standard MAC level ARQ. In particular, the gain for the two considered sequences ranges between 1.5 and 1.8 dB PSNR. The performance gain is easily explained considering

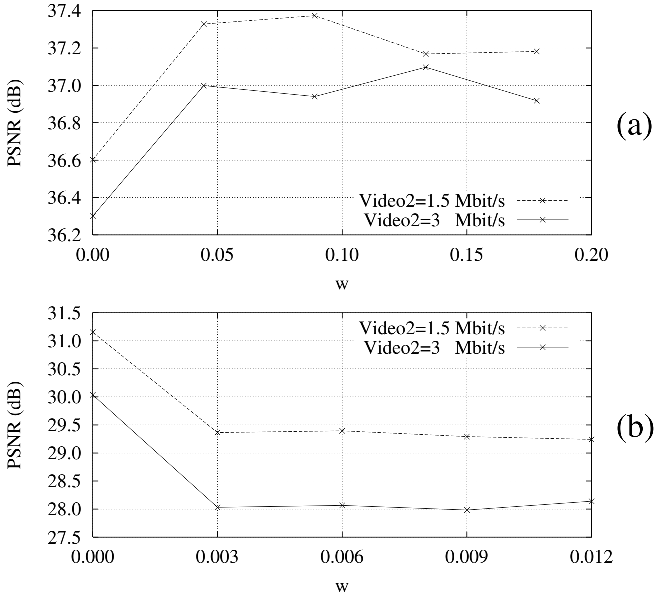

Fig. 4. Performance as a function of the w parameter: *Foreman* with B_{max}=160% (a) and *Paris* with B_{max}=130% (b); the receiver report interval is 50 ms.

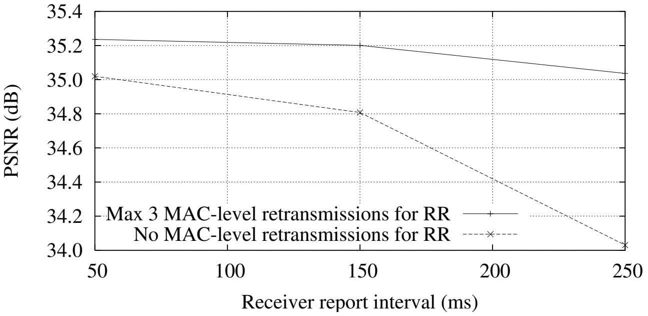

Fig. 5. PSNR as a function of the receiver report interval; Paris sequence; B_{max}=130%; w=0; Video2 bandwidth is 3 Mbit/s.

that the proposed ARQ algorithm has access to information not available to the link-layer level, such as the perceptual importance and the deadline of each packet. The standard 802.11 MAC level ARQ simply retransmit each packet until success or until reaching the maximum number of allowed retransmissions, regardless of its usefulness for the multimedia decoding process.

An important parameter of the proposed ARQ method is the weight given to the temporal importance (w in Equation 2). Figure 4 shows the PSNR values for two different levels of network congestion, that is Video2 bandwidth equal to 1.5 and 3 Mbit/s. As shown in the figures, the PSNR maximum depends on the considered sequence and on the network status. The *Foreman* sequence contains many perceptually important packets, therefore it is important that a high amount of them arrives on time, especially in case of high congestion. Therefore, the weight w of temporal importance in the priority function should be high. The *Paris* sequence, instead, presents a limited number of perceptually important packets, hence it is always important to privilege them setting the temporal importance weight w to zero.

Finally, the impact of the receiver report frequency is shown in Figure 5, for the case of zero and three maximum retransmissions. A lower value of the interval between two consecutive receiver reports leads to higher PSNR performance, because the sender status is more synchronized with the receiver status, therefore the scheduling and selection decisions can be more effective in improving the quality of the decoding process. Figure 5 also shows that, with an adequate protection of the receiver reports (i.e. up to three retransmissions), the impact of the receiver report frequency is very limited. If no protection is used instead (i.e. no retransmissions), it is important to quickly provide receiver report updates not to incur in significant performance degradation.

VI. Conclusions

In this paper we proposed and analyzed a cross-layer perceptual ARQ algorithm to transmit video streams on 802.11 wireless networks. The technique computes a priority function for each packet to determine the best scheduling and transmission instants to retransmit packets. Simulations with *ns* in a high traffic scenario showed consistent performance gains with respect to video transmissions using the content-transparent 802.11 MAC–level ARQ scheme. Finally, the impact of the main parameters on the algorithm has also been analyzed.

Acknowledgements

The authors would like to thank Mauro Bottero for his precious and timely help in performing some of the simulations. This work was supported in part by STMicroelectronics and by MIUR, Project FIRB-PRIMO, http://primo.ismb.it.

References

[1] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *ISO/IEC 8802-11, ANSI/IEEE Std 802.11*, 1999.

[2] 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.

[3] 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.

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

[5] F. De Vito, L. Farinetti, and 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. C. De Martin, "Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, vol. 3, Baltimore, MD, July 2003, pp. 345–348.

[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] E. Masala, D. Quaglia, and J. C. De 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.

[12] F. De Vito, D. Quaglia, and J. C. De Martin, "Model based distortion estimation for perceptual classification of video packets," in *Proc. IEEE Workshop on Multimedia Signal Processing*, September 2004, to appear.

[13] IEEE 802 Committee, "Draft supplement to standard - LAN/MAN specific requirements - Part 11: Medium access control (MAC) enhancements for quality of service (QoS)," *IEEE Std 802.11e Draft*, July 2003.