Application-Level Perceptual ARQ for H.264 Video Streaming over 802.11 Wireless LAN's

Application-level Perceptual ARQ for H.264 Video Streaming over 802.11 Wireless LAN's

P. Bucciol[2], G. Davini[2], E. Masala[1], E. Filippi[3], J. C. De Martin[2]

[1]Dipartimento di Automatica e Informatica / [2]IEIIT-CNR — Politecnico di Torino, Torino, Italy
[3]Advanced System Technologies, STMicroelectronics S.r.l. — Cornaredo (Milano), Italy
e-mail: [paolo.bucciol | gabriele.davini | masala | demartin]@polito.it, enrica.filippi@st.com

***Abstract*— A new ARQ algorithm for video streaming over 802.11 wireless networks is presented. The algorithm operates at the application level in order to exploit information about the perceptual and temporal importance of each packet. A priority value is associated to each packet to determine which one to retransmit at each retransmission opportunity. With respect to the standard 802.11 MAC-layer ARQ scheme, the proposed cross-layer technique delivers higher perceptual quality because it retransmits only the most perceptually important packets. Video streaming of H.264 test sequences has been simulated using *ns* in a realistic home network scenario, in which data flows have been assigned to different 802.11e access categories according to their QoS requirements. Results show that the proposed method consistently outperforms the standard link-layer 802.11 retransmission scheme, delivering more than 1.5 dB of PSNR gain with a very limited impact on concurrent traffic.**

***Keywords*— Perceptual ARQ, perceptual video importance, cross-layer techniques, H.264 video streaming, 802.11 wireless LAN.**

I. Introduction

The IEEE 802.11 wireless local area networking standard [1] provides network access to an ever expanding array of mobile devices. In 802.11, radio link noise and MAC-level collisions are addressed by an automatic link-layer retransmission scheme. While data-agnostic, link-layer ARQ is both fast and simple to implement, for the specific —and increasingly important— case of multimedia traffic, more advanced ARQ techniques could use network resources more efficiently as well as deliver higher perceptual quality.

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 [2], 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 [2].

Techniques based on assigning different priorities to the individual syntax elements of the compressed multimedia bitstream have also been proposed. In [3] 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 [4]. 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.

Further improvements are possible optimizing the transmission policy for each single packet, rather than relying on a priori determination of the average importance of the elements of the compressed bitstream [5][6]. The low-delay wireless video transmission system presented in [7] 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 [8][9].

For the specific case of video streaming over 802.11 networks, we propose to implement an ARQ scheme at the application level, to exploit information about the *perceptual* and the *temporal* importance of each packet —as opposed to the 802.11 MAC–level ARQ that retransmits all packets regardless of their importance. The proposed cross-layer ARQ algorithm determines, for each GOP, a set of retransmission opportunities and then retransmits non-acknowledged packets according to their priority. Each packet's priority is computed using a simple and flexible formula, that combines perceptual importance and maximum delay constraint. Perceptual importance is evaluated using the analysis-by-synthesis technique.

The proposed technique has been extensively studied by running simulations in a realistic home network scenario which included several concurrent interfering flows. Standard test sequences were encoded using the state-of-the-art H.264 video coding standard [10]. Both perceptual (measured by PSNR) and network performance results show the gains achieved by the proposed scheme with respect to the standard 802.11 retransmission technique. The results also show that the proposed perceptual ARQ technique has a limited impact on concurrent traffic.

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 [10]. 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 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) [11] 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

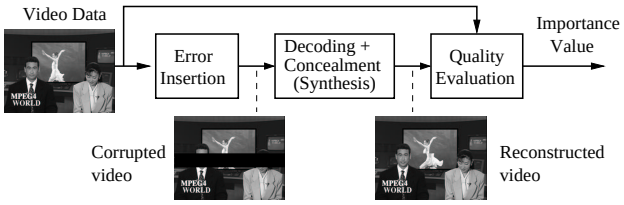

Fig. 1. Block diagram of the analysis-by-synthesis technique.

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;
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 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 [5] [6] [9].

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.

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 retransmissions, 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 retransmissions. 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

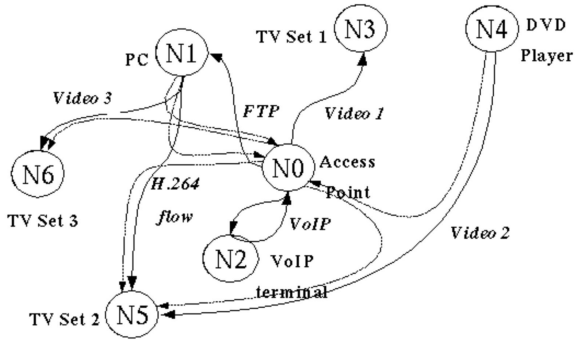

Fig. 2. The 802.11 network topology.

Tab. 1. Characteristics of the concurrent streams.

Stream	Bandwidth
Video1	1.5 Mbit/s
Video2	3 Mbit/s
Video3	6 Mbit/s
FTP	variable
VoIP	70 kbit/s

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].

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 size of the receiver buffer T_B is, in fact, approximately equal to the mean value of the distance from the deadline, assuming that the receiver buffer is almost full. 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* [12]. 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 in *ns* based on BER curves obtained from 802.11 channel measurements, with different noise levels and packet sizes. The scenario is shown in Figure 2. Traffic has been assigned to the 802.11e Access Categories as follows. The FTP stream is assigned to Access Category 0 (AC0). The tested H.264 stream is assigned to AC1, while all the remaining video flows are sent as AC2. The VoIP flows and the receiver reports are assigned to AC3, to provide protection against receiver report losses. The highest QoS is clearly offered by AC3. The maximum number of MAC retransmissions is three for all the classes except AC1, for which no MAC level retransmissions are used. We assigned the tested H.264 video stream and the other video flows to different access categories because the retry limit can be specified only for each access category and not for each flow. To ensure fairness in the comparisons, however, the tested H.264 stream flow has been assigned to an access category whose priority is lower than the other video streams. Table 1 reports the bandwidth of the concurrent flows. The rate of the RTCP flow due to the receiver reports is very modest. It ranges from 3 to 6 kbit/s for a 100 ms receiver report interval, and, if needed, could be further improved by packing ACK and NACK information more efficiently than the current implementation.

Tab. 2. Performance of the proposed ARQ scheme as a function of the maximum transmission bandwidth; *foreman* sequence.

B_{max} (%)	Used bandwidth (%)	PSNR (dB)	Application-layer packet loss rate (%)
170	136	32.59	16.60
200	145	35.13	9.21

Tab. 3. Performance of the proposed ARQ scheme as a function of the maximum transmission bandwidth; *paris* sequence.

B_{max} (%)	Used bandwidth (%)	PSNR (dB)	Application-layer packet loss rate (%)
170	143	32.86	19.31
250	163	34.26	4.66

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 [10] with a fixed quantization parameter, resulting in a bitrate of respectively 128 kbit/s and 765 kbit/s. The GOP encoding scheme is IBBPBBPBBPBB. The encoding distortion is 38.48 dB and 35.68 dB for the *foreman* and *paris* sequence respectively. 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.

The first set of results shows the performance of the proposed ARQ technique for different values of the maximum bandwidth parameter, B_{max}, expressed as a percentage of the sequence average bitrate. Table 2 and 3 show the PSNR performance for the *foreman* and *paris* sequences. Note that the actual bandwidth used by the algorithm is much lower, as shown by the second column of the previous tables. The B_{max} value 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.

Tab. 4. Performance of the MAC-level ARQ scheme; the maximum number of retransmissions is equal to three.

Sequence	Used bandwidth (%)	PSNR (dB)	Application-layer packet loss rate (%)
foreman	132	33.51	4.16
paris	140	31.52	9.72

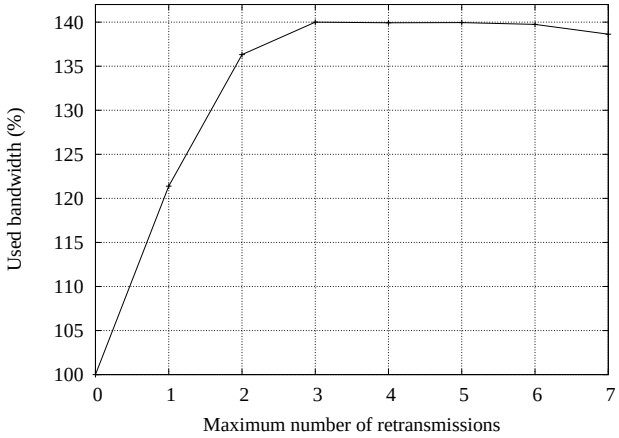

Fig. 3. Used bandwidth as a function of the retry limit for the MAC-level ARQ case; *paris* sequence.

A second set of results regards the comparison with the standard 802.11 MAC level ARQ scheme. Table 4 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 the proposed cross-layer perceptual ARQ technique can achieve a higher PSNR value with respect to the standard MAC level ARQ technique, using a slightly higher amount of retransmission bandwidth. For the *foreman* sequence, the PSNR value using about the same retransmission bandwidth is lower but remember that the other video sources are assigned to a higher-QoS access category than the H.264 video stream, therefore the performance of the proposed ARQ technique is underestimated. With a modest transmission bandwidth increase, however, the PSNR performance is definitely higher also for the *foreman* sequence. In particular, the gain for the two considered sequences ranges from 1.6 to more than 2 dB. The performance gain is easily explained considering 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 technique simply retransmits each packet until success or until reaching the maximum number of allowed retransmissions, regardless of its usefulness for the multimedia decoding process. Moreover, note that, even though the packet loss rate of the proposed ARQ technique is higher compared to the standard 802.11 ARQ technique (compare, for instance, the first row of Table 3 with Table 4), the PSNR of the proposed technique is higher. The packet losses are, in fact, concentrated on less perceptually important packets, for instance the ones containing B-type frames. The perceptual importance of these packets is limited because their data are not used to predict subsequent frames, therefore the visual artifacts caused in case of loss are limited to the frame to which they belong.

Tab. 5. Performance of the proposed ARQ scheme as a function of the w parameter; *paris* sequence.

Weight parameter	PSNR (dB)
0	34.26
1	33.80
13	33.62

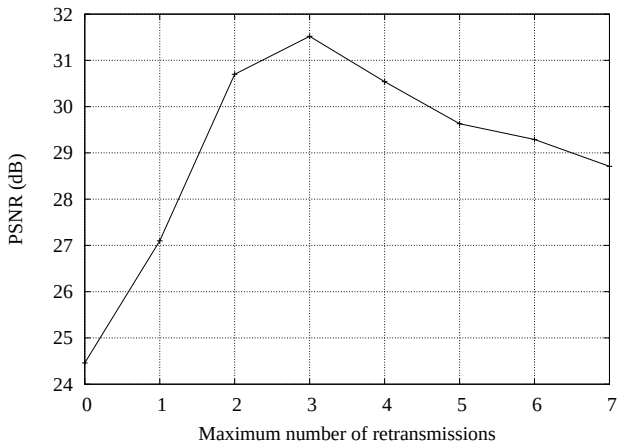

Fig. 4. PSNR as a function of the retry limit for the MAC-level ARQ case; *paris* sequence.

Tab. 6. Impact of the various techniques on concurrent traffic. Retry limit set to 3 for the MAC-level ARQ; B_{max} set to 170% for the proposed ARQ. *Paris* sequence.

Technique	Video1 PLR (%)	Video2 PLR (%)	Video3 PLR (%)	VoIP PLR (%)
MAC-level ARQ	25.72	28.23	27.89	0.33
proposed ARQ	26.32	29.20	28.90	0.34

The performance of the MAC-level ARQ technique is now evaluated to assess the impact of the retry limit setting. Figure 3 and 4 show the performance of the MAC-level ARQ scheme as a function of the retry limit, in terms of used transmission bandwidth and PSNR values. The first graph clearly shows that the used bandwidth saturates if the maximum number of retransmissions is increased over a certain threshold, that is about three in our simulations. The PSNR presents a maximum for that value. For higher values, the performance decreases due to the higher packet delay caused by severe network congestion.

An important parameter of the proposed ARQ method is the weight given to the temporal importance (w in Equation 2). Table 5 shows the PSNR values for three different values of the w parameter for the *paris* sequence. The value that maximizes the performance is zero. In fact, the number of perceptually important packets present in the *paris* sequence is limited, hence it is always important to privilege them by setting the temporal importance weight w to zero, regardless of less important packets whose deadline may be closer.

Finally, the impact of the proposed ARQ technique on the concurrent traffic is assessed. Table 6 shows the packet loss rate experienced by the concurrent flows in the network. The FTP flow is not shown because the throughput it can deliver is very limited and not significant due to the high network congestion. The packet loss rate increase for the video streams is very limited, namely about 1%, and it is negligible for the VoIP transmissions.

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 over the standard content-transparent 802.11 MAC–level ARQ scheme with a very limited impact on concurrent traffic.

Acknowledgements

This work was supported in part by STMicroelectronics and by MIUR, Project FIRB-PRIMO, http://primo.ismb.it.

The authors would like to thank Mauro Bottero for his precious and timely help in performing the simulations.

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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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*, Baltimore, MD, July 2003.

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

[11] 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.

[12] UCB/LBNL/VINT, "Network Simulator – ns – version 2," *URL: http://www.isi.edu/nsnam/ns*, 1997.

[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.